# Can Automatic Personal Categorization deal with User Inconsistency?

**Abstract.** Document categorization is a daily task in every organization, but it is a very subjective process. While automatic document categorization has been widely studied, much challenging research still remains, to support user subjective categorization. This study evaluates and compares the application of Self-Organizing Maps (SOM) and Learning Vector Quantization (LVQ) to automatic document classification according to a subjectively predefined set of clusters in a specific domain, and assesses the effect of user inconsistency on this process. We used a set of documents from an organization that was manually clustered by a domain expert. Results show that despite the subjective nature of human categorization, automatic document clustering methods correlate well subjective, personal clustering, with the LVQ method giving better results than the SOM. Users reclassified the documents that were misclassified by the system. The reclassification process revealed an interesting pattern: about one third of the documents were classified according to their original classification, about one third according to the system's suggestions and one third got a different, new classification. Based on these results we conclude that automatic support for subjective categorization is feasible, and it can represent user inconsistent preferences with a precision level of up to 86%.

**Keywords:** Subjective text categorization, text categorization with SOM and LVQ

## 1 Introduction

### 1.1 Motivation

For years, manual categorization methods have been defined and employed in libraries and other document repositories according to human judgment. Organizations search for information and save it in categories in a way that is meaningful to them. On the other hand, information users define subjective topic trees or categories based on personal preferences, and assign documents they read or write to categories according to this subjective definition. Obviously, the categories used by information users are idiosyncratic to the specific user. A major problem in generating an automatic, adaptable system is to determine to what extent an automatic system can reflect the subjective user's viewpoint, regarding his/her domain of interest. Users are usually inconsistent about their classifications (Dawes, 1979) and have a tendency to change the document classification they use over time. In

addition, they may find a document relevant to more than one category, usually choosing just one to host the document. Our question then is how can an automatic clustering system "guess" the user's subjective classification?

The motivation for the current study was to evaluate the possible use of automatic categorization techniques to support manual categorization in a company that performs this task on a daily basis by human experts. In our experiment the system performed a task similar to one of human categorizers, based on his data. In this study we did not use existing well-defined sets of categorized documents, like Routers or TREC, as they do not reflect the specific subjective user's point of view). Rather, we used a set of manually categorized financial news, and trained Self-organizing maps (SOM) and Learning Vector Quantization (LVQ) Artificial Neural Nets (ANN) to automatically categorize them. Then we measured the difference between the automatically generated set of clusters (or categories) and the pre-defined manual clusters. Our assumption is that if there is a small difference between them, we can use a trained ANN to automatically cluster the user's documents.

The study specifically addresses the following questions:

1. To what extent can automatic categorization represent personal, subjective user categorization?
2. What is the effect of the training set size on automatic categorization performance?
3. What is the difference between supervised (LVQ) and unsupervised (SOM) training on the above questions?
4. How does user inconsistency affect automatic clustering results?

### 1.2 Background

Clustering is defined as unsupervised classification of patterns into groups. A wide variety of clustering methods have been presented and applied in a variety of domains, such as image segmentation, object and character recognition, data mining, and information retrieval (Jain et al., 1999). One well-known clustering method implements Artificial Neural Nets (ANN).

ANN is a network of simple mathematical "neurons" that are connected by weighted links. The ANN is trained by adjusting the weights between the "neurons". Information retrieval and information filtering are among the various applications where ANN has been successfully tested (Boger et al., 2000).

There are two main branches of ANN which are distinguished by whether their training method is supervised or unsupervised:

1. The supervised ANN uses a "teacher" to train the model. An error is defined as the difference between the model outputs and the known (expected) outputs. The error back-propagation algorithm adjusts the model connection-weights to decrease the error by repeated presentation of inputs.
2. The unsupervised ANN tries to find clusters of similar inputs when no previous knowledge exists about the number of the desired clusters.

In both cases, once the ANN is trained, it is verified by analyzing inputs not used for the training (a test set).

The self-organizing map (SOM), a specific kind of ANN, is a tool that is used for the purpose of automatic document categorization (Honkela el at., 1997, Kohonen, 1997, Jain et al., 1999). The SOM is an unsupervised competitive ANN that transforms highly dimensional data to a two-dimensional grid, while preserving the data topology by mapping similar data items to the same cell on the grid (or to neighboring cells). A typical SOM is made up of a vector of nodes for input, an array of nodes as output map, and a matrix of connections between each output unit and all the input units. Thus, each vector of the input dimension can be mapped to a specific unit on a two-dimensional map. In our case, each vector represents a document, while the output unit represents the category that the document is assigned to.

The learning vector quantization (LVQ) algorithm is a supervised competitive ANN that is closely related to the SOM algorithm. Like the SOM, the LVQ transforms high dimensional data to a two-dimensional grid, but without taking into account data topology. To facilitate the two- dimensional transformation, LVQ uses pre-assigned cluster labels to data items, thus minimizing the average expected misclassification probability. However, unlike the SOM, where clusters are generated automatically based on item similarities, the clusters are predefined. In our case, the cluster labels represent the subjective categorization of the various documents supplied by the user. LVQ training is somewhat similar to SOM training, but it requires that each output unit receive a cluster label a priori to training (Kohonen, 1997).

To use a clustering mechanism, such as an ANN-based approach, an appropriate document representation is required. One popular model is the vector space model in which a document is represented by a weighted vector of terms (Baeza-Yates and Ribiero-Neto, 1999). This model suggests that a document may be represented by all meaningful terms included in it. A weight assigned to a term represents the relative importance of that term. One common approach for term weighting is TF ("term frequency") where each term is assigned a weight according to its frequency in the document (Salton and McGill, 1983). Sometimes document relations are also considered by

combining TF weighing with IDF ("inverse document frequency") into TF*IDF weights. Using the SOM algorithm relieves the necessity of document relations weighting since it already takes this combination into account during the training phase. In this study, we adopted the TF method in order to define meaningful terms for document representations for both SOM and LVQ processing.

### 1.3 Related work

The need for personalized categorization has been felt for quite some time now, and a great deal of work has been done in various application areas. For example, mail filing assistants, such as the MailCat system (Segal and Kephart, 1999), proposes folders for email that are similar to the categories or labeling processes performed by the ANN. MailCat employs the TF*IDF algorithm where every folder has a centroid representative vector. The MailCat system provides the user with categories (folders) from which he chooses the one he deems most appropriate. Those authors did not require that the classifier be based upon TF-IDF, but they underlined the importance of other factors, such as reasonable accuracy, incremental learning, and ranking of several possible categories.

Clustering methods can define similar groups of documents among huge collections. When clustering is employed on the result of a search engine, it may enhance and ease browsing and selecting relevant information. Zamir & Etzioni (1998) evaluated various clustering algorithms and showed precision improvement of the initial search results, which varied from 0.1- 0.4 to 0.4 – 0.7. Rauber & Merkl (1999) showed that clustering which is applied to general collections of documents could result in an ordered collection of documents grouped into similar groups that is easily browsable by users. Other studies, that resembles the "one button" (one category) implementation of MailCat reported similar precision results (Segal and Kephart, 1999).

The main difference between MailCat and the method we propose is that MailCat is based on specific algorithms developed to achieve "reasonable accuracy" in order to support and adapt specific user interests. In our case, we use a well-known categorization method to represent subjective, personal behavior, which may have interesting implications about the generalization of results to different domains.

The rest of the paper is structured as follows: Next we present the user model implemented in this study. Then we describe the first experiment performed for evaluating the clustering methods, followed by the experimental results. Afterwards, we present a second experiment, analyze the errors of the first experiment, and describe the effect of user inconsistency on subjective classification. We conclude with a discussion and suggestions for future work.

### 1.4 User Model Implementation

A major problem facing researchers is to decide what the user model should contain. It is obviously impossible (and probably unnecessary) to know everything about the user. Therefore, it is necessary to choose the most relevant and useful information (Chin, 1989).

Kass & Finin (1988) classified the information contained in the user model by four aspects: goals and plans, capabilities, attitudes, and knowledge and beliefs. Other studies considered also identity (gender, age, place of residence, marital situation, etc.); personal background (CV); traits (such as patience and sensitivity); preferences; physical condition; etc.

Another basic question that that needs to be considered when a user model is implemented is the technique for acquiring information about the user. There is much debate in the literature about whether a basic default model is required, or if it is possible, or even preferable, to begin with a tabula rasa (blank page) (Dix, 1997). Sukaviriya and Foley (1993) concluded that in order to facilitate interaction with users, some basic information about them is necessary. They suggest asking the user directly for the required information by means of a short questionnaire.

In the present study, the user model is represented by a trained ANN which is an implicit representation of the user's preferences. It represents the user's categorization preferences, e.g., it is limited to a specific context. The ANN requires no default model; it needs examples of user's classified information items without requiring any other information about the user. The user model is limited to the specific professional task and represents the specific categorization preferences of the user implicitly. The main advantage of the ANN approach is that it performs well in a wide variety of application areas. The main weaknesses of that approach are the lack of explicit knowledge in the model that allows the user to trace the decision process and the lack of adaptation as a result of new information. When enough new data is accumulated, the ANN requires re-training to update its model.

Our user model was built from a set of examples taken from a set of 1115 categorized messages provided by a user. These documents were categorized according to 15 categories. Using different quantities of training documents, several ANNs were trained to represent various sizes of training sets; hence several user-models were generated and compared. A specific ANN was built as follows: First, a set of training documents was selected. Then, the documents were processed, stop words were removed and the remaining terms were stemmed using the porter algorithm (Salton and Mcgill, 1983). This resulted in a list of terms that under went TF*IDF calculation that yielded a weighted vector of terms for each document. The set of vectors contained subsets of vectors for each category,

thus every subset of vectors represented an n-dimensioned space of that category, and collectively, represented the subjective user's categorization. The vectors were the input for the ANN training, thus, after training, the ANN interconnections weights (between the "neurons") represented the specific user's preferences and the ANN served as an implicit "user model".

No adaptation was implemented in the present study for several reasons. First, we tested a static data set and from this data set we draw the training set and the testing set. Second, the purpose of this study was to test the suitability of the clustering algorithms for implementing personal idiosyncratic classification. As explained the ANN runs once for the training set and once for the testing set. There is no adaptation mechanism within this algorithm; it just can classify at one step an entire testing set.

## 2 First Experiment

### 2.1 Method

#### 2.1.1 Data

The first experiment was performed in a company that deals with information extraction and categorization. Our purpose was to evaluate possible automation of data items categorization within the company's domain using actual, available data. We chose this environment in order to deal with real-life data in a real-life organization. Therefore, the data collection method resembled a field study. We did not ask the information expert to do an experimental task. Instead, we took a collection comprised of previously clustered items, which was built incrementally as a result of common working procedures. A data set containing 1115 economics and financial data items was used. The data was extracted from an online Internet based source (Yahoo.com). An information specialist read each item and manually clustered the items into one of 15 possible clusters. The size of the cluster varied from 1 to 125 documents per cluster. This approach represents a normal daily operation for information seeking and categorization.

#### 2.1.2 Experiment Design

The first experiment compared one manual classification method with two ANN automatic classification algorithms (SOM and LVQ) as independent variables.

In order to evaluate the ANN performance two dependent variables were used: precision and recall. Recall, in a specific category, is calculated as the number of documents that were automatically clustered correctly by the system, divided by the overall number of documents that belong to that cluster originally (given by the expert). Precision is the number of documents automatically clustered correctly, divided by the overall documents clustered by the system to the same category (correct + incorrect).

### 2.1.3 Procedure

All 1115 documents were analyzed, using a classical text analysis methodology. "Stop words" were removed. The resulting terms underwent further stemming processing using the Porter stemming algorithm (Frakes and Baeza-Yates, 1992). Finally, normalized weighted term-vectors were generated to represent the documents. In order to reduce the vector length, a low frequency threshold was implemented as a means of dimensionality reduction. Non-informative terms, which appeared in less than 100 documents (less than 10%), were excluded from the matrix. Then, the SOM and LVQ simulations were implemented using the SOM Toolbox for Matlab 5 (Vesanto et al. 1999).

Since experimental results may be influenced by the selection of the test and training sets, ten test sets were randomly selected. The size of each was 20% of the overall set (223 items). This size was selected for the test set and the initial training set in order to allow (when available) a sufficient number of examples for each original cluster to be used (an average of 15 documents from each cluster). For each test set, several training sets of different sizes were generated from the remaining 892 items by random selection so that the test set items were never part of the training set. Training sets sizes comprised 20%, 40%, 60% and 80% of the original data set (223, 446, 669, and 892 items). The experiment was repeated ten times for each algorithm, with a randomly pooled test set to preclude the occurrence of biasing effects resulting from the data sets selection.

The labeling process of the auto-generated clusters consisted of the following process: First, for each automatic cluster (output unit), we verified the original manual classification of each of the documents clustered. . Then, we checked the frequency of each manual category within each cluster. Last, the manual cluster with the highest frequency rendered its name to the automatic cluster (output unit).

### 2.1.4 Set-up

Maps of different sizes were generated to match the different training sets. The initial weights of each map were generated randomly just before the learning phase began. Two runs were performed, one for SOM and one for LVQ. Hence, slight differences were found in the randomly selected test and training sets. Table 1 presents document distribution among the ten sets; the columns represent the ten randomly drawn test sets and the rows represent the categories. The cells contents are the number of documents of every test category.

**Table 1.** Original data clusters

| **\ set cat ** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | 1 | 0\1 | | | | | |
| 2 | | 1\0 | 1\0 | | | | | | | |
| 3 | 0\1 | 0\1 | | | | 0\1 | | 1 | | 1\0 |
| 4 | 0\1 | 0\1 | 0\1 | 1 | 1 | 1 | 1 | 1 | 0\1 | 1 |
| 5 | 2\1 | 2 | 1\3 | 2 | 2\3 | 2 | 2\1 | 2\1 | 2\3 | 1 |
| 6 | 5 | 4\5 | 5\4 | 5 | 5\4 | 5\4 | 4\5 | 5 | 5\4 | 5 |
| 7 | 6 | 6 | 6 | 6 | 7\6 | 6 | 6 | 6 | 6 | 6 |
| 8 | 11 | 11 | 11 | 11 | 10\11 | 11 | 11 | 11 | 11 | 11 |
| 9 | 19\18 | 19\17 | 18\17 | 18 | 19\18 | 20\19 | 18\17 | 18 | 18 | 17\18 |
| 10 | 21\20 | 20 | 22\21 | 18\20 | 19\20 | 18\19 | 20\21 | 19\20 | 21\20 | 21 |
| 11 | 21 | 21 | 21 | 22\21 | 22\21 | 22\21 | 21 | 22 | 22\21 | 21 |
| 12 | 26\27 | 27\26 | 25 | 26 | 27\26 | 27 | 27\26 | 27 | 26 | 26 |
| 13 | 29 | 29 | 30 | 31\29 | 29\30 | 29\30 | 29 | 29 | 29 | 30\29 |
| 14 | 39 | 39\40 | 39\40 | 39 | 38\39 | 38\39 | 40\41 | 38 | 39\40 | 40 |
| 15 | 44 | 44 | 44 | 43\44 | 44\43 | 44 | 44 | 44 | 44 | 43\44 |

Note: Categories with a "\" indicate that a different number of documents appeared in the LVQ test set and the SOM test set. The number in the left hand side represents the LVQ test set while the number of the right hand side, represents the SOM test set.

## 2.2 Experiment Results

In order to evaluate the performance of the automatic classification of the SOM and the LVQ, we computed the average precision and recall on the ten different test sets.

**Table 2.** Precision and recall for LVQ with different training sets sizes

| Category | Training Set Size | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 80% | (892) | 60% | (669) | 40% | (446) | 20% | (223) |
| | **R** | **P** | **R** | **P** | **R** | **P** | **R** | **P** |
| 1* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0.46 | 0.97 | 0.26 | 1 | 0.20 | 1 | 0.04 | 1 |
| 7 | 0.77 | 0.59 | 0.80 | 0.58 | 0.63 | 0.57 | 0.30 | 0.44 |
| 8 | 0.95 | 0.81 | 0.94 | 0.74 | 0.93 | 0.71 | 0.81 | 0.69 |
| 9 | 0.77 | 0.88 | 0.72 | 0.84 | 0.71 | 0.88 | 0.69 | 0.77 |
| 10 | 0.52 | 0.59 | 0.60 | 0.59 | 0.60 | 0.57 | 0.59 | 0.53 |
| 11 | 0.70 | 0.51 | 0.62 | 0.53 | 0.60 | 0.46 | 0.51 | 0.44 |
| 12 | 0.78 | 0.68 | 0.74 | 0.65 | 0.73 | 0.64 | 0.68 | 0.59 |
| 13 | 0.69 | 0.81 | 0.69 | 0.83 | 0.65 | 0.82 | 0.65 | 0.76 |
| 14 | 0.81 | 0.84 | 0.81 | 0.82 | 0.80 | 0.83 | 0.80 | 0.82 |
| 15 | 0.79 | 0.84 | 0.82 | 0.83 | 0.79 | 0.78 | 0.75 | 0.74 |

* All marked categories contain a minimal number of documents and appears as specific categories only in part of the 10 randomly generated sets – see Table 1 for details.

### 2.2.1 LVQ results

Recall and precision for the LVQ are summarized in Table 2. The rows represent each of the initial 15 manually defined "original" categories, while the columns represent the four different sizes of training sets. Precision and recall results are denoted "P" and "R" respectively. Each cell contains the precision or recall for a specific category and a specific training set size. The precision and recall results are the average results of the ten runs. Several "empty" categories were found in the training/test sets (Table 1 summarizes the actual number of documents in each category, for each test set).

The results show that recall improves significantly as the size of the training set increases to 60% (669 documents). For the 80% training set size the results are mixed, meaning that several categories improved their precision and recall compared to the 60% training size, while others did not. This may indicate that at 60% enough data is accumulated to learn and adapt the ANN to the user's subjective classification. However, the recall of categories 1-6 is very low, ranging from 0 - 0.2 which may result from the small number of documents in these categories. This low recall implies that learning is not possible in categories where a training set is not available (or has a marginal size).

Precision improves as the size of the training set grows. Table 2 indicates that 80% training set size yielded the best results for most of the categories with the following exceptions: category 6, which did not have enough examples, category 11, where the slight decrease (-2%) from the 60% to the 80% training set can be related to the increase in recall (+8%) and category 13 where the precision is generally high from the 2nd training set (20%) and on.

**Table 3.** Precision and recall for SOM with different training sets sizes

| Category | Training Set Size | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 80% | (892) | 60% | (669) | 40% | (446) | 20% | (223) |
| | **R** | **P** | **R** | **P** | **R** | **P** | **R** | **P** |
| 1* | 0 | | 0 | | 0 | | 0 | |
| 2** | | | | | | | | |
| 3* | 0 | | 0.33 | | 0 | | 0 | |
| 4 | 0.50 | 0.53 | 0.50 | 0.61 | 0.20 | 0.42 | 0.10 | 0.25 |
| 5 | 0 | | 0.15 | 0.5 | 0 | 0 | 0 | |
| 6 | 0.39 | 0.71 | 0.22 | 0.77 | 0.23 | 0.56 | 0.17 | 0.25 |
| 7 | 0.47 | 0.45 | 0.35 | 0.51 | 0.35 | 0.49 | 0.40 | 0.45 |
| 8 | 0.92 | 0.74 | 0.88 | 0.69 | 0.79 | 0.67 | 0.73 | 0.68 |
| 9 | 0.68 | 0.87 | 0.67 | 0.86 | 0.68 | 0.86 | 0.52 | 0.83 |
| 10 | 0.78 | 0.56 | 0.79 | 0.53 | 0.75 | 0.52 | 0.72 | 0.45 |
| 11 | 0.53 | 0.61 | 0.50 | 0.58 | 0.47 | 0.61 | 0.41 | 0.50 |
| 12 | 0.70 | 0.66 | 0.69 | 0.57 | 0.64 | 0.53 | 0.60 | 0.50 |
| 13 | 0.66 | 0.80 | 0.66 | 0.78 | 0.62 | 0.77 | 0.57 | 0.66 |
| 14 | 0.79 | 0.85 | 0.76 | 0.90 | 0.76 | 0.91 | 0.76 | 0.85 |
| 15 | 0.79 | 0.72 | 0.74 | 0.71 | 0.70 | 0.71 | 0.67 | 0.69 |

* Marked categories appeared as categories only in part of the 10 randomly generated sets

** Such category was not defined by the SOM during the learning sessions

### 2.2.2 SOM results

Recall and precision results of the SOM are summarized in Table 3. There, each "original" category is represented by a row, while columns represent the four different sizes of the training sets. The precision and recall results for each training set are the average of the precision and recall for the ten runs. Like for the LVQ results, several "empty" categories were found in the training/test sets. Table 1 summarizes the actual number of documents in each category for each test set.

Table 3 shows that recall improves significantly as the size of the training set grow, with the exception of categories 1-5 that had nearly no documents. For the 80% training set size some of the results approximate the

results obtained at 60%, which may indicate that at 60% enough data has been accumulated and the ANN has succeeded to learn the user's specific preference.

Category 2 was not represented at all by the SOM, for some unknown reason. Category 11 included all documents that could not be assigned (by the human expert) to any category, hence generated a special category of items that was not well defined.

Table 3 shows that precision improve as the size of the training set grows. The 80% training set yielded the best results with the exception of categories 1-5 that had nearly no documents and categories 6 and 7 that contained a minimal number of documents. The decrease in category 14 may be related to the increase in recall.

### 2.2.3 Overall results and a comparison of SOM and LVQ methods

We computed an average recall and an average precision measures for each training set including all the 15 categories. The overall results for automatic clustering by SOM and LVQ are summarized in Table 4. LVQ average recall ranges from 49% to 60%, while SOM average recall of ranges from 45% to 57%. LVQ average precision ranges from 65% to 75% while SOM average precision ranges from 55% to 69%.

When there were enough training documents (a learning set of 892 documents), both methods yielded similar results (about 70% average precision and about 60% average recall). However, when the training set is smaller, as in the case of the 20% with 223 documents, supervised learning yielded better results (49% average recall using LVQ vs. 45% average recall using SOM, and 65% average precision using LVQ, but only 55% average precision using SOM).

**Table 4.** Average LVQ and SOM Precision and Recall results (all categories)

| Learning set | Percentage | 80% | 60% | 40% | 20% |
|---|---|---|---|---|---|
| | Actual size | 892 | 669 | 446 | 223 |
| **Measure** | **Method** | | | | |
| Recall | LVQ | 0.60 | 0.58 | 0.55 | 0.49 |
| | SOM | 0.57 | 0.54 | 0.50 | 0.45 |
| Precision | LVQ | 0.75 | 0.73 | 0.71 | 0.65 |
| | SOM | 0.69 | 0.63 | 0.60 | 0.55 |

As shown in Table 1, at some of the test-sets, several original categories contained a minimal number of documents, or no documents at all. Therefore, we calculated a new average precision and recall for those categories with more than 10 documents per category (categories 8-15) - see Table 5. It can be seen that recall improves

significantly for both LVQ (0.75) and SOM (0.73) while precision of SOM increases (0.73) and LVQ stays the same (0.75). This result supports the assumption that categories with only a few documents actually lower the overall level of precision and recall since the system could not be trained to cluster them.

**Table 5.** Average LVQ and SOM Precision and Recall results for categories with more than 10 documents.

| Learning set | Percentage | 80% | 60% | 40% | 20% |
|---|---|---|---|---|---|
| | Actual size | 892 | 669 | 446 | 223 |
| **Measure** | **Method** | | | | |
| Recall | LVQ | 0.75 | 0.74 | 0.73 | 0.69 |
| | SOM | 0.73 | 0.71 | 0.68 | 0.62 |
| Precision | LVQ | 0.75 | 0.73 | 0.71 | 0.67 |
| | SOM | 0.73 | 0.70 | 0.70 | 0.65 |

## 3. Second Experiment - User Re-evaluation

3.1 Method

### 3.1.1 Data

Since users are known to be inconsistent in their behavior (Dawes, 1979), in this experiment we examine the impact of user inconsistency on the automatic categorization. Explicit relevance feedback was requested from the user, with respect to those documents that were misclassified by the system. Since asking the user for feedback on all the data was unfeasible, we concentrated on the items wrongly classified by the system in the first experiment. We randomly chose one out of the ten datasets used with LVQ system and one for the SOM system, for which the training set size was 892 messages (80%), for re-classification. In those data sets, 62 messages were wrongly categorized by the LVQ method, and 68 messages were wrongly categorized by the SOM method. This was the collection presented to the expert for re-classification.

### 3.1.2 Experiment Design

The second experiment is a replica of the first experiment. Here too, we compared one manual classification method with two ANN automatic classification algorithms (SOM and LVQ) as independent variables. However, the data set implemented is the information items that were misclassified by the system in the first experiment.

The dependent variables used were precision and recall.

### 3.1.3 Procedure

In order to evaluate the system performance in light of user inconsistency, the following experiment was performed. The user was asked to reclassify the messages without knowing either the original categorization (that the user assigned in the past), or the system categorization. The same categories were used as in the first experiment. The user simply read the messages, and assigned them to one of the fifteen possible categories.

## 3.2 Experiment Results

The results of this experiment were quite interesting. For the LVQ, out of a total of 62 messages, 40.3% (25) were categorized to same original categories, 35.5% (22) were categorized to same categories assigned by the system and 24.2% (15) were assigned to totally different categories.

The same process was repeated for SOM with similar results. Out of the 68 messages, 44.1% (30) remained in original categories, 32.3% (22) were changed to system categorization and 23.6% (16) were assigned to new categories).

Taking the new categorization into account, we re-calculated precision and recall for both methods, for that specific test set.

### 3.2.1 SOM results

SOM-based results are summarized in table 6. Rows represent each of the initial 15 manually defined "original" categories, while columns represent the original and new categorizations for the selected set, with the re-classified items. An improvement is seen in most categories that contain more than a minimal number of documents and in most cases there are better recall and better precision. The range of recall increased from 0. - 0.83 to 0. - 0.96, while the precision range increased from 0.43 - 0.89 to 0.43 - 1.

**Table 6. – Comparison of SOM original and reclassified Recall and Precision**

| \ Result | Recall | | Precision | |
|---|---|---|---|---|
| Category \ | Original | Reclassified | Original | Reclassified |
| 1* | 0 | 0 | none | none |
| 2* | 0 | 0 | none | none |
| 3* | 0 | 0 | none | none |
| 4* | 0 | 0 | none | none |
| 5 | 0 | 0 | none | none |
| 6 | 0.4 | 0.4 | none | none |

| | | | | |
|---|---|---|---|---|
| 7 | 0.5 | 0.5 | 0.43 | 0.43 |
| 8 | **0.73** | **0.89** | 0.89 | 0.89 |
| 9 | **0.71** | **0.75** | 0.75 | 0.75 |
| 10 | **0.76** | **0.96** | **0.53** | **0.9** |
| 11 | 0.45 | 0.45 | 0.56 | 0.56 |
| 12 | **0.77** | **0.8** | **0.54** | **0.65** |
| 13 | **0.69** | **0.72** | **0.87** | **1** |
| 14 | **0.83** | **0.88** | **0.87** | **1** |
| 15 | **0.73** | **0.76** | **0.54** | **0.65** |

* All marked categories appeared as categories only in part of the 10 randomly generated sets – see table 1

### 3.2.2 LVQ results

LVQ-based recall and precision results are summarized in table 7. Rows represent each of the initial 15 manually defined "original" categories, while columns represent the original and new categorizations for the selected set, with the reclassified items. An improvement is seen in most categories that contain more than a minimal number of documents and in most cases there are both better recall and better precision results than in the first experiment. For the LVQ, the range of recall results increased from 0.4-0.92 to 0.4-1, while precision increased from 0.54-1 to 0.68-1.

**Table 7. Comparison of LVQ original and reclassified Recall and Precision**

| \ Result | Recall | | Precision | |
|---|---|---|---|---|
| Category \ | Original | Reclassified | Original | Reclassified |
| 1* | 0 | 0 | 0.00 | 0.00 |
| 2* | 0 | 0 | 0.00 | 0.00 |
| 3* | 0 | 0 | 0.00 | 0.00 |
| 4* | 0 | 0 | 0.00 | 0.00 |
| 5 | 0 | 0 | 0.00 | 0.00 |
| 6 | 0.40 | 0.40 | 1.00 | 1.00 |
| 7 | 0.83 | 0.86 | **0.83** | **1.00** |
| 8 | 0.91 | 0.91 | 0.71 | 0.71 |
| 9 | **0.78** | **0.93** | 0.82 | 0.82 |
| 10 | **0.45** | **0.48** | **0.67** | **1.00** |
| 11 | **0.68** | **0.71** | **0.54** | **0.68** |
| 12 | **0.92** | **1.00** | **0.62** | **0.76** |
| 13 | **0.70** | **0.92** | **0.88** | **0.92** |
| 14 | **0.82** | **0.98** | **0.78** | **0.98** |
| 15 | **0.66** | **0.94** | **0.74** | **0.77** |

* All marked categories didn't appear in the 10 randomly generated sets – see table 1

### 3.3 Revised comparison of SOM and LVQ methods

Table 8 presents the average recall and precision results for the specific set selected for re-categorization (for this reason the original numbers differ from the overall numbers in Table 5). From Table 8 it seems that both SOM and LVQ results improved after the re-classification phase, but LVQ improvement is bigger. Table 9 presents the average recall and precision results only for categories containing more than 10 documents per category. Again, there is a noticeable improvement, mainly for LVQ, in comparison to the results from the first experiment. For the LVQ, the average recall increased from 0.72 to 0.81 and average precision increased from 0.76 to 0.86.

**Table 8.** Average LVQ and SOM original and reclassified precision and recall results (all categories)

| **Learning set** | Percentage | Original | Reclassified |
|---|---|---|---|
| **Measure** | **Method** | | |
| Recall | LVQ | 0.48 | 0.54 |
| | SOM | 0.44 | 0.47 |
| Precision | LVQ | 0.51 | 0.58 |
| | SOM | 0.40 | 0.46 |

**Table 9.** Average LVQ and SOM original and reclassified precision and recall for categories with more than 10 documents.

| **Learning set** | Percentage | Original | Reclassified |
|---|---|---|---|
| **Measure** | **Method** | | |
| Recall | LVQ | 0.72 | 0.81 |
| | SOM | 0.69 | 0.75 |
| Precision | LVQ | 0.76 | 0.86 |
| | SOM | 0.66 | 0.76 |

## 4. Discussion and Future Work

The main purpose of this work was to test the possibility of automating the classification of subjectively categorized data sets. For this we worked with real data gathered from a daily work of information search and categorization.

The overall results confirm the hypothesis that it is possible to automate (with reasonable error), a subjective categorization. Both LVQ and SOM succeeded to learn user's categorization. Performing automatic clustering using either SOM or LVQ provided an overall of 69% to 75% recall and 67% to 75% precision for the two methods. The automatic categorization performance was achieved with a learning set ranging from 223 to 892 documents. This indicates that it is possible to train a system to provide useful results even with a minimal training set.

Another finding is that the supervised learning (LVQ) yields better results than the unsupervised (SOM) method, mainly at the initial steps (69% vs. 62% for recall and 67% vs. 65% for precision for the 223 documents training set). The most surprising part is that the overall performance of supervised and unsupervised ANN is quite similar, given a sufficient number of documents. Our results show that from the initial 10 test sets (Table 1), several categories got eliminated because there wasn't enough data. For most data sets, in categories 1-7 the small number of documents was insufficient for training and therefore yielded incorrect categorization. However, these problematic data sets represent real life.

The question of what is the effect of the training set size on automatic categorization performance interested us from the beginning of this study.

The results show that recall and precision improves significantly as the size of the training set grows. However, for the 80% training set size (our larger training set) the results are mixed, meaning that several categories improved their precision and recall compared to the 60% training size, while others did not. This may indicate that at 60% enough data is accumulated to learn and adapt the ANN to the user's subjective classification. However, we should notice that learning is not possible in categories where a training set is not available (or has a marginal size). We confirmed the hypothesis of the minimal number of documents required for each category computing the average Precision and Recall results for both methods for categories with more than 10 documents. Table 5 indicates that for LVQ increased from 0.67 to 0.75 and from 0.69 to 0.75 respectively. For the SOM precision and recall increased from 0.62 to 0.73 and from 0.65 to 0.73 respectively.

A major question for the adoption of automatic subjective classification systems is the required size of the training set. If a large training set is needed, it is most likely that users will not use such a system. In our case, the size of the training set varied from ~200 to ~900 data items, an average of 15 to 60 data items per category. While 15 data items may be a reasonable size for a data set, bigger sizes may be problematic. However even initial results for a small training set approximated the overall performance achieved by the larger set.

We conclude from our results that despite the subjective nature of human categorization, an automatic process can resemble subjective categorization, with considerable success. Input of the number of clusters as determined by the user, and a training set, into the automatic classifier, enabled the classification system to "learn" the human categorization.

In the second experiment we were interested in the nature of the errors that arose due to user inconsistency. For that purpose, the documents that were misclassified by the system were re-classified by the users. The reclassification of documents reveals an interesting pattern: About 40% of the documents were classified to their original categories, about one-third was classified according to the system suggestion and the rest got a different, new classification. These findings indicated that given a sufficient training set, the SOM and the LVQ can consistently support user subjective categorization.

When we take into account categories with more than 10 items per cluster, the average recall increased from 0.72 to 0.81 for LVQ and from 0.69 to 0.75 for SOM, while the average precision increased from 0.76 to 0.86 for LVQ and from 0.66 to 0.76 for SOM.

At this point is interesting to question what would happen if the whole set was re-classified. Unfortunately, such a task would not be feasible since users would never be willing to undertake such a tedious task. Nonetheless, based on this real data study, we can conclude that despite user's inconsistency (as appeared here) an automatic classification can provide valuable support to their users.

Based on the results of this study, future work should focus on the following questions:

1. Is automatic classification as effective for subjective classification as it is for objective classification (as based on a well classified data set)?
2. Will other clustering algorithms support or even improve the present results for subjective clustering?
3. Are the users inconsistent in the same way when dealing on subjective classification? In other words, are users consistent in their inconsistent classification?
4. In order to support users' changing preferences the user model needs to be adaptable. ANNs are not adaptable in nature, in order to change their performance they need re-training. Hence the questions are when re-training is needed? And what is the kind and amount of information is needed for re-training?

## References


1. Baeza-Yates and Ribiero-Neto (1999) *Modern Information Retrieval*, Addison-Wesley, 1999.
2. Boger, Z. Kuflik, T., Shapira, B. and Shoval, P. (2000) Information Filtering and Automatic Keywords Identification by Artificial Neural Networks *Proccedings of the 8th Europian Conference on Information Systems*. pp. 46-52, Vienna, July 2000.

3. Chin, D.N. (1989). Knome: Modeling What the User Knows in UC. In Wahlster, W. & Kobsa, A. (Eds.) *User Models in Dialog Systems*. Berlin - New York. Springer - Vela. pp 74-107.
4. Dawes, R. M. (1979). The Robust Beauty of Improper linear models in Decision Making. *American Psychologist*, 34, 571-582.
5. Dix, A. Finlay, J. Abowd, G. & Beale, R. (1997). 2nd Edition, *Human-Computer Interaction.* Prentice-Hall, UK.
6. Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1997). WEBSOM - Self-Organizing Maps of Document Collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo,* Finland, June 4-6*,* pages 310-315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.
7. Jain, A. K., Murty, M. N., Flynn, P. J. (1999) Data Clustering: A Review, *ACM Computing Surveys*, Vol 31, No. 3 pp. 264-323, September 1999
8. Kass, R. & Finin, T. (1988). Modeling the User in Natural Language Systems. *Computational Linguistics*, 14 (3): 5-22. Special Issue on User Modeling (Kobsa, A. & Wahlster, W. Eds.)
9. Kohonen, T. (1997). *Self-Organizing Maps*. 2nd ed., Springer-Verlag, Berlin.
10. Rauber A. and Merkl. D. (1999). Using self-organizing maps to organize document archives and to characterize subject matters: How to make a map tell the news of the world *Proceedings of the 10th Intl. Conf. on Database and Expert Systems Applications* (DEXA'99), Florence, Italy.
11. Salton, G., McGill, M. *Introduction to Modern Information Retrieval.* McGraw-Hill New-York (1983).
12. Segal, B. R., and Kephart, J. O., (1999) MailCat: An Intelligent Assistant for Organizing E-Mail, *Proceedings of the Third International Conference on Autonomous Agents*. Pp. 276-282
13. Sukaviriya, P. & Foley, J. D. (1993) Supporting Adaptive Interfaces in a Knowledge-Based User Interface Environment. In Proceedings of the International Workshop on Intelligent User Interfaces. Orlando, Florida. ACM Press. pp. 107-114.
14. Vesanto, J., Alhoniemi, E., Himberg, J., Kiviluoto, K., & Parviainen, J. (1999). Self-Organizing Map for Data Mining in Matlab: The SOM Toolbox. *Simulation News Europe*, (25):54.
15. Zamir, O., and Etzioni O. (1998), Web Document Clustering: A Feasibility Demonstartion, *Proceedings of SIGIR 98*, Melbourne, Australia.