INTELLIGENT INFORMATION MANAGEMENT - THE CHALLENGES OF A SOM-BASED MAIL ORGANIZER

Antti Salovaara, Heikki Hyötyniemi, Juha Syri

Nokia Research Center, P.O. Box 407, FIN-00045 Nokia Group

The electronic forms of communication, such as e-mails, have in a brief time become extremely popular and are in wide use. The ease of electronic communication easily leads into the explosion of information, and the users soon notice that the management of communication becomes a challenge. The present study examined a way to alleviate this problem; an organizer for information management of e-mail messages was designed. The mail organizer uses linguistic technology and analyses the contents of the mail messages, producing for every message an index about key terms. Based on these key terms, a self-organizing map (SOM) analysis is run with the messages, producing a two-dimensional graph about them. The messages that have similar contents are shown close to each other in the graph. The results obtained suggest that a lot of work needs to be done before practical applications can be implemented based on this approach. The promises of the service as well as some challenges in it are also described in this study.

1. INTRODUCTION

During the past decade, the usage of different electronic applications for communication has seen a notable rise. Especially the Internet has gained popularity and the usage of e-mail has become very wide. In some countries, nearly half the population uses e-mail for their communication [7] and the percentage is on a constant rise. E-mail is in use both as a means for personal communication and within business. In business use, an estimated 50% raise in e-mail message volumes during the year 2002 in the USA has been reported [8]. The huge volumes have resulted in a situation where some companies already constrain their employers' Internet use [1]. In the future, the problem of information overload with e-mails will be an even vaster issue than what it currently is, and tools for helping in mastering them would be valuable. Specially in the future when e-mail messages can be read through mobile phones and important messages should be easily detected having only the very limited display area available, these questions become acute.

At the present, there exist some means for mastering the task of mail organizing. In some applications, such as the Microsoft Outlook [5], there are rule-based mail classifiers that the user can tailor for this purpose. An example of rule-based mail organizing is having all mails coming from a particular person arrive automatically in an assigned folder. Such functions have also been adapted for mobile use in some coverage [6]. Without such tools e-mails have to be manually organized in folders, that is, the user specifically selects on every occasion if he/she wants to put the mail in a particular folder.

Still, the present means of organizing e-mail messages are not satisfactory in all respects. First, they demand a huge user effort. It may take hours to create appropriate rules that could automatically classify an arriving mail into a correct place. Secondly, it is necessary that the

user has an understanding of what kind of mails he/she will get in the future, which cannot in principle be known. If the user starts getting e-mails that are of a new subject, the old rules do not apply any more, and continuous updating of the rule base is needed. Another flaw with the present means of organizing e-mails is that the results are very *granulated*: The mails, in whatever way they are organized, are organized in discrete, separate folders in a binary way – either the mail belongs to a class or it does not.

In Nokia Research Center the problem of automatically organizing e-mail messages was explored during the year 2001. The organizer was aimed at improving mail organizing in the following respects:

1. The user would not need to create the folders manually him/herself.
2. No beforehand knowledge about which folder each mail should belong to is required from the user.
3. The mail organization would be easily adaptable with change in mail contents.
4. The mail classification should be robust and – in some sense – *fuzzy*.
5. Finally, the ordering should be easily comprehensible to the system user.

It seems that the promises given by SOM enthusiasts promoting the idea of *self-organizing maps* [3] would be a key to all of the above problems: a self-organizing map (SOM) can visualize the "semantic space" of textual documents in a very understandable form [2][4].

However, it is clear that the traditional application areas where SOM has been used for content analysis (patent databases, news groups, etc.) are easier than what the current one is: E-mail messages are typically very short, and the messages are plagued by typing errors. In this paper, the promises of the SOM idea are put in practical test: Is SOM really beneficial in a non-ideal environment? The developed experimental application is here referred to as the *mail organizer*.

2. MAIL ORGANIZER

The algorithm that produced the mail organization in the application was based on a two-level process: First, a linguistic analysis of the mails and their contents was carried out, and after that, the mails were all fed into a self-organizing map. The final result was shown as a two-dimensional graph.

The linguistic analysis was run for each mail message separately, producing an index of its key terms and estimate about the relevance of each term. In the analysis, diverse knowledge about language structure, word frequencies and grammatical rules was taken advantage of. Too common and too rare words (probable typos) were rejected. In Table 1, an excerpt of one e-mail message and the term index given for it are presented. The numeric estimation of term importance has a value between 0 and 1, a higher value meaning a more important term according to the linguistic analysis. In the analysis, only nouns were analyzed. Because of the peculiarities of the Finnish language, in the English translation not all of them are nouns.

The list of key terms characterizes the message, defining a kind of "fingerprint" for it. Based on the key terms thus found, the mails can be organized. For further processing, the message fingerprints are coded in vector form, that is, each term has a unique entry index in the sparsely coded term vectors. If the message contains a term, the corresponding entry in the fingerprint vector is non-zero. In the current experiments, it is the term relevance values as given by the linguistic analysis that are used in the fingerprint to emphasize their assumed significance for characterizing the message contents.

Every key term that is common to more than one mail creates overlap in the mail fingerprints, and this redundancy can be utilized for modeling of the similarity between the

messages. In this experiment, the self-organizing map (SOM) was used for modeling the messages. The self-organizing map is a data clustering method that creates ordering between the clusters, so that nearby clusters finally contain data that are "near" each other in the data space.

Table 1. The linguistic analysis of the mail contents. The key terms in the right column are extracted from the mail's body text with linguistic means.

An excerpt of the original e-mail (translated from Finnish)	**Assumed key terms**
Hello! We selected the students of the usability school for the Technical University's quota today. It was tricky because there were more applicants than we could take in and all of them seemed capable. Congratulations for you who won! A major study right: Cathy Morgan Niels Deneuve Joan Smith Jacob Eastwood A minor study right: Mark Gabriel There will be plenty of paper work and forms to be filled in at the beginning of the usability school. (continues)	usability school student (0.3766); filling in forms (0.3757); Technical University's quota (0.3746); usability school (0.2890); paper work (0.2624); (continues)

When such features like terms are used as data, one can only hope that the organization between clusters happens based on the semantic contents of the messages. This kind of approach to document modeling has been shown to be successful also in very complicated domains; in this experiment, the goal was to test how well the ideas can be implemented in a real-life environment, where the quality of the input data cannot be very well controlled. Does the SOM work in the intuitively correct way even if the input is not optimally conditioned?

The end result of the analysis is a two-dimensional map of the mails. Each mail is shown as a separate entity and the more there have been common key terms in any given two mails, the closer they should be to each other. In the same manner, if any two mails do not have many common key terms, they should be farther from each other. With a mail that has not any key terms that are common to some other mail, the placement in the two-dimensional graph is random.

The two-dimensional map graph changes according to the changes in the mail contents, but once the organization is formed, the changes will be gradual and sliding rather than steplike and abrupt. In Figure 1 one example with real news data visualizes the way the messages are shown in the two-dimensional graph, where each number represents one mail message. To reach better readability of the map, the messages were not shown exactly where the appropriate best matching nodes (e.g. the winning neurons) were located, but the locations of the neighbouring nodes were also taken into account. The plotting algorithm is given in the following section.

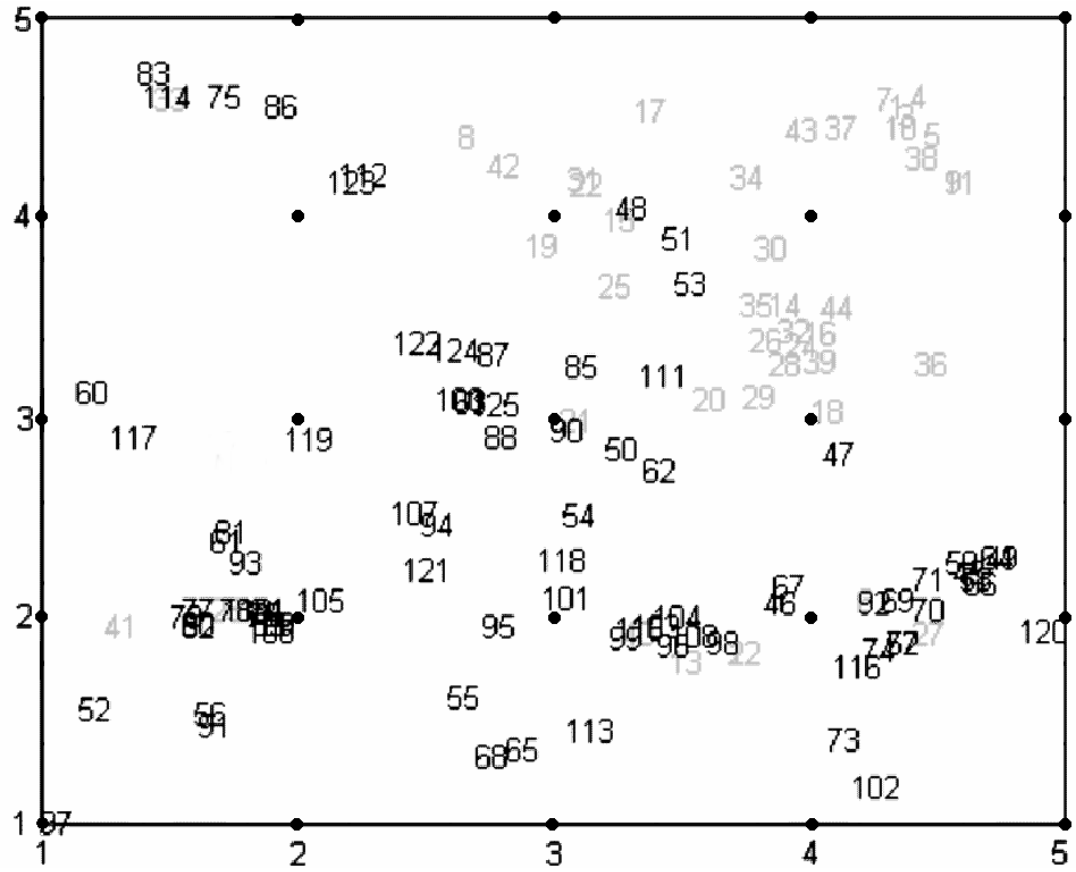


Figure 1. The two-dimensional graph of 124 news messages from the news groups `soc.culture.nordic` *and* `soc.history.medieval`*. There seems to exist ordering between messages: The ones marked grey mainly reside in the upper right corner, even though there are some outliers. This result was obtained using a 5x5 grid. The prototypes have been plotted as small solid circles (●).*

In the Figure 1, the totality of 124 mail messages, coming from two different news groups, is shown. The details of the plotting algorithm are discussed in the following section. The hypothesis is that the mails should be quite well grouped according to the news groups – assuming that the map has been capable of capturing this "semantic space". At least for this input material the approach seems to give relevant results – but this data was still "easy".

3. PERFORMANCE OF THE MAIL ORGANIZER

The mail organizer was tested with a test subject's real inbox mail data (287 messages from thirteen folders) and with different settings in the algorithm (varying the number of index terms extracted from the messages; making the terms more or less general; using different kinds of index term weighting strategies, etc.). In this case the correct folders were known, and the goal was to see if the algorithm could automatically find something similar-looking.

Our study was targeted to work as an acid test for evaluation of SOM's clusterization capabilities in natural, informal language. Therefore we started by using the same data in teaching and testing. This test condition gives naturally overly optimistic results, but this is justified by the remark that only if the classifier can act moderately within these constraints there is hope that it would work well also with previously unseen mails, which is the case in real use.

3.1. How data was arranged on the screen

The modelling of the textual documents was carried out using SOM with a two-dimensional grid. The map was taught with all the mails' fingerprints, with 10 000 batch-mode iterations. The initial weights in the node vectors were drawn from a [0,1] uniform distribution.

As was said in the previous section, the mails were not plotted exactly onto the position of the winning neuron, but instead so that neurons' representativeness was also taken into consideration. This alleviates partly the problem of plotting too many mails on top of each other on the screen.

Let **f** denote the fingerprint of a mail, that is, its feature vector constructed from the key terms found in the mail. Let x and y denote the grid positions of prototype nodes (neurons) as coordinate pairs (x,y). These values can be used as indices when specifying unique nodes in the map. In a 10x10 grid we thus have $x = 1,2,\ldots,N_x$ and y likewise, where $N_x=N_y=10$. Let $\mathbf{F}_{xy}$ be the prototype vector of the node at location (x,y) in the grid.

After being taught, the screen coordinates $(x(\mathbf{f}),y(\mathbf{f}))$ of each mail **f** were calculated as a weighted average of the node locations:

$$x(\mathbf{f}) = \frac{\sum_x^{N_x} \sum_y^{N_y} \mathbf{f}^T \mathbf{F}_{xy}\, x}{\sum_x^{N_x} \sum_y^{N_y} \mathbf{f}^T \mathbf{F}_{xy}} \quad (1)$$

$$y(\mathbf{f}) = \frac{\sum_x^{N_x} \sum_y^{N_y} \mathbf{f}^T \mathbf{F}_{xy}\, y}{\sum_x^{N_x} \sum_y^{N_y} \mathbf{f}^T \mathbf{F}_{xy}} \quad (2)$$

Here $\mathbf{f}^T\mathbf{F}_{xy}$ is the correlation of the fingerprint with prototype in (x,y). Multiplying it with x's and dividing the sum with the sum of correlations one gets the position in a continuous coordinate scale, as opposed to the discrete positions (x,y) of the nodes in the grid. Thus the mail may fall into the space *between* the nodes on the screen, not just over the winning node.

In the mapping given above, a problem is that each node's importance is weighted solely by its correlation with the fingerprint. That is, a node's contribution to the mail's position is not weighted by its distance from the actual winning node in the grid. This results as an averaging tendency to place every mail closer to the center of the on-screen map than to the borders, especially if there are multiple prototypes that have equal correlation with the fingerprint. In the plot in figure 1 this did not appear to be a problem. The interpretation is that there usually existed one prototype whose correlation was profoundly greater than that of others and thus the other neurons did not draw the mail remarkably towards the center. In figure 4 the averaging behaviour is more visible, however. Thus, a more sophisticated weighting scheme could be appropriate.

3.2. How performance was measured

The performance was measured using two metrics: *averaged within-class variance* and *mutual separability* of classes. Both of the metrics were calculated in the two-dimensional on-screen output space. Within-class variance measures how close to each other in the on-screen map are those mails that should belong together. This was measured by taking the average of the variances of the mails around the center points of their classes. Let $\mathbf{x}(\mathbf{f}) = [\, x(\mathbf{f}) \,,\, y(\mathbf{f}) \,]^T$ denote the position of the mail **f** on the screen as calculated in equations (1) and (2), and N_i be the number of mails belonging to class C_i . Then

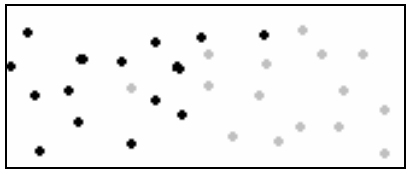
$$\text{averaged within-class variance} = \frac{1}{L} \sum_{i=1}^{L} \frac{1}{N_i - 1} \sum_{\mathbf{f} \,\epsilon\, C_i} \|\mathbf{c}_i - \mathbf{x}(\mathbf{f})\|^2 \quad (3)$$

where L is the number of folders (e.g. classes) and $\boldsymbol{c}_i$ are the mean points of classes C_i in the 2-D output space:

$$\mathbf{c}_i = \frac{1}{N_i} \sum_{\mathbf{f} \in C_i} \mathbf{x}(\mathbf{f}) \tag{4}$$

The Euclidean norm in (3) is taken in the output space. Figure 2 clarifies the idea behind the metric.

Good within-class variance



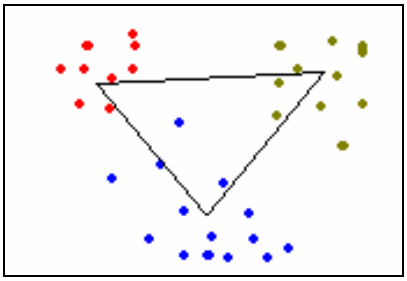
Bad within-class variance

Figure 2. Example plots on the screen of mails from two folders with good within-class variance and bad within-class variance.

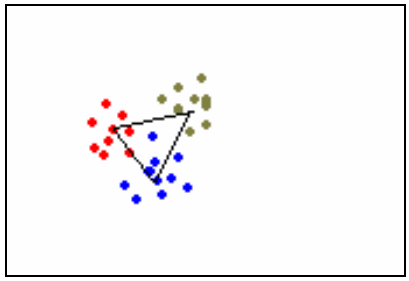
The other metric, separability, is used to find out how far are the mean points of the folders from each other. This was measured by summing together the mutual distances of mean points of mail folders and taking the average:

$$\text{separability} = \frac{1}{2L} \sum_{i=1}^{L} \sum_{j=1}^{L} \|\mathbf{c}_i - \mathbf{c}_j\| \tag{5}$$

A visualization of the metric is shown in Figure 3.



Good separability



Bad separability

Figure 3. Examples of mails from three different folders with good separability and bad separability.

From the metrics above, we see that the best results have a small within-class variance and big separability. Thus, as the linear discriminant analysis –inspired performance metric, we used the quotient of the two:

$$\text{performance} = \frac{\text{separability}}{\text{averaged within-class variance}} \tag{6}$$

3.2. Results

Figure 4 shows a result where the performance score reached one of the highest values in our evaluations (5.68). It is not a surprise that the different folders were mixed on the map – but even with the best result that were obtained in the evaluations, the mails that presumably had much in common did not fall very close with each other and the mails that were of totally different issues overlapped too much. There is clearly some degree of organization, but not enough when one thinks of professional e-mail usage: One misclassified message can ruin the user's trust in the system.

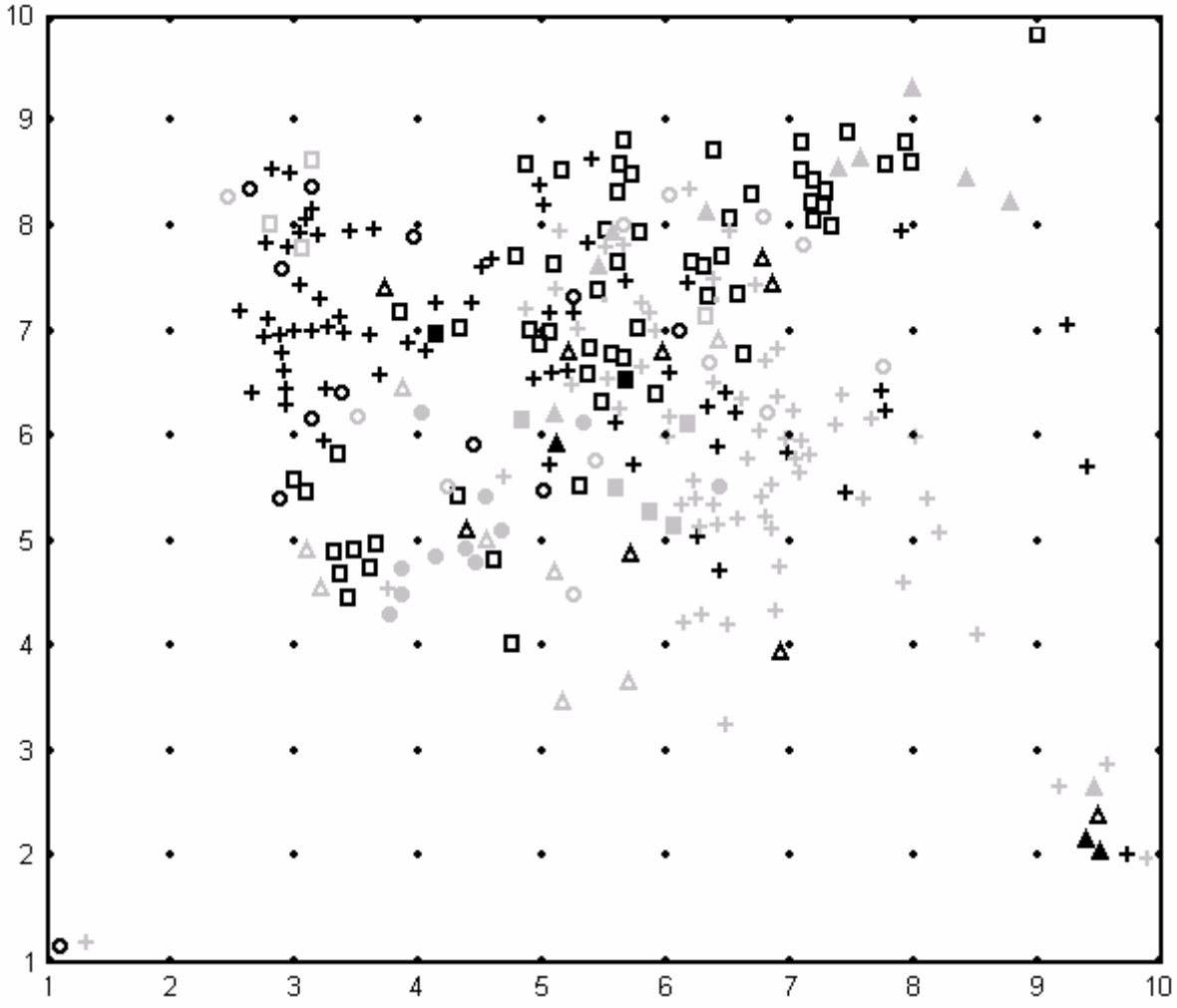


Figure 4. The two-dimensional graph of 287 e-mail messages from 13 different folders. The small solid circles (•) denote the positions of the prototype nodes in a 10x10 grid.

To be precise, the result gives too a positive impression from algorithm's performance, since the network was taught with the same data as it was tested. The reason for this arrangement was given in the beginning of section 3. In real use, the program cannot work that way, but instead it has to learn from the past mails to classify new mails. Then the results cannot be as good as above. Another problem was that the program performance depends very much of the initial values given to the weights in the network nodes.

There are also some technical problems what comes to possible implementation of such an organizer. The two-dimensional graph demands quite a big screen in order that the mail

totality is distinguishable. In practice, the application can only be used on a workstation computer; the present mobile devices and their displays are not competent to the space and color demands of this application.

The amount of the mails that can be organized with this application is limited. The largest amount of mails that the algorithm was tested with was 588 e-mail messages. Even with such moderate amount of data the mails get so overlapping in the graph that locating any single mail becomes difficult. On the other hand, with less than about 100 e-mail messages, there will not be enough overlapping key terms in the messages and the organization becomes too arbitrary.

Different kinds of data pre-processing strategies were tested to enhance the performance of the SOM algorithm. The basic dilemma here is that the map should organize by itself, and explicitly controlling the algorithm is somewhat questionable. However, in some cases the correct mail folders are known *a priori*, and guiding the algorithm towards reasonable results can be realized in different ways. First, in some experiments, the category information was explicitly added into the fingerprint vector, so that also this information affected the resulting map; second, the category information was used for determining optimal weighting for different terms, that is, if some term carries very much categorization information, it is weighted more in the fingerprints. However, even though the results became better, it seemed that still the messages could not be distinguished satisfactorily. These modifications were actually used to find out the capabilities of the classifier, and cannot be applied in real use when the correct folder information is not available.

4. CONCLUSIONS

During the summer 2001, we assessed an application for mail organizing that we had developed in the company Nokia, Finland. The idea of the application was that it would analyze the contents of every e-mail message and produce an indexed list about the key terms in the messages. Based on this, the terms were run in a self-organizing map analysis, which tracked the common key terms in the messages and produced a two-dimensional graph that would describe how much common there was in the mail contents and which mails were most similar with each other.

The mail organizer did not perform in an optimal manner: The mails that presumably had much in common, did not always fall close enough in the graphs and on the other hand, mails that subjectively were estimated to be on totally different topics, sometimes fell very close with each other.

There were many reasons for this. The mails were quite brief so that in the key term analysis, the average amount of key terms found per message was about 10. So, any one key term contributed quite much on the rest of the analysis. If two mails had nine non-common key terms and one term that was common to the two mails, this one common term could result in the mails being located near to each other in the final graph. The key terms that were extracted in the linguistic analysis were not on the whole on a common enough level (see Table 1). So there were not enough overlapping key terms in the mails, which made the function of the self-organizing map analysis unreliable.

Still, this application was designed for helping the user in organizing his/her mails as an alternative to both manual organization and rule-based, stable organizers as most of the applications nowadays are, for instance the Microsoft Outlook feature [4]. If the algorithm worked in a satisfactory way, the application could be used as a computer application for preliminary mail organizing; after that, the user could check the mails manually, and, for example, draw folder boundaries him/herself after the organization is visible in the two-

dimensional graph. After drawing the boundaries other pattern recognition approaches such as support vector machines could possibly be applied to realize the actual classifier.

SOM is a valuable tool when making complex data better graspable. It has been claimed – or at least this impression has not been actively refuted – that SOM would be a panacea for solving almost any high-dimensional modelling problem, also semantic ones. However, it should not be a surprise that the modelling results are very much dependent of the data properties. The old truth – "trash in, trash out" – still holds, even if this data were analyzed by SOM.

REFERENCES

[1] Conger, S., Loch, K. D. Ethics and computer use. *Communications of the ACM*, *38* (12), 1995.

[2] Kaski, S., Honkela, T., Lagus, K., Kohonen, T. WEBSOM – self-organizing maps of document collections. *Neurocomputing, 21* (1-3), 101-117, 1998.

[3] Kohonen, T. *Self-organizing maps.* Springer-Verlag, Berlin, 1995.

[4] Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J, Honkela, J., Paatero, V., Saarela, A. Self organization of a massive document collection. *IEEE transactions on neural networks, 11* (3), 574-585, 2000.

[5] Microsoft Outlook e-mail programme. Information available at <http://www.microsoft.com/office/outlook/>. [Cited 23 Sep 2002].

[6] NEC N503iS mobile phone. Information available at <http://www.javamobiles.com/docomo/n503is/>. [Cited 23 Sep 2002].

[7] Nurmela, J., Heinonen, R., Ollila, P., Virtanen, V. *Mobile phones and computer as parts of everyday life in Finland.* Phase II of the project "The Finns and the future information society", Report 1, Reviews 2000/5. Statistics Finland, Helsinki, 2000. Information available at <http://www.stat.fi/tk/yr/tietoyhteiskunta/suomalaiset_linkit_en.html>. [Cited 23 Sep 2002].

[8] Sampson, M., Ferris, D. *The corporate email market, 2000-2005.* Ferris Research, San Francisco, CA, 2001.