Web Mining with Self-Organizing Maps

Emil Şt. Chifu
Department of Computer Science
Technical University of Cluj-Napoca
Bariţiu 28, RO-400027 Cluj-Napoca
Romania
Emil.Chifu@cs.utcluj.ro

Ioan Alfred Leţia
Department of Computer Science
Technical University of Cluj-Napoca
Bariţiu 28, RO-400027 Cluj-Napoca
Romania
letia@cs.utcluj.ro

Abstract – **The self-organizing map (SOM) is a data mining and visualization method for complex high dimensional data sets. We have applied the SOM model in Web mining, by giving sets of documents as input data space for SOM. The result of applying SOM on a set of documents is a map of documents, which is organized in a meaningful manner so that documents with similar content appear at nearby locations on the two-dimensional map display. From the information retrieval point of view, our implemented SOM-based system creates document maps that are readily organized for browsing. A document map also clusters the data, resulting in an approximate model of the data distribution in the high dimensional document space. The paper describes some promising experimental results, where a couple of meaningful clusters have been discovered by our system in a subset of the "20 newsgroups" data set. The clustering capability of our system allows users to find out quickly what is new in a Web site of interest by comparing the clusters obtained from the site at different moments in time.**

I. INTRODUCTION

The self-organizing map (SOM) is a very popular unsupervised neural network model for the analysis of high dimensional input data [7]. It is a clustering, visualization and abstraction method based on displaying the data set in another, more usable representation form. SOM allows mapping the high dimensional input data onto a two-dimensional output space. The resulting map is a two-dimensional grid of arrays, which preserves the structure of the input data as faithfully as possible: data items close to each other in the high dimensional data space are close to each other on the map. The main advantage of the self-organizing maps is that large quantities of data can be organized quickly into a compact form that reveals the structure within the data. As such, a SOM map displays an overview of the data.

A somehow non-classical approach in the mining of Web documents is the one based on the self-organizing maps [4, 7]. The method is applicable to any collection of (hyper) text documents and is especially suitable when the user has rather limited knowledge about the domain or the contents of the text collection. Our implemented SOM-based system manages a large collection of HTML documents by spreading them on a SOM map. Semantically similar documents occupy the same position or neighbor positions on the map, depending on the degree of semantic content similarity. The system allows the user to navigate on the document map, in order to retrieve relevant documents from different topics.

We will also show that our self-organizing maps are also capable of finding semantically meaningful clusters on a map of documents. By a cluster in a SOM map we mean a contiguous group of neurons in an area of the map where all the neurons contain similar data items (for instance, documents similar in content on a document map). The cluster visualization capability is based on applying the unified-distance matrix (U-matrix) algorithm on a SOM map [3, 13, 14]. The flat clusters can be visually discovered with the help of different grey-levels on the map as induced by the U-matrix algorithm. Our experimental results from clustering documents are encouraging.

II. SELF-ORGANIZING MAPS

The self-organizing maps have been created by Teuvo Kohonen as a particular kind of neural networks [7]. There are multiple views on SOM; the different definitions are the following. SOM is a model of specific aspects of biological neural nets (the ordered "maps" in the cortex). SOM is a model of unsupervised machine learning and an adaptive knowledge representation scheme. SOM is a tool for statistical analysis and visualization: it is both a projection method which maps a high dimensional data space into a lower dimensional one and a clustering method so that similar data samples – represented as vectors of numerical attribute values – tend to be mapped on nearby neurons. The resulting lower dimensional output space is a two-dimensional grid of arrays (the SOM map) which visualizes important relationships among the data, – which are latent in the input data set – in an easily understandable way. This dimensionality reduction maintains the topology of the input vectors, i.e. inputs that are close to each other – in other words, similar – in the input space are also close to each other in one of the clusters of the map.

In short, SOM is a data mining and visualization method for complex high dimensional data sets. Even though there are no explicit clusters in the input data set, important relationships are nevertheless latent in the data. SOM can discover and illustrate these latent structures of an arbitrary data set. SOM can describe different aspects of a phenomenon in any domain, provided that the data in the domain can be represented by vectors of numerical attributes.

The map learns by a self-organization process. No a priori knowledge about the membership of any input data item (vector) in a particular class or about the number of such classes is available. Hence, the training proceeds with unlabeled input data like any unsupervised learning. The clusters (classes) are instead discovered and described with gradually detected characteristics during the training process.

The map consists of a regular two-dimensional (rectangular) grid of processing units – the neurons. Each unit has an associated model of some multidimensional observation, represented as a vector of attribute values in a domain. SOM learning is an unsupervised regression process which consumes at every iteration one available observation represented as a vector of values for the attributes in a given domain. The role of a learned map is to represent all the available observations with optimal accuracy by using a restricted set of model vectors associated to the map units.

A. The Learning Algorithm

The initial values for the model vectors – also referred to as reference vectors – of the map units can either be chosen depending on the problem domain or they can be taken randomly. Each iteration of the learning algorithm processes one input (training) vector (one sample) $\mathbf{x}(t)$ as follows. Like usually for unsupervised neural networks, some form of a competitive learning takes place: the winner unit index c, which best matches the current input vector, is identified as the unit where the model vector is most similar to the current input vector in some metric, e.g. Euclidean:

$$\| \mathbf{x}(t) - \mathbf{m}_c(t) \| \le \| \mathbf{x}(t) - \mathbf{m}_i(t) \| , \qquad (1)$$

for any unit index i. Then all the model vectors or a subset of them that correspond to units centered around the winner unit c – i.e. units in the neighborhood area of c –, including the winner itself, are adjusted in the direction of the input vector, as follows:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t) * [\mathbf{x}(t) - \mathbf{m}_i(t)] , \qquad (2)$$

where h_{ci} is the neighborhood function, which is a decreasing function on the distance between the i-th and c-th units on the map grid, and whose maximum value corresponds to $i = c$. In practice, the neighborhood area is chosen to be wide in the beginning of the learning process, and both its width and height decrease during learning. The updating in (2) forms a globally ordered map in the process of learning.

A map unit has six immediate neighbors in a hexagonal map topology, which is usually the preferred topology. This is only a hexagonal lattice type of the two-dimensional array (grid) of neurons, so the SOM map continues to be a planar rectangle. The hexagonal neighborhood topology effect is gained by shifting correspondingly rows number 1, 3, 5, ... of the rectangular map to the right and keeping rows 0, 2, 4, ... untouched. A rectangular topology corresponds to a rectangular lattice, and a map unit has only four immediate neighbors. Consequently, the number of neighbor units affected during the learning is only four as compared to six in the hexagonal topology.

After the training of a map, its reference vectors have converged to stationary values and the result is a topology-preserved map. Similar reference vectors become close to each other, and dissimilar ones become far from each other on the map. Moreover, two input data items, which are close to each other in the input data space are mapped onto the same or neighboring neurons on the map. Each neuron together with its own reference vector represents similar data items of the input space, and a set of neighboring neurons with similar reference vectors creates a cluster.

B. Cluster Visualization

A subset of data items which are close to each other in a high dimensional input data space – and thus defines a cluster in the input space – are arranged to a map area consisting of neurons close to each other also in the two-dimensional SOM display. As a consequence, the problem of discovering a cluster in a high dimensional data set with the help of the self-organizing maps reduces to the problem of discovering the map area whose neurons to contain all the data in the cluster. Actually, we have to find the boundaries of the map cluster. Finding the boundaries of a SOM map cluster is based on applying the unified-distance matrix (U-matrix) algorithm on a SOM map [13].

U-matrix visualizes the map in grey-levels, in order to express how similar or dissimilar adjacent neurons are [3, 14]. In a hexagonal self-organizing map topology, six hexagons (extra neurons) around each neuron separate geometrically the neuron from its six immediate neighbors and show its similarity with each of them. The lighter a separating hexagon, the bigger the similarity of the reference vectors of the two separated neurons, and the darker the hexagon, the bigger the dissimilarity of the reference vectors. This way, SOM map clusters can be discovered visually as "valleys" or "depressions" (light areas) separated by "hills" (dark areas or borders). Moreover, the higher (i.e. darker) a hill separating two clusters, the more dissimilar the clusters in the multidimensional input data space.

In [13], an older (in fact the original) version of the U-matrix algorithm is used, by calculating at each map unit the sum of the distances of the reference vector of that neuron to the reference vectors of the immediate neighboring neurons.

III. SELF-ORGANIZING MAPS IN WEB MINING

Applying SOM on natural language data means doing data mining on text data, for instance Web documents [8]. The role of SOM is to cluster numerical vectors given at input and to produce a topologically ordered result. The main problem of SOM as applied to natural language is the need to handle essentially symbolic input such as words. If we want SOM to have words as input then SOM will arrange the words into word categories. But what about the input (training) vector associated to each input word? What should be the vector components, i.e. the attributes of a word? Similarity in word appearance is not related to the word meaning, e.g. "window", "glass", "widow".

We have chosen to classify words by SOM, creating thus word category maps. The attributes of the words in our experiments were the count of the word occurrences in each document in a collection of documents. Consequently, we have chosen to represent the meaning of each word as related to the meanings of text passages (documents) containing the word and, symmetrically, the semantic content of a document as a bag-of-words style function of the meanings of the words in the document.

The lexical-semantic explanation of this contextual usage meaning of words is that the set of all the word contexts in which a given word does and does not occur provides a set of mutual constraints that captures the similarity of meaning of words and passages (i.e. documents, contexts) to each other. The measures of word-word, word-passage and passage-passage relations are well correlated with several cognitive phenomena involving semantic similarity and association [10]. The meaning of semantically similar words is expressed by similar vectors.

After training a SOM on all the words in a collection of documents – where the vectorial coding of words represents the contextual usage –, the result self-organizing map groups the words in semantic categories. There are also other possibilities to code words, which lead to grammatical or semantic word categories [4, 5, 7].

IV. SYSTEM ARCHITECTURE

The architecture of our system is based on two self-organizing maps. The first one creates a semantically ordered spread of all the word forms in a large collection of Web documents. This is also called the map of word categories or level 1 SOM. The second SOM (called document map or level 2 SOM) represents a semantically ordered spread of all the documents in the collection, where the documents are codified as vectors that are histograms of word categories. The word categories are the ones as already induced into the word category map units. For every word category, the histogram representation of a document contains the number of word form occurrences in the document which belong to that word category. This way we have reduced the dimensionality of the document vectors from thousands of components which would correspond to thousands of different word forms in a classical bag-of-words approach. The dimensionality is reduced to around 200 or 300 components which correspond to 200 or 300 different word categories, enough to express the number of different concepts in a shallower or wider domain. Thus the reduced dimensionality removes the noise caused by the variability in word usage; since the number of dimensions is much smaller than the number of word forms, minor differences in terminology will be ignored. Our category based approach is able to solve the terminology problem in information retrieval, i.e. the problem of possibly different terminologies used in the documents and in a user query for one and the same concept, in other words, the problem of synonymy or near-synonymy.

The aim of our system is to classify the document collection by using the criterion of semantic similarity. Hence the graphical browsing interface of our system is in essence a document map (level 2 SOM).

V. SYSTEM IMPLEMENTATION

The system is written in C and bash script. We have used the LEX software package [12] for implementing the preprocessing module, which reads and counts the word occurrences in all the documents in a collection, by ignoring all the HTML tags. The preprocessing module also ignores 450 common words, i.e. English words having no semantic load. These words have been taken from the information retrieval software package GTP [2]. Finally, the preprocessing also means a stemming phase that uses a morphological analyzer for English, which is part of the GATE system [1]. The stemming is done in order to reduce the number of word forms by keeping only their stem.

The SOM_PAK [6] system is used for the training of all our SOM maps. The result of training the document SOM is a text file containing for every document category a list of document names that belong to that category, i.e. the list of documents managed into the corresponding map unit. The format of this text file is exemplified with seven document categories in Fig. 1, where each row corresponds to a different map unit. The first two integer numbers in each row represent the rectangular coordinates (*x* and *y*) of the current unit. The document category name follows the coordinates of the unit and becomes the identification label of the unit. The document category name is given by the name of the first document in the (training) data set that "hit" the unit during the training process. This name occurs as the last in the enumeration of document names in the category, after the colon.

All the seven document categories in Fig. 1 are semantically related as they all contain as documents emails from one and the same newsgroup (*talk.politics.mideast*) in the "20 newsgroups data set" [11]. The seven corresponding map units are neighbors on the document map and they constitute together an area or *cluster*. The aggregation of the neurons in this cluster is noticeable from their coordinates and from the hexagonal topology adopted.

A. Graphical User Interface

The graphical interface has been implemented by using the PHP language. The system creates the graphical interface as an interactive graphical display that is implemented as a dynamically created HTML file. This PHP module reads the text file of document categories (created by the document SOM and exemplified in Fig. 1) and translates this document classification into a dynamical HTML file which is the graphical display of the document map itself. Every map unit is labeled with the associated document category name, which is found out as explained at the beginning of the current section (Section V). A better alternative would be to label a map unit with the most relevant and representative document in the corresponding category, i.e. the name of the document whose vector is closest to the model vector of that unit [9]. A second label on each map unit represents the number of documents in the corresponding category. For instance, the map unit for the last document category in Fig. 1, having coordinates 9, 5 on the map, is also labeled 6.

The interface allows the navigation on the document map from any Web browser. The aim of this browsing is the retrieval of relevant documents in two steps. Click on a map unit gives access to the index of documents in that unit, which is also a dynamically generated HTML file, containing a list of links, each of which having as text the document name and pointing to the document itself. Then click on a document name in this list allows viewing that document.

```
 8   3 75381   :   75381

 8   4 75382   :   75369  75382

 9   4 75394   :   75371  75389  75394

10   4 75393   :   75393

11   4 75400   :   75370  75392  75400

 8   5 75395   :   75395

 9   5 75399   :   176854  75366  75387  75388
75390  75399
```

Fig. 1. Example document categories

VI. EXPERIMENTAL EVALUATION

The experiments reported here take as test data the "20 newsgroups data set" [11]. This data set contains 20,000 UseNet news postings having the form of email messages. The 20,000 messages were collected at random from 20 different Netnews newsgroups, 1000 messages from each newsgroup [11]. The data set is "labeled", by being already partitioned into twenty categories. This labeling helped us in evaluating the clustering results of the same set of email documents as discovered and visualized by our document SOM. In one of our most successful experiments, we have selected randomly 40 documents from each newsgroup, summing up a total set of 800 message documents. This balanced subset of the original "20 newsgroups" data set has been taken as input data space for our SOM-based system in order to arrive at an email document SOM map.

An important question in this experiment was to choose a size for the SOM map, in order to arrive at a map with the highest degree of visual expressiveness for clustering [14]. The map size means the total number of neurons of the rectangular grid. For a given data set, different map sizes mean different granularity levels, in terms of the average number of data items to belong to a neuron. If the map is too small, it is too rough and consequently it might hide some important differences that should be detected in order to separate the clusters. This is because too many unsimilar data items could belong to the same neuron. When the map is too big, then it is too detailed and, besides the important differences, the map displays also too small differences, which are often unimportant for clustering. This is because data items which are very similar could belong to different neurons, when normally we expect them to belong to one single neuron.

We have chosen a map size of 16 (columns) times 12 (rows) considered as suitable for the input data space of 800 data items (800 email documents). This also conforms to the suggestions in [14], where a suite of experiments with input data sets of different cardinality and different SOM map sizes is described. Fig. 2 shows the result email document SOM map image, where grey levels occur as an effect of applying the U-matrix algorithm for cluster visualization. The U-matrix algorithm used here is included in the SOM_PAK program package [6], which is part of our system. The algorithm conforms to the description in Section II, *B*.

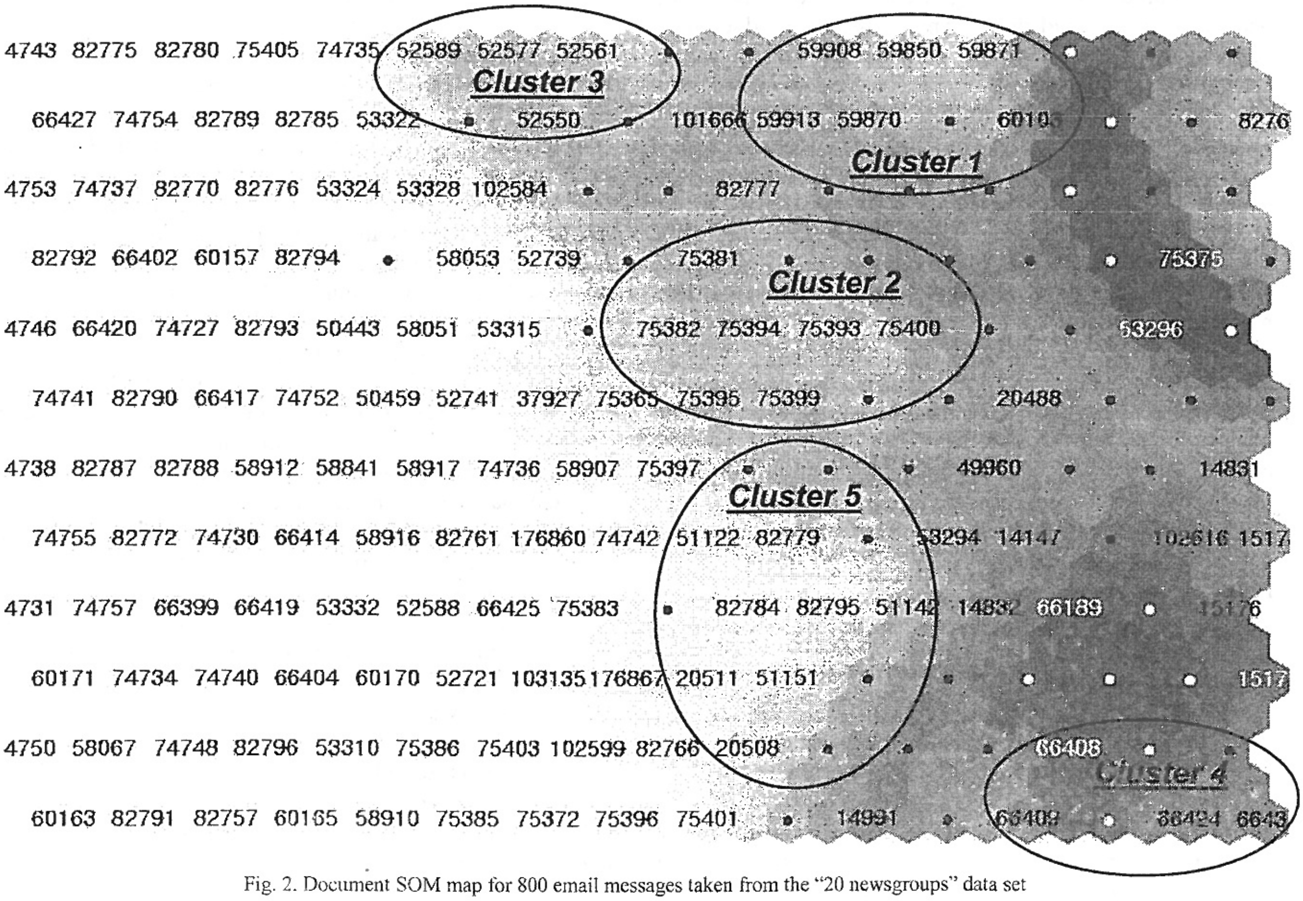

Fig. 2. Document SOM map for 800 email messages taken from the "20 newsgroups" data set

A. Clustering Results

The document map in Fig. 2 clearly illustrates four clusters discovered by the map. Neurons in Cluster 1 contain 15 email messages, which all belong to the newsgroup *science.space.* Only 15 out of a total of 40 messages in the *science.space* newsgroup in the input space were discovered to belong to Cluster 1. Hence, even if the *accuracy* of this cluster is 100%, its *coverage* is only 15/40, so 37.5%.

All the 17 documents in Cluster 2 belong to the newsgroup *talk.politics.mideast*, but there are 41 messages in the input data that belong to this newsgroup. Cluster 2 contains seven neurons whose explicit description in terms of email messages grouped as document categories in each neuron is given in Fig. 1. Actually only 40 input messages are "officially" labeled by the authors of the "20 newsgroups" data set to belong to the newsgroup *talk.politics.mideast.* One more message, named `176854`, and found out by our map to belong to Cluster 2, has been "abusively" put by the authors into another newsgroup, namely *talk.politics.misc*. The header of this email indicates explicitly `Newsgroups: talk.politics.mid-east, misc.headlines, talk.politics.misc.`

Similarly, Cluster 3 contains 12 messages, 11 of them from the newsgroup *rec.sport.hockey*. This cluster is less clearly bordered on the map, because of the semantic overlap with other messages some of them form the related newsgroup *rec.sport.baseball.* In fact, the only message in Cluster 3, which is outside of the expected newsgroup *rec.sport.hockey*, is from the related newsgroup *rec.sport.baseball.* Finally, Cluster 4 on the map represents 11 messages, 10 of them from the newsgroup *comp.windows.x*, and one from the related newsgroup *comp.sys.ibm.pc.hardware.* Table I shows the classification quality parameters accuracy and coverage associated with the four clusters.

B. Discussion of Results

There are some more results found out from our document map induced from 800 news messages, and illustrated in Fig. 2. For instance, there is one more cluster, Cluster 5, also mentioned in Table I, which contains 26 email messages, 21 of them being a mixture of messages from three different newsgroups: *talk.religion.misc*, *soc.religion.christian*, and *alt.atheism*. The first two newsgroups are obviously related to each other, and they are also semantically related with the third, even if this relation sounds more like an antonymy. Similar topics are nevertheless discussed in messages about religion and atheism.

About 85% of the 800 email messages are contained in about the left half of the map, which is completely white, and constitutes a huge cluster. Such a cluster has no clear semantic content, because it contains messages from all the 20 newsgroups, including the messages left out from the five clusters already mentioned. The technical explanation for this phenomenon is that the document SOM map was unable to display semantic differences in this big cluster. The differences in the semantic content of the messages could be too small when the authors of the messages use too few words specific to the domain of the newsgroup or sometimes when they communicate announcements with no bearing with the domain of the newsgroup.

Another explanation for the huge cluster is that the majority of the email messages in the "20 newsgroups" data set are addressed to many different real newsgroups. The more newsgroups a message is addressed to, the more arbitrary its inclusion (by the authors of the "20 newsgroups" data set) in one of the 20 groups, and the fewer semantic differences discernable by our SOM-based system for such messages.

Our clustering results were worse when we didn't ignore the 450 stop words mentioned in Section V, because these words with no semantic load introduced noise that reduced the capability of displaying semantic differences between email message documents. We have also examined some word category SOM maps. One of our first interesting results with word maps is that, when we didn't ignore the stop words, the word category maps contained isolated (unclustered) neurons representing stop words, which were situated only near the margins of the map. The explanation is that the word map separates the stop words from the other content-rich words, the latter being contained in the interior of the map.

VII. CONCLUSIONS

The self-organizing maps constitute a powerful model for Web mining by defining a visual overview of a set of Web documents. A document SOM map is a semantically ordered spread of the documents in the set.

TABLE I

CLASSIFICATION ACCURACY AND COVERAGE ASSOCIATED WITH DOCUMENT CLUSTERS IN FIG. 2

Cluster	Newsgroup	Correct	Actual	Predicted	Accuracy (Correct/Actual)	Coverage (Correct/Predicted)
Cluster 1	*Science.space*	15	15	40	100%	37.5%
Cluster 2	*talk.politics.mideast*	17	17	41	100%	41.5%
Cluster 3	*rec.sport.hockey*	11	12	40	91.5%	27.5%
Cluster 4	*comp.windows.x*	10	11	40	90.9%	25%
Cluster 5	Combination of *talk.religion.misc,* *soc.religion.christian,* *alt.atheism*	21	26	120 (3x40)	80.8%	17.5%

Our SOM-based implemented system is a powerful information retrieval tool for browsing a set of Web documents. The system is especially useful when the user has rather limited knowledge about the domain or the contents of the text collection. By using the unified distance matrix (U-matrix) algorithm, our Web mining system is also able to find semantically meaningful clusters on a map of documents. We have reported here some promising experimental results from document clustering.

As a further work, we can apply our Web mining system in order to arrive at a visual overview of all the documents in a given Web site. This overview is a snapshot image of the given site for a given moment in time. When a new snapshot of the same site is captured after a period of time, then the new document map could be compared visually with the old one. The differences in the clusters of documents on the two maps will easily indicate the dynamics of the changes intervened in the site. This is useful for users to find out quickly what is new in their sites of interest.

In order to improve the ability to display semantic differences for clustering, we will introduce some weighting in our bag-of-words approach, for instance the inverse document frequency. Words or word categories occurring in too many documents will receive a low weight, because of their low discriminating power (i.e. low information gain). We can also ignore all the words that only occur once in the whole set of documents.

VIII. ACKNOWLEDGEMENTS

Part of this work was supported by the Romanian CNCSIS under Grant 528/2002.

IX. REFERENCES

[1] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A framework and graphical development environment for robust NLP tools and applications", in *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[2] J.T. Giles, L. Wo, and M.W. Berry, "GTP (General Text Parser) software for text mining", in H. Bozdogan, ed., *Statistical Data Mining and Knowledge Discovery*, CRC Press, Boca Raton, 2003, pp. 455-471.

[3] S. Hautaniemi, O. Yli-Harja, J. Astola, P. Kauraniemi, A. Kallioniemi, M. Wolf, J. Ruiz, S. Mousses, O.-P. Kallioniemi, "Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps", *Machine Learning*, vol. 52, 2003, pp. 45-66.

[4] T. Honkela, "Self-organizing maps in natural language processing", *PhD thesis*, Neural Networks Research Center, Helsinki University of Technology, Finland, 1997.

[5] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "Exploration of full-text databases with self-organizing maps", in *Proceedings of the International Conference on Neural Networks*, 1996, vol. I, pp. 56-61.

[6] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "SOM_PAK: The self-organizing map program package", *Technical Report A31*, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.

[7] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela,. V. Peetero, and A. Saaerela, "Self-organization of a massive document collection", *IEEE Transactions on Neural Networks*, vol. 11, no. 3, 2000, pp. 574-585.

[8] K. Lagus, "Text retrieval using self-organized document maps", *Technical Report A61*, Helsinki University of Technology, Laboratory of Computer and Information Science, 2000.

[9] K. Lagus and S. Kaski, "Keyword selection method for characterizing text document maps", in *Proceedings of the 9th International Conference on Artificial Neural Networks*, 1999, vol. 1, pp. 371-376.

[10] T.K. Landauer, P.W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis", *Discourse Processes*, vol. 25, 1998, pp. 259-284.

[11] K. Lang, "NewsWeeder: Learning to filter news", in *Proceedings of the 12th International Conference on Machine Learning*, 1995, pp. 331-339.

[12] M.E. Lesk and E. Schmidt, "Lex – a lexical analyzer generator", *Computing Science Technical Report 39*, AT&T Bell Laboratories, Murray Hill, 1975; *UNIX Programmer's Manual*, vol. 2B, Bell Laboratories, 1975.

[13] A. Ultsch, "Self organized feature maps for monitoring and knowledge acquisition of a chemical process", in S. Gielen and B. Kappen, eds., *Proceedings of the International Conference on Artificial Neural Networks*, 1993, pp. 864-867.

[14] E. Wilppu, "The visualization capability of self-organizing maps to detect deviations in distribution control", *Technical Report 153*, Turku Centre for Computer Science, 1997.