

Classification of Textual Data with Self-Organising Map

Neural Computing as Email Filtering Method

Tony Manninen, Raahe Laboratory, University of Oulu, email: tony.manninen@ratol.fi

Jani Pirkola, Nokia Mobile Phones, email: jani.pirkola@nmp.nokia.com

Esko Heiniemi, Per Brahe Laboratory, Raahe Institute of Computer Engineering, email: esko.heiniemi@ratol.fi

Abstract

In the experimental research work conducted at Raahe Laboratory of Oulu University the aim was to find and develop methods for classification of textual data with self-organising maps. The basis for the research is the need to find adequate solution(s) to solve the difficult problem of automatic email categorisation. The basic idea is to use computer and accompanying self-organising map to classify the emails to categories and thus provide ready-made filtering and forwarding tool for the task, which otherwise would be in the responsibility of the people sending mail. The aim of our research is to create methods and tools for automatic classification of abstract, open-ended, and thematically overlapping emails. The research paper contains background information about neural networks in general and self-organising maps in particular, as well as, a brief survey to the application domain. Furthermore, we describe our experiment with SOM_PAK self-organising map tool and analyse the results of the research.

1. Introduction

Electronic mail (email) seems to be even more popular form of communication in the near future. Especially large companies are receiving high amount of email from their customers. Customers do not usually want to struggle with numerous different email addresses, so companies tend to provide one general email address for most of the activities and correspondence. This, although practical solution for the customers, may cause major problems inside the company when considering mail exchange of thousands emails on a single day. Obviously that kind of information overload will be impossible to handle by a single human being.

The basis for our research is the need to find adequate solution(s) to solve the aforementioned problem. The basic idea is to use computer and accompanying self-organising map to classify the emails to categories and thus provide ready-made filtering and forwarding tool for the task, which otherwise would be in the responsibility of the customers (or companies). The computer-based classification and forwarding of emails with the proposed intelligent agent save the customers from trying to locate the correct person to send their mail to.

The aim of our research is to create methods and tools for automatic classification of abstract, open-ended, and thematically overlapping emails, in which the boundaries of different classes may be relatively vague. Several solutions from various fields of research have been presented to solve the problem of automatic classification of free-form textual data, but so far all of those seem to lack the needed flexibility and adaptability.

In this paper, a brief introduction to neural nets in general and self-organising maps (SOM) in particular are provided with a short discussion of existing solutions within the application domain. Furthermore, a description of our solution and corresponding experiments is provided with accompanying analysis of the results. The context of the work is limited to the development and evaluation of several SOM-based approaches to email classification problem.

Neural networks are biologically motivated and statistically based. They represent entirely different models from those related to traditional physical symbol systems. Instead of information being localised, the information is distributed throughout a network. Neural networks, or neural nets, are known for their ability to make rapid memory associations rather than for high-precision computational processing [Noy92].

Neural network applications can be described as “intelligent”. This is because neural nets adjust (i.e. they learn) to evolving conditions automatically. They provide means for tasks, which involve large amount of ambiguous data and dynamic environments. Furthermore, neural networks can be used to visualise massive amounts of data and to find coherent groups within the material [Koi94].

Generally speaking, learning is the process by which the neural network adapts itself to a stimulus, and eventually produces a desired response. Learning is also a continuous classification process of input stimuli; when a stimulus appears at the network, the network either recognises it or the network develops a new classification. When the actual output response is the same as the desired one, the network has completed the learning phase i.e. it has acquired knowledge. The learning of the neural network may be supervised, unsupervised, reinforced or competitive [Kar96].

Self-Organising Maps (SOM) are one application of neural networks. SOMs have been widely used for material classification and categorisation purposes especially within the cases of unknown classes. The basic Self-Organising Map can be visualised as a sheet-like neural-network array consisting of cells (or nodes). During the learning, these cells become specifically tuned to various input signal patterns or classes of patterns in an orderly fashion. The learning process is competitive and unsupervised, meaning that no teacher is needed to define the correct output for an input. In the basic version, only one map node (winner) at a time is activated corresponding to each input. The locations of the responses in the array tend to become ordered in the learning process as if some meaningful non-linear co-ordinate system for the different input features were being created over the network [Hon97].

Some properties that distinguish the SOM from the other neural networks are that it is numerical instead of symbolic, non-parametric, and capable of learning without supervision. The numerical nature of the method enables it to treat numerical statistical data naturally, and to represent graded relationships. Because the method does not require supervision and is non-parametric, i.e. no assumptions about the distribution of the data need to be made beforehand, it may even find quite unexpected structures from the selected data [Kas97]. Figure 1 illustrates the u-matrix presentation of a trained self-organising map.

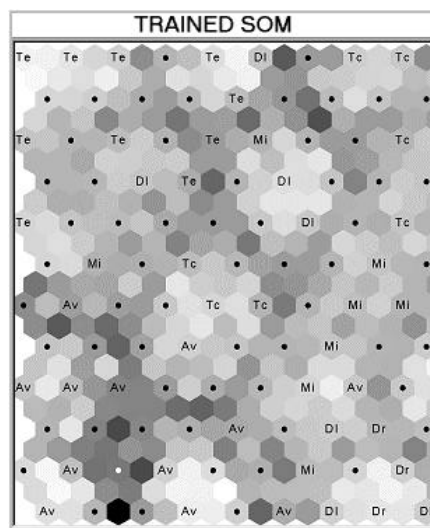


Figure 1. U-matrix presentation of SOM with some data clusters.

The basic idea in the SOM learning process is that, for each sample-input vector, the winner and the nodes in its neighbourhood are changed closer to in the input data space. During the learning process, individual changes may be contradictory, but the net outcome in the process is that ordered values emerge over the array. If the number of available input samples is restricted, the samples must be presented reiteratively to the SOM algorithm [Hon97].

Detailed guidelines for how to actually compute the self-organising maps are given in the documentation of the public domain program package SOM_PAK [Koh95a]. The reference vectors of the map are first initialised to lie in an ordered configuration on the plane spanned by the two principal eigen-vectors of the data, and thereafter taught in a two-phase process. The learning starts with a wide neighbourhood kernel covering most of the map, and during the first phase the kernel quickly narrows close to its final width, at the same time becoming smaller in its peak amplitude. During the second, longer phase, the neighbourhood kernel continues from the narrower form and slowly shrinks to its final width and magnitude. The first phase enforces a global ordering of the map, while in the second phase the final accurate state of the map is formed gradually. The final neigh-

neighbourhood width determines the "stiffness" of the map, i.e., how closely the map will follow the local structures in the data [Kas97].

An **intelligent agent**, made from software, has, as can be told from its name, some degree of intelligence built in it. Furthermore, an Intelligent Agent is usually also required to learn from the experiences, and thus being able to adapt to the current conditions.

"An agent is a software thing that knows how to do things that you could probably do yourself if you had the time." Ted Selker of the IBM Almaden Research Center

Exact definition of an Intelligent Agent is still under discussion. However, some concepts have been widely accepted as properties of an agent:

- **Autonomy:** Agents control their actions themselves
- **Social ability:** Agents interact with other entities (agents and/or humans)
- **Reactivity:** Agents notice and react to changes in their environment
- **Proactivity:** Agents can also start doing things by taking the initiative
- **Continuity:** Agents are at least sleeping processes all the time
- **Goal orientation:** An agent is capable of handling high level tasks. [Her96]

One of the key application areas for intelligent agents will be information access and management. Mail and messaging applications may also be categories to be in this area [Her96]. Intelligent agents and their applications in the fields of pattern recognition and language processing have been discussed e.g. in [Mit97 and Hut97].

2. Survey of Existing Work

One possible solution to the presented classification problem could be the predefined string searches from an email. Unfortunately this method has some drawbacks, e.g. it can not learn new categories without human intervention, and in some cases it may be impossible to create categories just by searching certain strings included in the messages.

Natural language processing, possibly accomplished with the aid of rule-based AI system, is very complex and vast area of research. The applicable solution would require either too many resources, or, the implementing process should be completed with numerous shortcuts. The narrowed focus, and thus, very limited application domain result in difficult compromises when considering handling of free-form textual input from individual persons [Win92, Ric91, Rau96].

One additional solution to the classification problem would be the introduction of www-forms, which could have enough machine computable information in them so that the processing would be relatively easy and inexpensive. This, however, is not very flexible solution because forms do not necessarily have all the fields needed in every situation. If the form is made to be very general (e.g. Helpdesk form), it will have numerous open-ended text questions, and thus requires complex processing.

Other solutions, like the usage of evolution-based genetic algorithms [Ria97], and the utilisation of fuzzy function approximation [Mit97] have also been presented as possible solutions for the classification problem.

Text classification has already been experimented with neural nets. For example abstracts for WSOM'97 were classified using a self-organising map. In this approach the scope for the texts was quite limited and texts were also well formed, as they were abstracts. In this case, SOM training material was selected by hand [Lag97]. Furthermore, the WEBSOM research where thousands of news-articles from comp.ai.neural-nets were analysed using a SOM provides potential guidelines for the application domain. The work conducted by Helsinki University of Technology offers several applicable solutions in the field of classification of free-form textual data [Web98]. Further information on SOMs may be found from [Koh95b].

Smart Software Company has conducted research about coding open-end questions from survey responses. Their approach was to make training data for neural network partly by hand (semi-automatic). This process involved mostly marking synonyms to be equal and removing unwanted or unique words [Rau96].

Rialle et. al have been researching semiotics and modelling computer classification of text with genetic algorithms. In their work they have implemented computer technology that offers readers assistance in attaining some aspects of the informational or semiotic content of a text (discursive, lexical, hypertextual, thematic, stylistic, etc.). The system (CARAT) is an assistant in reading and analysing texts [Ria97].

3. Experiment

In our experiment we tried to demonstrate, how a self-organising map could categorise a selected set of emails received by the author. The focus of the work was on the automated procedures leading to a situation where it was not practical for us to do anything manually, and thus it was reasonable to use separate algorithms (scripts) to process the data. This ensured that every step was repeatable and possible to achieve by using software only without any intervention required from a human operator.

The experiment concentrated on methods and algorithms that could be used as part of an Intelligent Help-desk Agent (Figure 2). This agent would replace the human operator in email classification and forwarding process, and thus would provide clear advantages when compared to current situations.

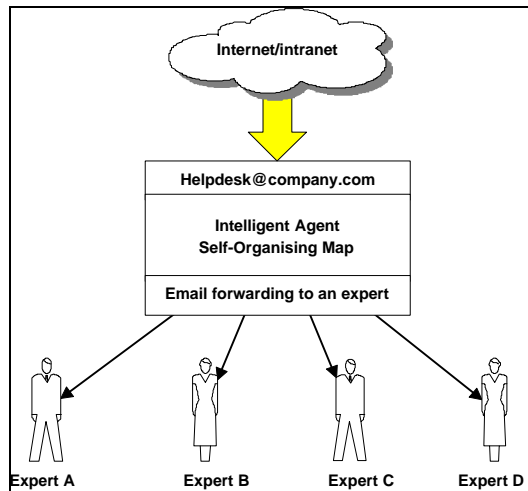


Figure 2. Automatic email forwarding to corresponding expert.

The main objective of the experiment was to create methods for automatic email categorisation based on the "semantics" of the messages. Other as well important target was autonomous formation of the training material so that human intervention would not be needed in the designed system.

The classification of textual data using Self-Organising Maps requires several pre-processing phases to be completed before the abstract material can be used to train the map. After several planning iterations, a three-phase pre-processing procedure was designed by the authors. The procedure, as also described by [Hon97] as the general procedure to be used with vector-space model for converting documents to vectors, is as follows:

1. Identify a vocabulary from the document collection.
2. Delete some common words using a stop list and remove the most and least frequently occurring terms.
3. Index the document collection on the basis of the remaining list. The components of a document vector can consist of:
 - (a) binary digits - the value depends on whether the corresponding term appears in the document or not;
 - (b) weights based on the frequency of occurrence of the term in the document;
 - (c) weights based on the term frequency and the inverse document frequency. The inverse document frequency is the inverse of the number of documents in which the term occurs.

Figure 3 illustrates the material pre-processing phase, which is in accordance with [Hon97].

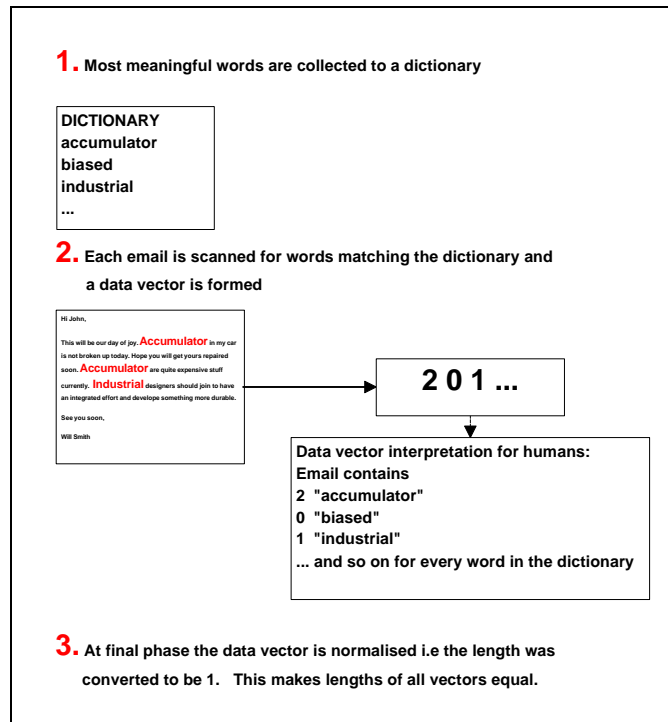


Figure 3. Pre-processing of the training material.

As a basis for the research material, there were total of 179 emails collected from the emails received by the author. These emails, written in Finnish, were from 6 different manually categorised classes, with one of the classes containing miscellaneous emails. The emails, already categorised semantically to different classes, provided us with a possibility to compare the results of the SOM (computer-operated) to our own categorisation (human-operated). The amount of emails in each class was decided to be approximately 30. With the similar email quantities, none of the classes could drop out because of the indifferences in material amounts.

The demonstration data was not the easiest possible to classify. Emails were collected from real, received emails. Some of the classes were semantically and thematically very close to each other and we even had one class for miscellaneous emails, which did not belong to any of the other 5 classes. The difficult and complex training data was selected because there was a need to have real-life situation introduced to the SOM and so to have as realistic results as possible. The 6 email classes and the corresponding labels used in data files were: Internal mailing list of RATOL (Te), Teaching (Tc), MSc studies (Di), Project Dreamscape (Dr), Aviation (Av), and Miscellaneous (Mi).

The most important phase of preparing the training material from textual data is the determination of numerical values (or vectors) for data items. Despite the drawbacks described by [Hon97], the vector-based approach was selected to create numerical data from the considerably large set of words. The components of the vectors to be formed were indices corresponding to the order of the words. When a word existed in the email, the component was set to the non-zero value, whereas the missing words resulted components with value "0". This method, however, is only practicable in small experiments. With a vocabulary picked from an even reasonably large set of material the dimensionality of the vectors would become intolerably high. Even if the processing of high-dimensional vectors could be possible with the current systems, there is another drawback occurring from large dictionary: most of the words are unique and as such can not be effectively used for classification purposes.

The first phase of the pre-processing consisted of word extraction and dictionary creation. In the beginning, a list of unique strings that occurred in emails was created. There were about 13000 unique strings in the emails, thus yielding too large dictionary. The amount of words was cut down by leaving out words, which were shorter than 3 letters, and by leaving out words, which were longer than 20 letters. The process was further continued by converting all the words to uppercase (capital letters), so that different or mixed cases did not count as separate words. All these steps reduced the amount of words relatively efficiently (from 13000 to 7000). At this

phase there still were thousands of unique strings in the dictionary. The decision was made to cut down the amount of words by stripping out words which appeared too frequently (like “and”) or were too unique to help in classification. The extraction was conducted with the aid of Perl-script containing frequent breakdown algorithm. When the size of the dictionary was reduced to a practical level (approximately 680 words), the training data was then formed from the dictionary and email data file according to the procedure described earlier.

The training of the neural net is usually the most automated work phase because of the large amount of iterations and algorithm calculations processed by the computer and corresponding software. In our research, the training of the Self-Organising Map (SOM) was conducted on the Unix platform with the SOM_PAK software provided by Helsinki University of Technology [Koh95a].

The overall training process of SOM can be divided into four separate phases: initialisation, training iterations, calibration and visualisation. Considering the visuals, in our research we concentrated only on the u-matrix presentation of the maps. This presentation visualises the distances between reference vectors of neighbouring map units using grey levels.

In order to get accurate and realistic results about the success rate of the maps, a series of tests was conducted. The tests were used to evaluate the behaviour of the maps in scenarios where new and non-deterministic data is introduced to the trained SOM. The aim of the testing was to find reliable enough indications of correct mapping with the aid of pre-defined test case.

The test data consists of 30 new emails from the original classes (6 classes, 5 emails from each), and of 20 new emails from additional 5 categories (4 emails from each). The external categories were selected from the author's email storage. In order to increase the reliability of the test, each category was already compiled before the research, and the emails were randomly selected from these. The classes and abbreviations of test data are: RATOL internal mailing list (TTe), Teaching (TTc), MSc studies (TDi), Project Dreamscape (TDr), Aviation (TAv), Miscellaneous (TMi), Protest project (TPr), Oulu University (TAs), Virtual Reality (TVr), Community (TYo), and Model Railroads (TPi).

In the test phase, the original dictionaries were used to create test data vectors. This corresponds to the scenario where static, or asynchronously updated dictionary file is used. After the test data vectors had been created and normalised, the trained map was calibrated with the new test data file, thus resulting a map containing information about the areas of test emails. The visualisation of the map was then used as a basis for the analysis comparing the test map to the original map, as well as, to the map from first test phase.

4. Analysis

The first phase of the analysis concentrates on the results of the training in several cases. The introductory look to the maps is taken with a high-abstraction view to the visualisations of the trained maps. Although not very scientific, this more or less subjective view provides some general indications of the results. Furthermore, it serves as a basis for the comparison of the cases and indicates best solution. Figure 4 illustrates the seven training cases in sequential order.

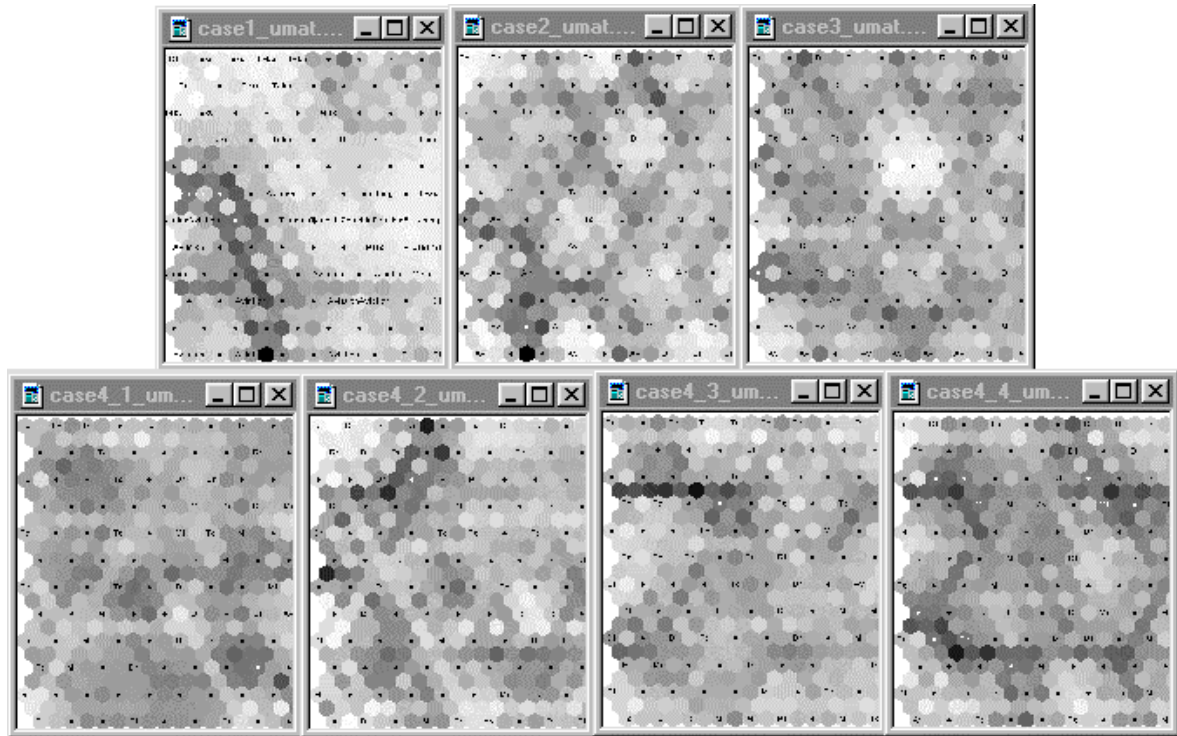


Figure 4. High-abstraction view to the trained SOMs.

Based on the visual and subjective evaluation made on the maps presented on Figure 4, a small comparative analysis is provided in the form of Table 1. The table lists all training cases, the sizes of dictionaries used in vector creation process, visual pattern evaluation, and cluster evaluation. The case 2 has been selected as an example case because of the best results provided by the training.

Table 1. Subjective evaluation of training results.

Case	Dictionary Words	Clusterisation Patterns (visual)	Clusterisation (classes)	Description
1	674	Poor	Good	Non-normalised vectors
2	674	Good	Good	Normalised vectors
3	77	Average	Average	Manual dictionary
4.1	668	Poor	Average	Email-based distribution
4.2	444	Poor	Average	Email-based distribution
4.3	523	Poor	Poor	Email-based distribution
4.4	532	Good	Poor	Email-based distribution

Case 1, although with non-normalised vectors not reliable, indicates relatively good clusterisation of the material. The length of the vectors being heterogeneous drives the training algorithm to pursue separation based not only for word distribution but also for quantities of separate words inside individual email.

Case 2, on the other hand, provides clear visual patterns indicating well-formed "terrain". Furthermore, the clusterisation is even better than in the case 1. Figure 5 illustrates the trained SOM of case 2 with the possible clusters outlined. All of the six categories seem to fall into relatively clear areas. The teaching (Tc) class has been divided into two separate clusters with the MSC studies related cluster in between them. Furthermore, there seem to be minor exceptions in the consistency of the clusters. For example, Te cluster contains also small occurrences of Mi and Di class emails. The nodes with inconsistent material have been named according to the class, which has most vectors falling on that particular cell. Thus, the cells labelled with Mi and Di contain also some amount of Te class emails.

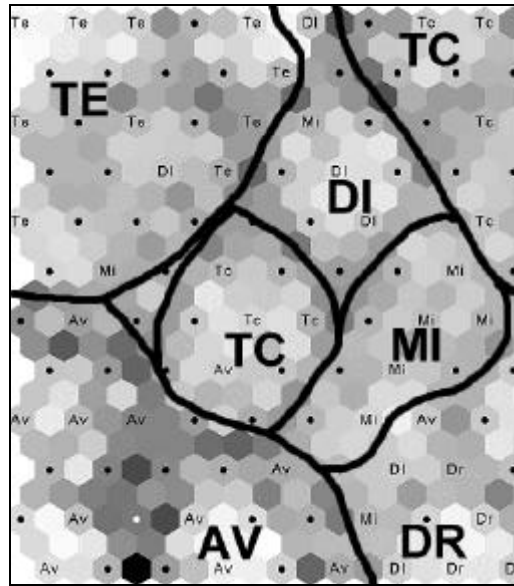


Figure 5. Trained map with possible clusters outlined.

Case 3 is comparable with case 2 with the major difference being the "blurred" terrain formation on case 3. Otherwise, the similarity of the pattern is clearly visible on the high-abstraction view image. The reason for terrain "blurring", as well as, the cause for the inferior results in case 3 can be explained with the sample rate theory. As described in the theory of telecommunications, the poorer the sample rate is, the more data will be lost during sampling. When trying to reconstruct the image by taking small samples from it, different sample rates would result in relatively different image. In this example the areas would be more or less similar, but with the poorer sample rate, the area of "common ground" would be hazier.

The sample rate (number of words in dictionary) is thus affecting strongly to the outcome of the training. If the dictionary is too small, there will be high amount of information loss. Some of the clusters may not be formed by the SOM, because there are not enough differences (or similarities) in the reference vectors.

Case 4 is good example of the sample rate scaling. When going from case 4.1 to 4.4 there are clear changes in the overall patterns of the map. The edges become clearer (darker nodes) and the whole terrain formation evolves to a new configuration. The case 4.1 is somewhat exceptional because the dictionary, although large, is combined from words with smaller amount of distribution (i.e. words found in 4 to 16 emails are included). The cases 4.2, 4.3 and 4.4 show the tendency of increased dictionary. Even if the increment is relatively small, the effects seem to be drastic. This emphasises the criticality of data pre-processing; careful planning and research should be conducted when selecting, for example, the frequency limits for the dictionary words.

The overall task to visually analyse the self-organised maps is difficult to do because there are too much room for subjective views. In the literature of the field there are some methods for systematic and more objective analysis and comparison of the maps [Kas97]. The visualisation of u-matrixes should be done in a 3D form in order to make the concept of distances and clusters more clear to the viewer.

The analysis of the testing phase consists of presenting and evaluating the distribution of separate test data on top of the trained map. The testing process represents a situation where new material is introduced to the map for categorisation purposes i.e. no training or learning is conducted at this point. Figure 6 represents the test data distribution on top of the separate clusters on the map.

- Linking the SOM approach to the LVQ method and exploring the advantages of each
- Developing the Pathfinder algorithm to map the test emails to the closest cluster "head"

6. Conclusions

Based on the results of the research conducted with difficult and challenging training material (free-form textual messages written in Finnish), the self-organised maps seem to be potential method for categorisation of free-form textual material like emails. The results indicate numerous possibilities for further development work, which would eventually lead to fully functioning application.

Although not perfect, the results were relatively successful and consistent. The fact that the material and context of the research were selected to be almost too challenging increases the value of the results even higher.

The data pre-processing seems to be the most critical phase when considering projects of the field. The seemingly automatic training phase requires solid and consistent preparations usually achievable only through manual work. Small changes in the training material could lead in totally different results of the maps.

Based on the research, adaptive and incrementally trained SOM would be ideal solution for the problem. The core structure and functionality of the methods and tools have been tested, but still there are many open questions waiting to be answered.

References

- [Noy92] James L Noyes, "Artificial Intelligence with Common Lisp - Fundamental of Symbolic and Numeric Processing", D C Heath and Company, 1992
- [Koi94] Pasi Koikkalainen (ed.), "Neurolaskennan Mahdollisuudet (in Finnish)", TEKES, 1994
- [Kar96] Stamatios V. Kartalopoulos, "Understanding Neural Networks and Fuzzy Logic", IEEE Press, 1996
- [Win92] Patrick Henry Winston, "Artificial Intelligence", Addison-Wesley, 1992
- [Hon97] Timo Honkela, "Self-Organizing Maps in Natural Language Processing", PhD thesis, Helsinki University of Technology, Finland, URL <http://www.cis.hut.fi/~tho/thesis/>, 1997
- [Kas97] Samuel Kaski, "Data exploration using self-organizing maps", Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82. DTech Thesis, Helsinki University of Technology, Finland. URL: <http://www.cis.hut.fi/~sami/thesis/>, 1997
- [Koh95a] Kohonen T, Hynninen J, Kangas J, Laaksonen J, "SOM_PAK - The Self-Organizing Map Program Package", Helsinki University of Technology, 1995
- [Her96] Björn Hermans, "Intelligent Software Agents on the Internet: an inventory of currently offered functionality in the information society & a prediction of (near-)future developments", Tilburg University, 1996
- [Mit97] Sanya Mitaim, Bart Kosko, "Fuzzy Function Approximation and Intelligent Agents", Proceedings of the SPIE, vol. 3165, pp.2-13, 1997
- [Hut97] William R. Hutchison, "Learned Emergence of Functional Symbol Systems in Adaptive Autonomous Agents, Guided by Skinner's Analysis of Verbal Behavior", Proceedings of ISAS'97, pp. 287-92, 1997
- [Ric91] Elaine Rich, Kevin Knight, "Artificial Intelligence", McGraw-Hill, 1991
- [Rau96] Raymond Raud, Michael Fallig, "Automating the Coding Process with Neural Networks", Smart Software Company, URL: <http://SmartSoftwareCompany.com/Autocode.html> (also in <http://dimm.ratol.fi/dimm/neuro/autocode/autocode.html>), 1996
- [Ria97] Vincent Rialle, Jean-Guy Meunier, Sofiane Oussedik, Georges Nault, "Semiotic and Modeling Computer Classification of Text with Genetic Algorithm: Analysis and first Results", Proceedings ISAS'97, pp. 325-30, 1997
- [Koh95b] Kohonen, T. (1995c) Self-Organizing Maps. Springer, Berlin.
- [Lag97] Krista Lagus, "Map of WSOM'97" Abstracts - Alternative Index", WSOM'97, 1997
- [Web98] Teuvo Kohonen, Jukka Honkela, Samuel Kaski, Krista Lagus, Vesa Paatero, Antti Saarela, Jarkko Salojärvi, "WEBSOM - Self-Organizing Maps for Internet Exploration", URL: <http://websom.hut.fi/websom/>, 1998