ELSEVIER

Neurocomputing 21 (1998) 101–117

NEUROCOMPUTING

WEBSOM – Self-organizing maps of document collections[1]

Samuel Kaski*, Timo Honkela, Krista Lagus, Teuvo Kohonen

Helsinki University of Technology, Neural Networks Research Centre, P.O. Box 2200, FIN-02015 HUT, Finland

Accepted 26 May 1998

Abstract

With the WEBSOM method a textual document collection may be organized onto a graphical map display that provides an overview of the collection and facilitates interactive browsing. Interesting documents can be located on the map using a content-directed search. Each document is encoded as a histogram of word categories which are formed by the self-organizing map (SOM) algorithm based on the similarities in the contexts of the words. The encoded documents are organized on another self-organizing map, a document map, on which nearby locations contain similar documents. Special consideration is given to the computation of very large document maps which is possible with general-purpose computers if the dimensionality of the word category histograms is first reduced with a random mapping method and if computationally efficient algorithms are used in computing the SOMs.

Keywords: Data mining; Information retrieval; Self-organizing map; SOM; WEBSOM

1. Introduction

The basic approaches for information retrieval and data mining in textual document collections are: (1) searching using keywords or key documents, (2) exploration of the collection referring to some organization or categorization of the documents, and (3) filtering of interesting documents from the incoming document stream. Keyword search systems can be automated rather easily whereas the organization of document collections has traditionally been carried out by hand. In manual organiza-

* Corresponding author.

[1] This work was supported by the Academy of Finland.

0925-2312/98/$ – see front matter
PII S0925-2312(98)00039-3

tion carried out, for example, in libraries, classification schemes are defined and each document is positioned into one or several classes by a librarian. Similarly, in the current hypertext systems the links between related documents are most often added by hand.

One of the traditional methods of searching for texts that match a query is to index all the words (hereafter called *terms*) that have appeared in the document collection. The query itself, typically a list of appropriate keywords, is compared with the term list of each document to find documents that match the query. Terms can be combined by Boolean logic in order to control the breadth of matching. The following three fundamental problems in applying Boolean logic to text retrieval (see, e.g., [39]) make it an unsatisfactory solution. (1) Recall and precision[2] of retrieval are sensitive to small changes in the formulation of a query. For Boolean queries there is no simple way of controlling the size of the output, and the output is not ranked in the order of relevance. (2) The results of a query offer no indication on how many valuable documents were *not* retrieved, especially if the document collection is unfamiliar. (3) If the domain of the query is not known well it is difficult to formulate the query, i.e., to select the appropriate keywords.

One basic problem of information retrieval culminates in the realization that the same idea or concept can be presented in many different ways. Natural language gives freedom for enormous variation in expression, from choosing between synonyms to using different styles, emphasis, different levels of abstraction, and anaphoric and metaphoric expressions. Furthermore, the authors use their unique style that depends on their background, knowledge and personal style of communication. Given this variation it should be clear that if information retrieval is based only on the keywords as they appear in the text, some interesting documents may not be discovered, or misleading documents may be returned. The traditional keyword-based information retrieval methods have limited possibilities to tackle this phenomenon.

One suggested solution for the vocabulary or keyword selection problem is the use of thesauri. Documents are looked up according to whether keywords specified in a query and/or their synonyms, found in the thesaurus, occur in the searched document. However, the synonyms found in the thesaurus often cause unrelated entries to be returned. Furthermore, the creation and maintenance of such a thesaurus requires considerable effort.

We have developed a methodology which can be used as a tool especially in exploring a document collection but also in searching and filtering tasks. The method called WEBSOM has the potential of tackling the vocabulary problem discussed above. The texts used in our first experiments were samples of discussions from a Usenet newsgroup related to neural networks. The document collection contained some 5000 documents [8]. Later, in the largest experiments, we have used texts of 85 newsgroups containing over one million documents. Several articles on various aspects of the method have already been published [6–9,12,13,15,17,18,21–24]. The

[2] The two standard measures of the effectiveness of the retrieval are *precision*, i.e., the share of retrieved documents that were relevant, and *recall*, the share of the relevant documents that were retrieved.

current article gives a thorough description of the method including one of the latest developments, i.e., the use of random mapping to reduce the dimensionality of the encoded documents. The article reports the novel results acquired by combining the random mapping with the methods that have been developed for creating considerably larger maps [16,19]. A demonstration of the WEBSOM method is available at the WWW address http://websom.hut.fi/websom/.

2. Related methods for document encoding

The representation of documents is a central issue in all of the approaches in document management, whether the management entails exploration, search, or filtering. The representations should be as compact as possible in order to allow efficient processing of large document collections. Yet the representations should contain all of the information needed in identifying the relevant content in the documents, in an explicit form that the processing methods can take into account. In this section we shall review some document encoding methods, concentrating on methods that are sufficiently similar to the one used in WEBSOM so that they could, in principle, be used as alternative preprocessing stages. All of the methods have been originally introduced in the context of document retrieval tasks.

2.1. Vector space model

In the vector space model [39] both the stored documents and the user queries are represented by vectors in which each component corresponds to one word. The value of a component is a function of the frequency of occurrence of the word in the document and of the statistically measured importance of the word. In the simplest case the function could be an increasing function (e.g., a logarithm) of the frequency of occurrence multiplied by a weight indicating the importance of the word. The importance could be computed, for instance, simply as the inverse of the number of documents in which the word occurs, or using an entropy-based measure such as the one discussed in Section 4.2.1. The resulting vectors can be thought to represent *word histograms* of the documents. When the vectors are normalized their directions reflect the contents of the documents.

The relevance of a document to a query may be measured using the similarity of the corresponding vectors, calculated, for example, by the cosine of the angle between the vectors. The retrieved documents can then be presented as a list ordered by the similarity.

Let us formulate one version of the vector space model in mathematical terms in order to facilitate comparisons with other methods that will be discussed later. Denote by $\boldsymbol{e}_i$ the unit vector with one in the ith component and zeros elsewhere, and by n_{ji} the frequency of occurrence of the word indexed with i in the jth document. The n_{ji} may as well be some more sophisticated functions of the frequency of occurrence and the importance of the word i.

Using these notations the vector $\boldsymbol{n}_j$ that is used to encode the document indexed with j in the vector space model can be expressed as

$$\boldsymbol{n}_j = \sum_k n_{jk}\boldsymbol{e}_k. \tag{1}$$

There are two major problems with the vector space model. First, the dimensionality of the vectors that represent the documents equals the size of the vocabulary which is immense in large document collections. Therefore, it is practically only possible to use a small proportion of the vocabulary, and some more or less heuristic methods are needed for selecting the set of most important words. Another problem is due to the orthogonality of the vectors $\boldsymbol{e}_k$ that correspond to the words in Eq. (1): semantic relationships of the words are not taken into account. For example, any pair of synonymous words is treated as being completely unrelated.

2.2. Latent semantic indexing

Latent semantic indexing (LSI) [1,3] is one alternative to the original vector space model. LSI tries to take into account the co-occurrence of terms in the documents when encoding the documents.

One way of interpreting the LSI is that it represents the jth document by the vector

$$\boldsymbol{n}'_j = \sum_k n_{jk}\boldsymbol{x}'_k, \tag{2}$$

where n_{jk} denotes again the number of times the word k occurs in the jth document. The $\boldsymbol{x}'_k$ is the code that the LSI forms for the kth word by investigating the co-occurrences of the words within the documents. The term-by-document matrix, a matrix where each column is the word histogram corresponding to one document, is decomposed into a set of factors (eigenvectors) using the singular-value decomposition (SVD). The factors that have the least influence on the matrix are then discarded. The motivation behind omitting the smallest factors is that they most likely consist of noise. The vectors $\boldsymbol{x}'_i$ can then be formed by using only the remaining factors, whereby their dimensionality is reduced.

The connection between the LSI and the vector space model is evident in the Eqs. (1) and (2). The only difference is that in the LSI the words are represented by the non-orthogonal vectors $\boldsymbol{x}'_k$ that are computed based on information about the co-occurrences of the words. Thus, LSI may be able to utilize some information about the semantic relations between the words.

2.3. MatchPlus

The MatchPlus method of HNC [5] encodes the documents as sums of the vectorial representations of the words rather like the LSI (Eq. (2)). However, in the MatchPlus system the vectors $\boldsymbol{x}'_k$ are constructed differently. The dimensions of $\boldsymbol{x}'_k$ are chosen manually to correspond to a set of important "basis terms". The value of each component in $\boldsymbol{x}'_k$ is then adjusted manually to indicate the similarity of the meaning of

the word k and the corresponding basis term. After representations have been formed in this manner for a sufficient number of terms it is then possible to use contextual information for forming codes for the other terms. The representation of a new word is tuned to resemble more closely the representation of an existing term if the two terms occur near each other in the running text.

3. Random mapping method

In the vector space model discussed in Section 2.1 the documents are encoded as vectors in a very high-dimensional space. There is a one-to-one correspondence between the dimensions and the words, and the frequency of occurrence of different words in a document governs the direction of the vector that represents the document.

Unfortunately, it is impracticable to encode the documents in a large document collection using the vector space model as such: the resulting codes would have a very high dimensionality. There are as many dimensions in $\boldsymbol{n}_j$ as there are words in the vocabulary. It is clear that some kind of a dimensionality reduction is needed.

There exists, fortunately, a very simple dimensionality reduction method called *random mapping* that does not require reduction of the vocabulary. Clearly, it is not possible to preserve the whole structure of the original document vector space since the dimensionality is reduced. However, with the method the distances or similarities between the original document vectors may be preserved to a considerable degree. The motivation for the random mapping method datesback to earlier experiments [37], in which the dimensionality of the encoded forms of the contexts of words was reduced. In the random mapping method the unit vectors $\boldsymbol{e}_i$ in Eq. (1) are simply replaced with lower-dimensional *random vectors* denoted by $\boldsymbol{r}_i$. Each component of such a random vector is a random variable, and the vectors $\boldsymbol{r}_i$ are normalized to have unit length.

When the unit vectors are replaced with random vectors in the vector space model (1), the jth document will be encoded by the vector

$$\boldsymbol{x}_j = \sum_k n_{jk}\boldsymbol{r}_k. \tag{3}$$

Denote by R the matrix that is formed of the vectors $\boldsymbol{r}_k$; the kth column of R is $\boldsymbol{r}_k$. The right-hand side of Eq. (3) can then be expressed as a simple matrix multiplication. Therefore, the random mapping can be expressed as a product of the original vector $\boldsymbol{n}_j$ with a random matrix R,

$$\boldsymbol{x}_j = R\boldsymbol{n}_j. \tag{4}$$

Here $\boldsymbol{n}_j \in \mathbb{R}^N$ and $\boldsymbol{x}_j \in \mathbb{R}^n$.

Although the random mapping method may seem surprisingly simple it has already been successfully applied to encoding documents [13] which have then been organized on a document map. In the following, we shall try to shed some light on

why reduction of the dimensionality using random mapping preserves enough information of the original document vectors to be useful. The proofs of the results have been omitted; the details can be found in [14].

3.1. Properties of the random mapping

We shall begin the characterization of the properties of the random mapping method by considering how it affects the mutual *similarities* of the encoded documents. Approximate preservation of the mutual similarities is, of course, especially important for any method that groups the documents.

Similarity of two vectors having identical norm is commonly measured by the inner product (or some monotone function of it). The inner product of two vectors, $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, that have been obtained by random mapping can be expressed using Eq. (4) as follows:

$$\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{x}_k = \boldsymbol{n}_j^{\mathrm{T}} R^{\mathrm{T}} R \boldsymbol{n}_k. \tag{5}$$

Here the matrix $R^{\mathrm{T}}R$ consists of components that are inner products of the random vectors $\boldsymbol{r}_k$. The matrix can be decomposed into two terms,

$$R^{\mathrm{T}}R = I + \varepsilon, \tag{6}$$

where

$$\varepsilon_{ij} = \boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_j \tag{7}$$

for $i \neq j$, and $\varepsilon_{ii} = 0$ for all i.

The components on the diagonal of $R^{\mathrm{T}}R$ have been collected into the identity matrix I in Eq. (6). They are always equal to unity since the vectors $\boldsymbol{r}_i$ have been normalized. The off-diagonal components have been collected into the matrix ε. If all the components in ε were equal to zero the matrix $R^{\mathrm{T}}R$ would be equal to I and the similarities of the documents would be preserved exactly in the random mapping.

It is possible to say something about the statistical properties of ε if we fix the distribution of the components of the vectors $\boldsymbol{r}_i$. In our experiments the components are initially chosen to be independent, identically and approximately normally distributed (with zero mean), and thereafter the length of all $\boldsymbol{r}_k$ is normalized. Then ε_{ij} is equal to zero on the average since the $\boldsymbol{r}_k$ have been chosen uniformly and independently from the unit sphere. In fact, denoting the dimensionality of the mapped space by n the inner products are normally distributed with zero mean. It can be shown easily that the variance, denoted by σ_ε^2, can be approximated by

$$\sigma_\varepsilon^2 \approx 1/n. \tag{8}$$

Since we now know the distribution of ε it is possible to investigate more closely how the similarities of the original vectors are transformed in the random mapping. More specifically, given a pair $\boldsymbol{n}$ and $\boldsymbol{m}$ of original data vectors it is possible to derive the distribution of the similarity of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ obtained by random mapping of $\boldsymbol{n}$ and $\boldsymbol{m}$, respectively. The similarity is measured here with the inner product between the vectors.

Using Eqs. (5)–(7) the inner product between the mapped vectors can be expressed as

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{n}^{\mathrm{T}}\boldsymbol{m} + \sum_{k \neq l} \varepsilon_{kl} n_k m_l. \tag{9}$$

Denote $\delta = \sum_{k \neq l} \varepsilon_{kl} n_k m_l$; this expression is the deviance of the inner product from the original value.

It is clear that the mean of δ is zero. It can be shown [14] that the variance of δ, denoted by σ_δ^2, is bounded by the inverse of the dimensionality (multiplied by 2)

$$\sigma_\delta^2 \leq 2\sigma_\varepsilon^2 \approx 2/n. \tag{10}$$

Thus, the distances between the original vectors will be preserved quite accurately in the random mapping if the final dimensionality n is sufficiently large. Therefore, while encoding documents using the vector space model it is possible to reduce the dimensionality of the vectors while still preserving approximately the information about the mutual similarities of the documents.

4. WEBSOM method

In the WEBSOM method [8] the documents are organized using the SOM algorithm [20] onto a document map. A graphical display of the map provides a general overview of the information contained in the document collection.

The vocabulary problem is addressed in WEBSOM by organizing the words into categories on a *word category map* (also called self-organizing semantic map; see [37,38]). The word category map can then be used to encode the documents in a manner that explicitly expresses the similarity of the word meanings. Even very large word category maps can be used since the dimensionality of the resulting encoded documents can be reduced by the random mapping method discussed above.

4.1. Using contextual information: Word category map

As noted before, a fundamental problem in the vector space model discussed in Section 2.1 is that it does not take into account information about the relations of words. For example, a pair of synonyms is treated exactly as any other pair of words. We shall next present an extension of the vector space model that takes into account the semantic similarities of words by grouping them automatically into word categories consisting of similar words.

Word category maps are SOMs which have organized words according to similarities in their roles in different contexts. Each unit on the SOM corresponds to a *word category* that consists of a set of similar words.

The use of the SOM requires that the input is presented as numerical vectors and that a metric for comparing the vectors is available. The metric cannot be based on the similarity of the appearances of the words since appearance does not generally

correlate with the content the words refer to. A useful metric can, however, be obtained by taking into account the sentential context in which the words occur.

The original method for creating maps of words based on contextual information is presented in [37]. Scholtes [40] has used this principle extensively. In studies of this kind, emergence of the word categories is based on the statistical occurrence of words in different contexts. The input vector for the word category map contains information about the context, for example, about the *average context*, in which the words occur. The experiment reported by Ritter and Kohonen [37] showed that it is possible to create lexical maps which match reasonably well with linguistic categories and the intuition regarding the semantics of the words. It must be emphasized that the positions of words on a word category map are based solely on the contextual information (e.g. predecessors and successors) and no prior classification is given.

4.1.1. Encoding of the contexts

Let us next consider how the averaged contextual information relating to the ith word in the vocabulary could be encoded. Denote the set of words that have occurred at displacement d from the word i in the running text by $I_i^{(d)}$ (each word may occur several times). For example, the set of words that have immediately succeeded word i is denoted by $I_i^{(1)}$. Our aim is to compute a statistical description, denoted by the vector $\boldsymbol{x}_i^{(d)}$ that will be defined later, of this set, and then to combine the descriptions obtained at different displacements d. The combination obtained by concatenating the descriptions computed at the set of displacements $\{d_1, \ldots, d_N\}$ is

$$\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{x}_i^{(d_1)} \\ \vdots \\ \boldsymbol{x}_i^{(d_N)} \end{bmatrix}. \tag{11}$$

The vector $\boldsymbol{x}_i$ is a numerical description of the contexts in which the word i has occurred, and it can be used when organizing the words onto the word category map.

One possible method for encoding the average context at distance d is to encode each word k by a different unit vector $\boldsymbol{e}_k$ as was done in the vector space model, and then to take the average over the occurrences of the different words. The average of the context of word i at the distance d could then be encoded as the vector

$$\boldsymbol{x}_i^{(d)} = \frac{1}{|I_i^{(d)}|} \sum_{k \in I_i^{(d)}} \boldsymbol{e}_k, \tag{12}$$

where $|I_i^{(d)}|$ denotes the number of words in $I_i^{(d)}$. The $\boldsymbol{x}_i^{(d)}$s can be additionally normalized. This expression resembles very closely expression (1) used to describe the vector space model; the difference is that here the averages are formed over the contexts of a word whereas in the vector space model the averages were taken over a document.

Here, as in the vector space model, we encounter a problem: the dimensionality of $\boldsymbol{x}_i^{(d)}$ in Eq. (12) is very large for large vocabularies. It is possible, however, to apply the random mapping method also here to reduce the dimensionality. In practice, the random mapping can be implemented by first encoding each word i by a preliminary,

say, 90-dimensional random vector $\boldsymbol{r}_i$. These vectors can then be substituted to the $\boldsymbol{e}_k$ in Eq. (12), resulting in the encoding of the context by

$$\boldsymbol{x}_i^{(d)} = \frac{1}{|I_i^{(d)}|} \sum_{k \in I_i^{(d)}} \boldsymbol{r}_k \tag{13}$$

and finally normalizing the resulting vector. This is the approach used originally by Ritter and Kohonen [37] when constructing the first self-organizing semantic maps.

4.1.2. Summary

The basic steps in forming a word category map can be summarized as follows.

1. Create a unique random vector $\boldsymbol{r}_i$ for each word i in the vocabulary.
2. Find all the instances of each word chosen to be considered, called *keyword* in the following, in the text collection. Calculate the average over the context vectors of each keyword using Eqs. (11) and (13). The random codes formed in step 1 are used in the calculation. As a result each keyword is associated with a contextual fingerprint.
   In practice, we often compute and use contextual information from only two displacements, from the words immediately preceding and succeeding the keyword, respectively. In this way the amount of computations remains manageable. The contextual information relating to word i is then represented by
   $$\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{x}_i^{(1)} \\ \boldsymbol{x}_i^{(-1)} \end{bmatrix}, \tag{14}$$
   where $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(-1)}$ are computed using Eq. (13).
3. Each vector formed in step 2 is input to a SOM. The resulting word category map is labeled after the training process by inputting the $\boldsymbol{x}_i$ once again and labeling the best-matching units according to the words corresponding to the $\boldsymbol{x}_i$. In this method a unit may become labeled by several words, often synonymous or with opposite meaning, or forming a closed attribute set.

The map units may thus be viewed as word categories. The overall organization of a word category map reflects the syntactic categorization of the words [10]. An example of a word category map is shown in Section 5. Several other studies have also been published on SOMs that map words into grammatical and semantic categories; see for instance [4,35,40].

4.2. Document maps based on a two-level architecture

Each document is encoded by locating the categories of its words on the word category map. The histogram of the "hits" on the word category map is updated at the location of the word, and finally the histogram is normalized. The histogram resembles the vectors used in the vector space model – here the components of the vectors correspond to word categories instead of single words. To speed up the

computation, the positions of the word labels on the word category map may be looked up by hash coding.

The *document map* is formed with the SOM algorithm using the histograms as "fingerprints" of the documents.

4.2.1. Entropy-based weighting of the words

If the documents can be classified into a set of groups having different topics, such as the newsgroups in the Internet, the word counts in the word category histogram can be further *weighted* by the information-theoretic *entropies* (Shannon entropies) of the words, defined in the following way. Denote by $n_g(w)$ the total frequency of occurrence of word w in the documents belonging to group g ($g = 1, \ldots, N_g$), normalized by the total number of words in the group. Here N_g is the number of the groups. Denote by $P_g(w)$ the probability that a randomly chosen instance of the word w occurs in group g (after the sizes of the groups have been normalized). The entropy H of this word is defined to be

$$H(w) = -\sum_g P_g(w) \log P_g(w) \approx -\sum_g \frac{n_g(w)}{\sum_{g'} n_{g'}(w)} \log \frac{n_g(w)}{\sum_{g'} n_{g'}(w)} \tag{15}$$

and the weight $W(w)$ of word w is defined to be

$$W(w) = H_{\max} - H(w), \qquad H_{\max} = \log N_g. \tag{16}$$

If no information about groupings of the documents is available the words may be weighted, for instance, by their inverse frequency of occurrence or some other traditional method (cf., e.g., [39]).

4.2.2. Reduction of the dimensionality of the word category histograms by random mapping

When the vocabulary is very large, of the order of tens of thousands, the number of word categories in a word category map will be large as well although it will be smaller than the dimensionality of the vectors used in the vector space model. It is necessary to use a large word category map since, unless the average size of the word categories is only a few words, several word categories will probably contain words that do not have much in common. The word category histograms will thus contain at least of the order of thousands of entries, and it will be costly to compute large document maps having so many input dimensions.

We have described a useful solution to the problem of large-dimensional inputs already in Section 3: the random mapping method. Instead of reducing the dimensionality of word histograms we now reduce the dimensionality of word category histograms by mapping them into a lower-dimensional space using the random mapping operation, Eq. (4).

4.2.3. Relation to other document encoding methods

When compared with the vector space model the WEBSOM offers automatic methods for reducing the dimensionality of the encoded documents, and by means of

categorizing the words it is possible to take into account some information about the semantic relatedness of the words.

LSI differs from the approach used in WEBSOM in two respects. In LSI the context from which the codes of the words are estimated consists of the whole documents whereas we have used shorter contexts. Another difference is that we use SOM-based clustering and random mapping to reduce the dimensionality of the codes whereas LSI uses the SVD method. The MatchPlus system utilizes some manual steps to take into account the relations of the words, although the authors hint at a possibility to use fully automatic methods.

4.3. Other SOM-based studies

In an early study, Lin formed a small map of scientific documents based on the words that occurred in the titles [26,28]. Later Lin has extended his method to full-text documents [27]. Scholtes has developed, based on the SOM, a neural document filter and a neural interest map for information retrieval [40]. In addition to encoding the documents based on their words, Scholtes has used character *n*-grams, sequences of *n* characters, in the encoding. This approach has also been adopted in [11]. Merkl [27–34] has used the SOM to cluster textual descriptions of software library components. Document maps have also been created by Zavrel [41]. The AI Group at the University of Arizona has used the SOM in their "ET-Map" in categorizing the content of Internet documents to aid in searching and exploration [2]. A system that is very similar to the WEBSOM has recently been used to organize collections of scientific articles on astronomy [25,36].

5. Results

In order to test and demonstrate the scalability of the WEBSOM method, over a million documents (1,124,134) from 85 Usenet newsgroups were organized on a document map consisting of 104 040 units. The size of the document map was chosen so that on the average there would be about ten documents in each map unit, an amount which was considered convenient for browsing.

5.1. Preprocessing

The documents were preprocessed by removing non-textual information, for instance, ASCII drawings and automatically included signatures. Numerical expressions and special codes were treated with heuristic rules, and punctuation was removed.

To reduce the computational load, the words that occurred less than 50 times in the whole document collection were neglected and treated as empty slots. Since very common words such as "take", "go", "he", etc., do not differentiate between discussion topics, such words were also discarded from the vocabulary. When signatures, headers and articles ("a", "the", "an") had been removed, the average length of the documents

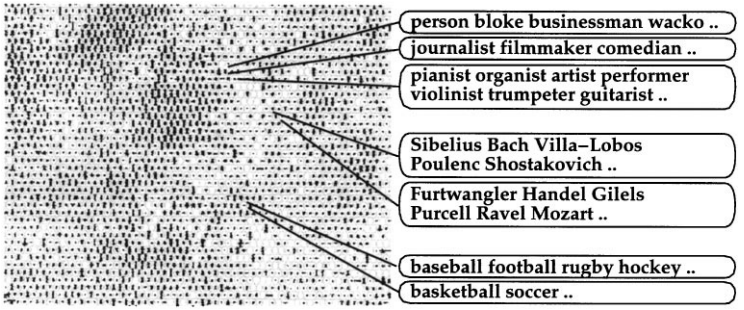


Fig. 1. A sample area of the word category map used to organize the words in the collection of a million documents. Due to the large amount of map units the contents of the units are not visible, but partial contents of a few units are highlighted in the ovals.

was 218 words. The removal of rare words reduced the number of unique words from 1 127 184 to 67 220 words, and after the set of common words was excluded the final vocabulary consisted of 63 773 words.

5.2. Word category map

A word category map of 13 432 units was organized to categorize the words. The word vectors were 180-dimensional; both the $\boldsymbol{x}_j^{(1)}$ and $\boldsymbol{x}_j^{(-1)}$ were 90-dimensional in Eq. (14). A sample area of the resulting map is illustrated in Fig. 1.

5.3. Document map

After encoding the documents as histograms on the large word category map, the 13 432-dimensional histogram vectors were mapped into a space of 315 dimensions with the random mapping method (see Section 3 for general description of the method). These 315-dimensional vectors were then used as input in computing the document map.

The computation of the document map was speeded up using methods described in detail in [16,19]. In short, the following two methods were used: (1) A smaller map of 18×45 units was computed, and a good initialization of the eventual large map of 204×512 units was *estimated* based on this small map, thus enabling considerably shorter computing time for the large map. (2) The fact that the map changes only gradually during its final fine-tuning phase can be utilized when finding the best matching units on the final map. It is possible to store with each training sample an address pointer to the location of the unit which was the winner when the sample was inputted previously. A *local search* in the neighborhood of the previous winner then yields a sufficiently good approximation to the location of the next winner. After finding the winner, the address pointer is updated. In order to guarantee that the asymptotic state is not affected by this approximation, updating of the address pointer can be performed based on a full winner search after, say, every 30 full passes through the data.

Once the document map has been computed based on some document collection, new, unseen documents can be mapped onto it rapidly without recomputing the map.

5.4. Exploring the document map

The resulting document map may be studied using a intuitive WWW-based interface (see Fig. 2). The interface serves two functions: It gives the user an overall view of the document collection by presenting a map of the whole material. Labels

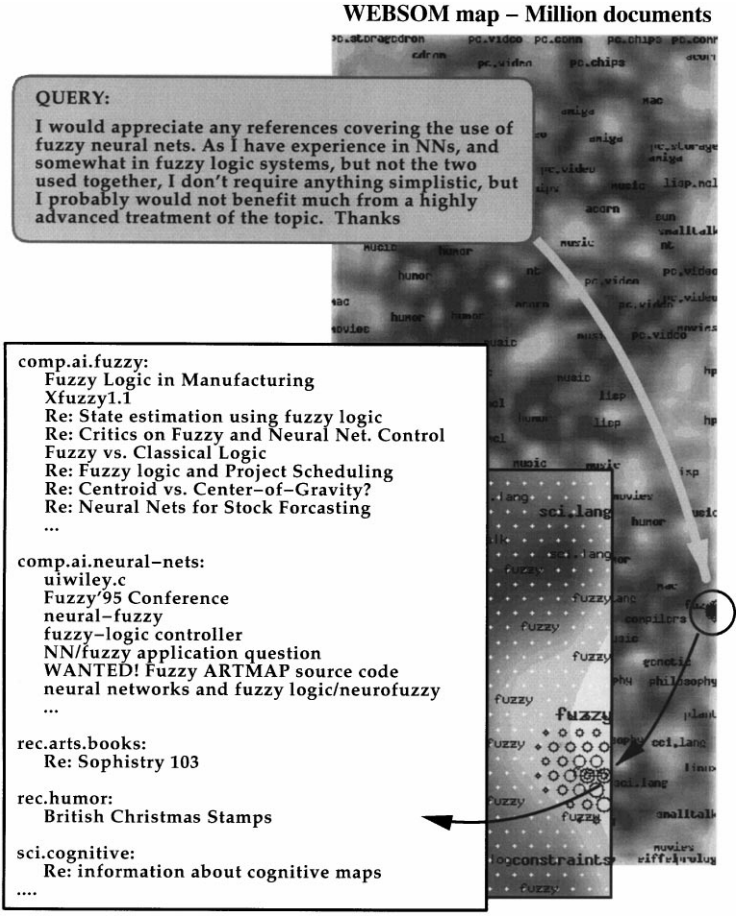


Fig. 2. An example of a content-directed search on a map of over a million documents from various Usenet newsgroups. The labels have been derived from newsgroup names and selected using an automatic method. As an alternative to navigating based on the labels one may perform searches on the map. A natural language description of interest, even a full document, can be used as a search key to find map units containing similar documents. The map units that are found can also serve as starting points for further exploration. In the example, most documents are clearly related to the query, but also some non-related documents are found, which is unavoidable when using a statistical method.

have been automatically selected and positioned on the map to describe the typical contents of the underlying area. The interface also aids the user in finding interesting information. The user may either utilize a content-directed search capability to locate interesting starting points for exploration, or simply start from an area that looks interesting.

The document map is visualized as a two-dimensional display where white dots mark map units and darkness of the colour denotes clustering in the document space: when moving on the map, on light areas the topics of the documents change slowly and dark areas correspond to abrupt changes. Zooming capabilities are necessary on large maps; in this case we used two intermediate zoom levels between the full view and the view of single map units. The map shown in Fig. 2 can be explored in the WWW address http://websom.hut.fi/websom/.

6. Conclusions

We have designed and implemented a new method called the WEBSOM for organizing vast collections of full-text documents that are available in electronic form. The WEBSOM appears to be especially suitable for exploration tasks in which the user either does not know the domain very well or has only a vague idea of the contents of the full-text database being examined. With the WEBSOM, the documents are ordered meaningfully on a document map according to their contents. Maps help the exploration by giving an overall view of what the information space looks like.

In the World Wide Web, one possible application of the method is the organization of home pages. Also electronic mail messages may be automatically positioned on a suitable map and filtered according to personal interests: relevant areas and single nodes on the map can be used as "mailboxes" that perform the automatic filtering. The method could also be used to organize official letters, personal files, research papers, library collections, newspaper articles and corporate full-text databases.

References

[1] M.W. Berry, S.T. Dumais, G.W. O'Brien, Using linear algebra for intelligent information retrieval, SIAM Rev. 37 (1995) 573–595.

[2] H. Chen, C. Schuffels, R. Orwig, Internet categorization and search: a self-organizing approach, J. Visual Commun. Image Representation 7 (1996) 88–102.

[3] S. Deerwester, S. Dumais, G. Furnas, K. Landauer, Indexing by latent semantic analysis, J. Amer. Soc. Inform. Sci. 41 (1990) 391–407.

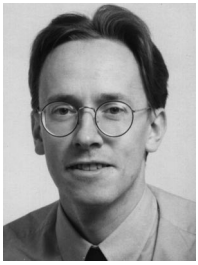
[4] S. Finch, N. Chater, Unsupervised methods for finding linguistic categories, in: I. Aleksander, J. Taylor (Eds.), Artificial Neural Networks, vol. 2, North-Holland, Amsterdam, 1992, pp. II-1365–1368.

[5] S. Gallant, W. Caid, J. Carleton, R. Hecht-Nielsen, K. Pu Qing, D. Sudbeck, HNC's MatchPlus system, ACM SIGIR Forum 26 (2) (1992) 34–38.

[6] T. Honkela, S. Kaski, T. Kohonen, K. Lagus, Self-organizing maps of very large document collections: Justification for the WEBSOM method, in: I. Balderjahn, R. Mathar, M. Schader (Eds.), Classification, Data Analysis and Data Highways, Springer, Berlin, 1998, pp. 245–252.
[7] T. Honkela, S. Kaski, K. Lagus, T. Kohonen, Exploration of full-text databases with self-organizing maps, Proc. ICNN96, Int. Conf. on Neural Networks, vol. I, IEEE Service Center, Piscataway, NJ, 1996, pp. 56–61.
[8] T. Honkela, S. Kaski, K. Lagus, T. Kohonen, Newsgroup exploration with WEBSOM method and browsing interface, Technical Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.
[9] T. Honkela, S. Kaski, K. Lagus, T. Kohonen, WEBSOM – self-organizing maps of document collections, Proc. WSOM'97, Workshop on Self-Organizing Maps, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997, pp. 310–315.
[10] T. Honkela, V. Pulkki, T. Kohonen, Contextual relations of words in Grimm tales analyzed by self-organizing map, in: F. Fogelman-Soulié, P. Gallinari (Eds.), Proc. ICANN-95, Int. Conf. on Artificial Neural Networks, vol. II, EC2 et Cie, Paris, 1995, pp. 3–7.
[11] H. Hyötyniemi, Text document classification with self-organizing maps, in: J. Alander, T. Honkela, M. Jakobsson (Eds.), Proc. Finnish Artificial Intelligence Conf. – Genes, Nets and Symbols, Finnish Artificial Intelligence Society, 1996, pp. 64–72.
[12] S. Kaski, Computationally efficient approximation of a probabilistic model for document representation in the WEBSOM full-text analysis method, Neural Process. Lett. 5 (1997) 139–151.
[13] S. Kaski, Data exploration using self-organizing maps, Acta polytechnica scandinavica, mathematics, computing and management in engineering series No. 82, Dr. Tech. Thesis, Helsinki University of Technology, Finland, March 1997.
[14] S. Kaski, Dimensionality reduction by random mapping: fast similarity computation for clustering, Proc. IJCNN'98, Int. Joint Conference on Neural Networks, Vol. 1, IEEE Service Center, Piscataway, NJ, 1998, pp. 413–418.
[15] S. Kaski, T. Honkela, K. Lagus, T. Kohonen, Creating an order in digital libraries with self-organizing maps, Proc. WCNN'96, World Congress on Neural Networks, Lawrence Erlbaum and INNS Press, Mahwah, NJ, 1996, pp. 814–817.
[16] T. Kohonen, Speedy SOM, Report A33, Laboratory of Computer and Information Science, Helsinki University of Technology, 1996.
[17] T. Kohonen, Exploration of very large databases by self-organizing maps, in: Proc. ICNN'97, Int. Conf. on Neural Networks, IEEE Service Center, Piscataway, NJ, 1997, pp. PL1–PL6.
[18] T. Kohonen, S. Kaski, K. Lagus, T. Honkela, Very large two-level SOM for the browsing of newsgroups, in: C. von der Malsburg, W. von Seelen, J.C. Vorbrüggen, B. Sendhoff (Eds.), Proc. ICANN96, Int. Conf. on Artificial Neural Networks, Lecture Notes in Computer Science, vol. 1112, Springer, Berlin, 1996, pp. 269–274.
[19] T. Kohonen, Self-Organizing Maps, 2nd extended ed., Springer, Berlin, 1997.
[20] T. Kohonen, The self-organizing map (SOM), Neurocomputing, 21 (1998), this issue.
[21] K. Lagus, Map of WSOM'97 abstracts – alternative index, Proc. WSOM'97, Workshop on Self-Organizing Maps, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997, pp. 368–372.
[22] K. Lagus, T. Honkela, S. Kaski, T. Kohonen, Self-organizing maps of document collections: a new approach to interactive exploration, in: E. Simoudis, J. Han, U. Fayyad (Eds.), Proc. KDD-96, 2nd Int. Conf. on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, 1996, pp. 238–243.
[23] K. Lagus, T. Honkela, S. Kaski, T. Kohonen, WEBSOM – a status report, in: J. Alander, T. Honkela, M. Jakobsson (Eds.), Proc. STeP'96, Finnish Artificial Intelligence Conf., Finnish Artificial Intelligence Society, Vaasa, Finland, 1996, pp. 73–78.
[24] K. Lagus, S. Kaski, T. Honkela, T. Kohonen, Browsing digital libraries with the aid of self-organizing maps, Proc. WWW5, the 5th Int. World Wide Web Conf., EPGL, 1996, vol. Poster Proceedings, pp. 71–79.
[25] S. Lesteven, P. Ponçot, F. Murtagh, Neural networks and information extraction in astronomical information retrieval, Vistas Astronomy 40 (1996) 395.

[26] X. Lin, Visualization for the document space, Proc. Visualization '92, IEEE Computer Society Press, 1992, pp. 274–281.
[27] X. Lin, Map displays for information retrieval, J. Amer. Soc. Inform. Sci. 48 (1997) 40–54.
[28] X. Lin, D. Soergel, G. Marchionini, A self-organizing semantic map for information retrieval, Proc. 14th Annual Int. ACM/SIGIR Conf. on R & D in Information Retrieval, 1991, pp. 262–269.
[29] D. Merkl, Structuring software for reuse – the case of self-organizing maps, Proc. IJCNN-93, Int. Joint Conf. on Neural Networks, Nagoya, vol III, IEEE Service Center, Piscataway, NJ, 1993, pp. 2468–2471.
[30] D. Merkl, Self-organization of software libraries: an artificial neural network approach, Ph.D. Thesis, Institut für Angewandte Informatik und Informationssysteme, Universität Wien, 1994.
[31] D. Merkl, Content-based document classification with highly compressed input data, in: F. Fogelman-Soulié, P. Gallinari (Eds.), Proc. ICANN'95, Int. Conf. on Artificial Neural Networks, vol. II, EC2, Nanterre, France, 1995, pp. 239–244.
[32] D. Merkl, Content-based software classification by self-organization, in: Proc. ICNN'95, IEEE Int. Conf. on Neural Networks, vol. II, IEEE Service Center, Piscataway, NJ, 1995, pp. 1086–1091.
[33] D. Merkl, Lessons learned in text document classification, Proc. WSOM'97, Workshop on Self-Organizing Maps, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997, pp. 316–321.
[34] D. Merkl, A.M. Tjoa, G. Kappel, Application of self-organizing feature maps with lateral inhibition to structure a library of reusable software components, Proc. ICNN'94, Int. Conf. on Neural Networks, IEEE Service Center, Piscataway, NJ, 1994, pp. 3905–3908.
[35] R. Miikkulainen, Subsymbolic Natural Language Processing: An Integrated Model of Scripts Lexicon, and Memory, MIT Press, Cambridge, MA, 1993.
[36] P. Poinçot, S. Lesteven, F. Murtagh, A spatial user interface to the astronomical literature, Astronomy Astrophys. 130 (1998) 183–191.
[37] H. Ritter, T. Kohonen, Self-organizing semantic maps, Biol. Cybernet. 61 (1989) 241–254.
[38] H. Ritter, T. Kohonen, Learning 'semantotopic maps' from context, in: M. Caudill (Ed.), Proc. IJCNN'90, Int. Joint Conf. on Neural Networks, Washington DC, vol. I, Lawrence Erlbaum, Hillsdale, NJ, 1990, pp. 23–26.
[39] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.
[40] J.C. Scholtes Neural Networks in Natural Language Processing and Information Retrieval, Ph.D. Thesis, Universiteit van Amsterdam, Amsterdam, Netherlands, 1993.
[41] J. Zavrel, Neural navigation interfaces for information retrieval: are they more than an appealing idea? Artificial Intell. Rev. 10 (1996) 477–504.

Samuel Kaski received his Dr. Tech. degree in Computer Science from Helsinki University of Technology, Finland, in 1997. He is a research associate at the Neural Networks Research Centre, Helsinki University of Technology. His main research interests are related to neural networks, especially self-organizing maps, and their applications in statistics and data mining. The common theme in most of his projects has been exploratory data analysis and visualization of structures in the data using self-organizing maps. Dr. Kaski has published several research papers connected to both the theory and applications of the methodology.



Timo Honkela obtained his M.Sc. degree in Information Processing Science at University of Oulu, Finland, in 1989, and Ph.D. degree in Computer Science from Helsinki University of Technology, Finland, in 1998. His main interests and research experience are related to artificial intelligence and natural language processing. In 1987–1989 he was responsible for the design of the semantic processing component in a project developing a natural language database interface for Finnish. In 1990–1994 he was a research scientist at VTT (Technical Research Centre of Finland). Since 1995 Dr. Honkela is with the Neural Networks Research Centre of Helsinki University of Technology conducting research related to various aspects of natural language interpretation using self-organizing maps. He is the chairman of the Finnish Artificial Intelligence Society.

Krista Lagus received her M.Sc. in Computer Science from Helsinki University of Technology, Finland in 1996, majoring in artificial intelligence and minoring in cognitive science. Her main research interests and experience is related to neural networks, especially self-organizing maps, and their application to natural language processing and data mining. Her general interests include natural language understanding both in humans and in artificial systems. Since 1995 Ms. Lagus works in the Neural Networks Research Centre of Helsinki University of Technology researching natural language processing using self-organizing maps.

The biography of **Prof. Teuvo Kohonen** can be found elsewhere in the special issue.