Pergamon

Phys. Chem. Earth, Vol. 25, No. 8, pp. 649–654, 2000

Interactive Text Retrieval Based on Document Similarities

A. Klose, A. Nürnberger, R. Kruse[1] and G. Hartmann, M. Richards[2]

[1]Institute for Knowledge and Language Processing, University of Magdeburg, Germany
[2]Max-Planck-Institut für Aeronomie, Katlenburg-Lindau, Germany

Received: 4 September 2000 – Accepted: 25 September 2000

Abstract.

In this article we present a prototypical implementation of a software tool for document retrieval which groups/arranges (pre-processed) documents based on a similarity measure. The prototype was developed based on self-organising maps to realise interactive associative search and visual exploration of document databases. This helps a user to navigate through similar documents. The navigation, especially the search for the first appropriate document, is supported by conventional keyword search methods. The usability of the presented approach is shown by a sample search.

1 Introduction

The results of the work presented in this article have been achieved as a part of a pilot project for the validation of atmospheric data, carried out at the Max-Planck-Institut für Aeronomie, Germany. The main goal of this project was to combine classical and modern documentation methods—comprising data classification, storage and retrieval—and validation methods for selected data and texts of the Earth's atmosphere, so that the data can be interactively graphically linked and the results can be displayed. The developed software functionality includes visualisation, animation and statistical processing and allows to reveal ozone trends or to ascertain differences between data sets.

In this article we focus on document retrieval from document databases. This part of the project was motivated by the limitations of standard text retrieval methods which usually just make use of specific keywords provided by the user, but neglect the capabilities of (state of the art) document preprocessing techniques (e.g. stemming) or document similarities. Furthermore, we present a prototypical implementation which has been developed as part of this project. The software prototype was designed for evaluation purposes and is not intended to replace a professional search engine. It is provided on the DUST-2 CD-ROM (Hartmann et al., 2000) which is available from the Copernicus Gesellschaft e.V. in Katlenburg-Lindau (http://www.copernicus.org). On the CD two sample datasets are provided which allow the user to validate the usability of the proposed methods for document retrieval.

The motivation of the approaches investigated in this project was to arrange documents based on their similarity in content. The arranging is based on self-organising maps and is combined with tools for interactive associative search and a visual exploration of the document database. This helps the user to navigate through similar documents.

In the following section we will briefly review the concepts of self-organising systems. In Sect. 3 we describe the implemented document pre-processing and the methods used for grouping the text documents based on different similarity measures. We present a sample search in Sect. 4 to show the usability of this tool. In Sect. 5 conclusion and possible extensions are given.

2 Self-organising maps

Self-organising maps (Kohonen, 1982) are a special architecture of neural networks that cluster high-dimensional data vectors according to a similarity measure. The clusters are arranged in a low-dimensional topology that preserves the neighbourhood relations in the high dimensional data. Thus, not only objects that are assigned to one cluster are similar to each other (as in every cluster analysis), but also objects of nearby clusters are expected to be more similar than objects in more distant clusters. Usually, two-dimensional grids of squares or hexagons are used (cf. Fig. 2). Although other topologies are possible, two-dimensional maps have the advantage of an intuitive visualisation and thus good exploration possibilities.

Self-organising maps are trained in an unsupervised manner (i.e. no class information is provided) from a set of high-dimensional sample vectors. The network structure has two layers (see Fig. 1). The neurons in the input layer correspond to the input dimensions. The output layer (map) contains

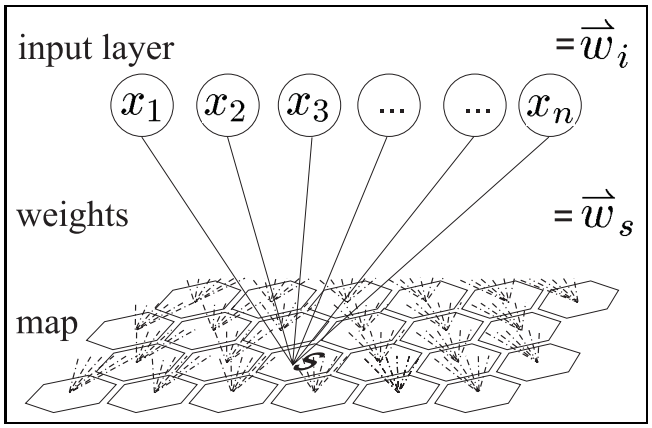

Fig. 1. Network architecture for self-organising maps

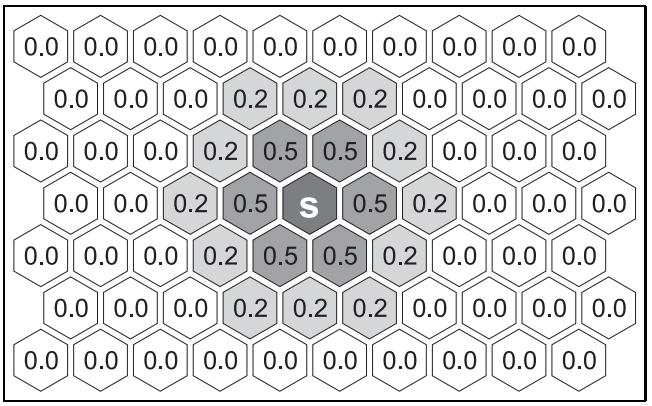

Fig. 2. Possible neighbourhood function v for increasing distances from s

as many neurons as clusters needed. All neurons in the input layer are connected with all neurons in the output layer (Fig. 1 shows only the connections between input layer and one output neuron). The weights of the connection between input and output layer of the neural network encode positions in the high-dimensional data space. Thus, every unit in the output layer represents a prototype. Before the learning phase of the network, the two-dimensional structure of the output units is fixed and the weights are initialised randomly. During learning, the sample vectors are repeatedly propagated through the network. The weights of the most similar prototype $\vec{w}_s$ (*winner neuron*) are modified such that the prototype moves toward the input vector $\vec{w}_i$. As similarity measure usually the scalar product is used. The weights $\vec{w}_s$ of the winner neuron are modified according to the following equation:

$$\vec{w}_s' = \vec{w}_s + \sigma \cdot (\vec{w}_s - \vec{w}_i),$$

where σ is a learning rate.

To preserve the neighbourhood relations, prototypes that are close to the winner neuron in the two-dimensional structure are also moved in the same direction. The weight change decreases with the distance from the winner neuron. Therefore, the adaption method is extended by a neighbourhood function v (see also Fig. 2):

$$\vec{w}_s' = \vec{w}_s + v(i,s) \cdot \sigma \cdot (\vec{w}_s - \vec{w}_i),$$

where σ is a learning rate. By this learning procedure, the structure in the high-dimensional sample data is non-linearly projected to the lower-dimensional topology. After learning, arbitrary vectors (i.e. vectors from the sample set or prior 'unknown' vectors) can be propagated through the network and are mapped to the output units. For further details on self-organising maps see (Kohonen, 1984).

3 Building a map of a document collection

The main idea of the presented approach is to use a self-organising map as a tool to arrange similar documents[1]. After training of this map, documents with similar contents are close to each other, and possibly assigned to the same neuron. So, when a user has discovered a document of interest on the map, he or she can search the surrounding area.

For the creation of the self-organising map, the documents must be encoded in form of vectors. To be suited for the learning process of the map, similar vectors must be assigned to similar documents, i.e. the vectors have to represent the document content.

As common in document retrieval, our approach is based on statistical evaluations of word occurrences. We do not use any information on the *meaning* of the words. In domains like scientific research we are confronted with a wide and (often rapidly) changing vocabulary, which is hard to catch in fixed structures like manually defined thesauri or keyword lists. However, it is important to be able to calculate significant statistics. Therefore, the number of considered words must be kept reasonably small, and the occurrences of words sufficiently high. This can be done by either removing words or by grouping words with equal or similar meaning. A basic way to do so is to filter so-called *stop words* and to build the stems of the words (Frakes and Baeza-Yates, 1992). Although these steps are quite common in publications on document retrieval, they are still rarely used in commercially available document retrieval approaches or search engines.

3.1 Stemming and filtering

The idea of stop word filtering is to remove words that bear no content information, like articles, conjunctions, prepositions, etc. Furthermore, words that occur extremely often can be said to be of little information content to distinguish between documents. Also, words that occur very seldom are likely to be of no particular statistical relevance.

Stemming tries to build the basic forms of words, i.e. strip

[1] For further information see also (Honkela et al., 1996b) and (Honkela et al., 1996a) and the website of the WEBSOM project: http://websom.hut.fi

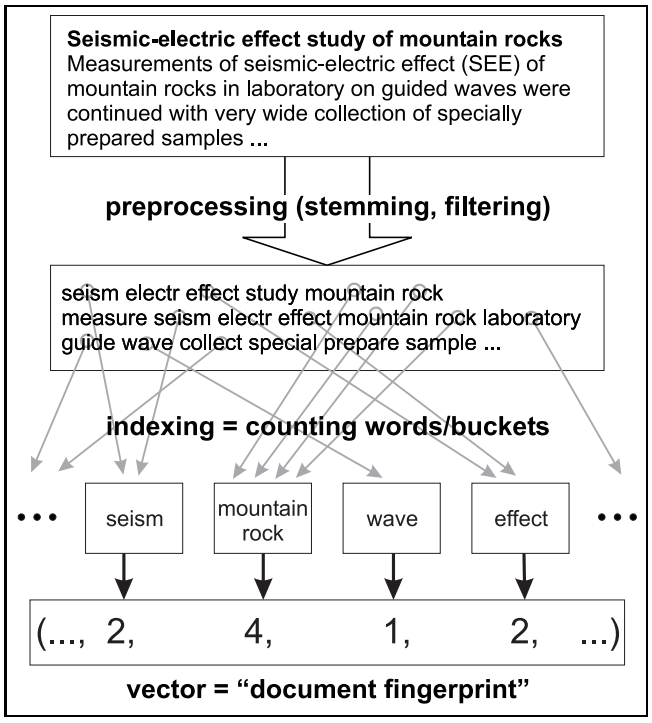

Fig. 3. Document preprocessing and coding

the plural 's' from nouns, the 'ing' from verbs, or other affixes. A stem is a natural group of words with equal (or very similar) meaning. We apply the stemming algorithm of Porter (1980), which uses a set of production rules to iteratively transform (English) words into their stems.

For the further reduction of relevant words we use two alternative approaches. The first reduces the vocabulary to a set of index words. These words are not selected manually, but automatically chosen by an information theoretic measure. The second approach is based on the work of Ritter and Kohonen (1989) and Honkela et al. (1996a). It uses a self-organising map to build clusters of similar words, where *similarity* is defined with a statistical measure over the word's context.

3.2 Selection of index words based on their entropy

For each word a in the vocabulary we calculate the entropy as defined by Lochbaum and Streeter (1989):

$$W(a) = 1 + \frac{1}{\ln(m)} \sum_{i=1}^{m} p_i(a) \cdot \ln(p_i(a)) \qquad \text{with}$$

$$p_i(a) = \frac{n_i(a)}{\sum_{j=1}^{m} n_j(a)}$$

$n_i(a)$: frequency of word w in document i
m : number of documents

Here, the entropy gives a measure how well a word is suited to separate documents by keyword search. E.g. words that occur in many documents will have low entropy. The entropy can be seen as a measure of importance of words in the given domain context. We choose a number of words that have a high entropy relative to their overall frequency (i.e. from words occurring equally often we prefer those with the higher entropy). This procedure has empirically been found to yield a set of relevant words that are suited to serve as index terms.

3.3 The word category map

This approach does not reduce the number of words by removing irrelevant words from the vocabulary, but by building groups of words which are frequently used in similar (three-word-)contexts. A self-organising map is used to find appropriate clusters of words.

To be able to use words for training of a self-organising map, the words have to be encoded. Therefore, to every word a random vector $\vec{w}$ with 90 dimensions is assigned, as recommended by Honkela (1997). This encoding does not imply any word ordering, as random vectors of dimensionalities that high can be shown to be "quasi-orthogonal": the scalar product for nearly every pair of words is approximately zero.

Then, the three-word-contexts of a word a are encoded by calculating the element-wise mean vectors of the words before $\vec{w}_{\mathrm{before}}$ and after $\vec{w}_{\mathrm{after}}$ the considered word over all documents and all occurrences of a. These mean (or expectation value) vectors $\langle \vec{w}_{\mathrm{before}} \rangle$ and $\langle \vec{w}_{\mathrm{after}} \rangle$ over the random vectors of enclosing words are used to define the context vector $\vec{w}_c$ of the considered word:

$$\vec{w}_c = (\langle \vec{w}_{\mathrm{before}} \rangle, \vec{w}, \langle \vec{w}_{\mathrm{after}} \rangle).$$

The obtained context vectors have $270 (= 3 \times 90)$ dimensions. Words a, b that often occur in similar contexts have similar expectation values and therefore similar context vectors $\vec{w}_c^{(a)}$, $\vec{w}_c^{(b)}$. The vectors $\vec{w}_c$ are finally clustered on a two-dimensional hexagonal grid using a self-organising map. Words that are used in similar contexts are expected to be mapped to the same or to nearby neurons on this so-called word category map. Thus, the words in the vocabulary are reduced to the number of clusters given by the size of the word category map. Instead of index terms, the word categories (*buckets*) are used for the document indexing.

The most apparent advantage of this approach over the index term approach is that no words are removed from the vocabulary. Thus, all words are considered in the document clustering step (Sect. 3.4). Furthermore, the word category map can be used as an expedient for the visual exploration of the document collection (Sect. 4), because one often finds related words clustered together in the same or adjacent neurons of the word category map. From these clusters the user may choose related keywords which are appropriate for a (new) keyword search to reduce (or increase) the number of considered documents. However, due to the statistical peculiarities of the approach and the rather weak semantic clues the context vectors give, there are often additional words in the clusters that stand in no understandable relation to the others. The main drawback of this approach is that the words

in one cluster become indistinguishable for the document indexing.

3.4 Generating characteristic document vectors

Fig. 3 shows the principle of the proposed document encoding. At first, the original documents are pre-processed, i.e. they are split into words, then stop words are filtered and the word stems are generated (Sect. 3.1). Afterwards the considered vocabulary is reduced to a number of groups or *buckets*. These buckets are the index words from Sect. 3.2 or the word category maps from Sect. 3.3. The words of every document are then sorted into the buckets, i.e. the occurrences of the word stems associated with the buckets are counted. Each of the n buckets builds a component in a n-dimensional vector that characterises the document. These vectors can be seen as the *fingerprints* of each document.

For every document in the collection such a fingerprint is generated. Using a self-organising map, these document vectors are then clustered and arranged into a hexagonal grid, the so-called *document* map. Furthermore, each grid cell is labeled by a specific keyword which describes the content of the assigned documents. The labeling method we used is based on methods proposed in (Lagus and Kaski, 1999). It focuses on the distribution of words used in the documents assigned to the considered grid cell compared to the whole document database. The labeled map can then be used in visual exploration of the document collection, as shown in the following section.

4 Using the maps to explore document collections

By the processing of the document database described in Sect. 3 we finally have a document map, where similar documents are grouped, and a word category map (if this approach is chosen) where the grouping of words is shown.

4.1 Software Implementation: SOMAccess

To assess the usability of this approach a software prototype has been developed as part of this project. The interactive user interface has been implemented in Java. The document preprocessing has been done by database queries and simple shell scripts. For the computation of the self-organising maps we used a software package presented in (Kohonen et al., 1996) which is publicly available from the web site http://www.cis.hut.fi/research/som-research/. Once pre-processed and learned, the indexes and maps are stored in a Microsoft Access database. During exploration, the data are accessed by the Java program via ODBC (Open Database Connectivity). In Fig. 4 an overview of the system structure is given.

As the generation of the database is only done once for a given document collection, the treatment of dynamic aspects is a critical point. Sect. 4.3 gives hints how changes of the document collection can be handled.

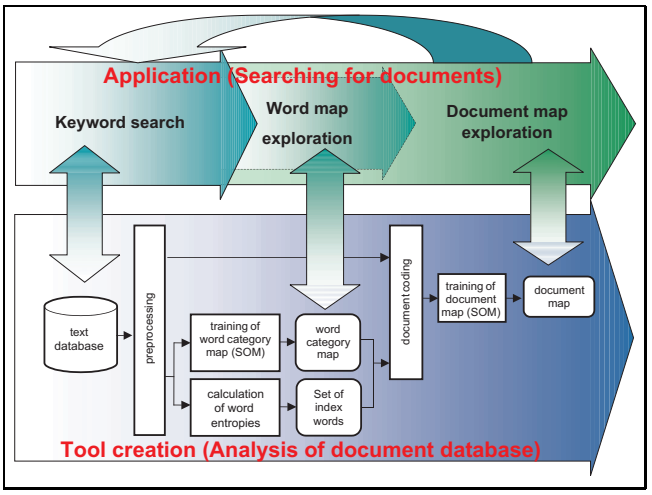

Fig. 4. System overview

4.2 Sample search

In the following, we present an example of a search to illustrate the ideas of using word category and document maps to find relevant documents. For the example we use the abstracts of the conference EGS 2000, that are provided on the DUST-2 CD-ROM.

A search usually includes several stages that may be repeated and iterated:

- Query the full text database (keyword search) to find initial interesting subsets of the document collection,
- visually inspect the document map, e.g. to see how wide the documents are spread over the map,
- browse the documents' fingerprints in the word category map to discover similarities and get ideas for further relevant keywords,
- refine the query with further keywords, and
- for potentially relevant documents, inspect documents that lie on the same or adjacent nodes of the document map.

To be more detailed, we describe these steps and the required maps and tools of the software in the following example of a search.

Full text query and refinement

For a concrete example, we assume that we are interested in groundwater and especially the problems of groundwater pollution. Therefore, we are looking for publications where these topics are discussed. Our document collection contains 6107 papers. We start with a keyword search for 'groundwater', which results in 102 matching documents. Not surprisingly, a number of the listed documents deals with different topics, like groundwater flow or the simulation of its dynamic. Therefore, we refine the search by the keyword

'pollution'. This search leads to (just) seven possibly relevant matches.

Using the word category map

The number of seven results seems to be very small, so we would like to broaden our search. On the other hand, we would like the query to be still specific. Therefore, we use the word category map. In this map we visualise the fingerprints of the matching documents. The highlighted nodes give us visual hints on which important keywords the document contains in addition to those keywords we have been searching for. Furthermore, we may find groups of documents with visually similar fingerprints (i.e. similar highlighted regions) and thus similar content. Therefore, we are supported in finding some keywords which describe the document content and which can then be used to refine the search by adding (or prohibiting) these keywords.

One of the seven documents of our refined search (i.e. paper No. 5606 entitled 'Anomalous High Pollution of a Natural Water by Fluorine and Other Elements in Novokuznetsk City (West Sibiria)') shows a marked cluster of highlighted nodes on the lower left of the map. The associated keywords in these nearby nodes contain 'pollution', 'pollut', and 'contamin'. The stemming algorithm obviously failed to automatically build the stem for 'pollution', whereas it works correctly for 'contamination', 'contaminate', etc. Therefore, we refine our search results for 'groundwater' by a disjunctive search for the three synonymuous terms. This yields 34 instead of seven matches.

Using the document map

Since we might also be interested in papers dealing with related problems we have a look at the document map. The search terms can be associated with colors. The nodes of the document map are highlighted with blends of these colors to indicate how well the documents assigned to a node match the search terms.

This feature enables the user to see how wide the results of his query are spread and thus can give him an idea, if he should further refine the search. If the highlighted nodes build 'clusters' on the map we can suppose that the corresponding search term was relevant for the neighborhood relations in the learning of the self-organising map. In this case the probability to find documents with similar topics in adjacent nodes can be expected to be higher.

In our example search, we select one of the nodes which is rather bright and thus indicates a good match. The node—labeled 'palladium'—contains the paper No. 5606 which we already found by the keyword search. Some of the other documents of this node deal with problems of, e.g., sedimentation and catchment water quality. One of these papers ('Magnetism of Technogenic Lake Sediments in Kuznetsk Alatau Mining Region', Paper No. 10179) seems to be of high interest to us, as it mentions problems of lake water pollution by cyanidation waste. Using conventional search methods we would not have discovered most of these papers, since they did not use any of the applied search terms. Furthermore, since we expect some relevant papers on the marked nodes in the document map, we are motivated to scan the list of surrounding documents. In Fig. 5 the word category maps of the mentioned abstracts are depicted.

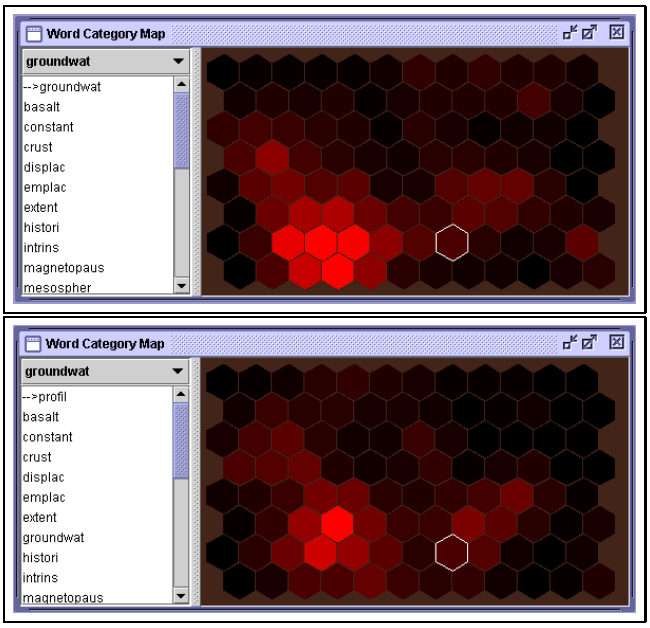

Fig. 5. Word category map of paper no. 5606 (top), which was discovered by keyword search, and the related paper no. 10179 (bottom)

4.3 Dynamical aspects

Generally, scientific research is a dynamic, evolving area. New documents will become available and obviously be of relevance to researchers. Furthermore, new topics will come up in scientific communities.

Traditional document retrieval, which is based on manually defined thesauri or fixed sets of index words, always suffered from its implicit inflexibility.

The databases provided with the DUST CD-ROM represent only a 'snap-shot-like' document collection. However, there are several ways to meet the challenges of changing document material.

First of all, keyword queries on full texts do not depend on predefined index terms or thesauri. Furthermore, as the system is built with very little user intervention and directly from the documents themselves, it can always be completely reconstructed for a changed document database.

However, this brute force approach is not always necessary, and more sophisticated methods are possible:

- For small changes we can keep the learned maps and just add the new documents and word stems to the nearest map nodes.
- If we expect extensive changes, we could re-train the document map. The learning algorithm of self organising maps is incremental. Thus, we can use the old map

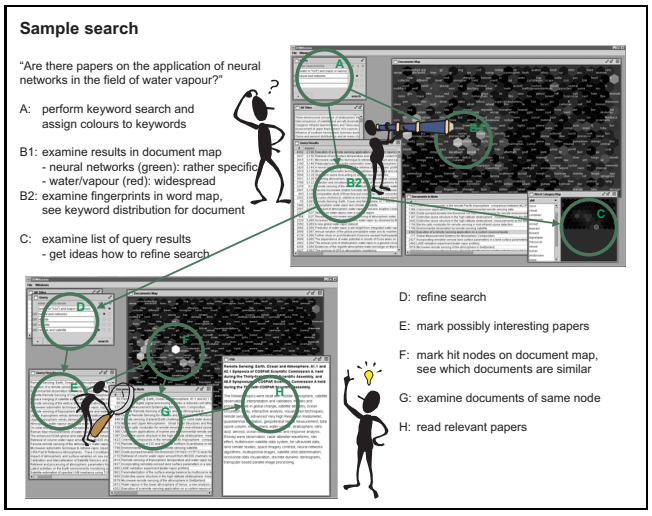

Fig. 6. Sample Search

as an initialisation and just slightly rearrange the nodes to better fit the new collection. Alternatively, we can also relearn a new, perhaps bigger document map from scratch. However, we can still use the old document encoding, i.e. the old word category map or automatically chosen set of index words.

– We should additionally relearn these vector encodings, when the changes in the collection are more severe. In the case of the word category map we have again the possibility of incremental adaptation. The index terms have to be chosen independently of the old terms. In both cases—word category map or index terms—an analysis of the changes might yield interesting hints on current developments of research topics, e.g. upcoming 'buzz words'.

5 Conclusions

The methods proposed in this article combine (iterative) keyword search with grouping of documents based on a similarity measure in an interactive environment. This enables a user to search for specific documents, but also to enlarge obtained result sets (without the need to redefine search terms) by navigating through groups of documents with similar contents surrounding the search hits. Furthermore, the user is supported in finding appropriate search keywords to reduce or increase the documents under consideration by using a word category map, which groups together words used in similar contexts.

Nevertheless, the proposed method has still some insufficiencies. The main problem is that the size of the used self-organising maps has to be defined manually. Therefore, the training process has to be done several times with modified map sizes, until an 'optimal' size has been discovered by manual inspection of the documents respectively words (number and contents) assigned to the grid cells. Especially for the word category map, it is often hard to say which size is satisfying. Therefore, methods that adapt the size of the map, e.g. growing self-organizing maps (for example Alahakoon et al. (1998)), have to be analysed in the continuation of this project.

Furthermore, to improve the clustering of words in the word category map the use of thesauri to define e.g. synonyms should be considered. Furthermore, dictionaries may be used to extend the method for multi-lingual databases. Therefore, it has to be analysed if the used 3-word-contexts may also be used for non-English documents.

Acknowledgements. The authors thank their institutes for the support and the Deutsche Zentrum für Luft- und Raumfahrt (DLR) for the financial support by the pilot study FKZ 50 EE 98038. Furthermore, they like to thank K.H. Weber, Fachinformationszentrum, Karlsruhe (FIZ) for the assistance in the STN inquiries and A. Richter, Copernicus Gesellschaft e.V. for providing the abstracts of the conference EGS 2000. Last but not least, they thank Dr. A. Noelle, Science-Softcon, for the excellent technical management.

References

Alahakoon, D., Halgamuge, S., and Srinivasan, B., A structure adapting feature map for optimal cluster representation, in *Proc. Int. Conf. On Neural Information Processing*, pp. 809–812, Kitakyushu, Japan, 1998.

Frakes, W. B. and Baeza-Yates, R., *Information Retrieval: Data Structures & Algorithms*, Prentice Hall, New Jersey, 1992.

Hartmann, G. K., Nölle, A., Richards, M., and Leitinger, R., Data utilization software tools 2 (DUST-2 CD-ROM), Copernicus Gesellschaft e.V., Katlenburg-Lindau, ISBN 3-9804862-3-0, Website: http://www.copernicus.org/EGS/EGS.html), 2000.

Honkela, T., *Self-Organizing Maps in Natural Language Processing*, Ph.D. thesis, Helsinki University of Technology, Neural Networks Research Center, Espoo, Finland, 1997.

Honkela, T., Kaski, S., Lagus, K., and Kohonen, T., *Newsgroup Exploration with WEBSOM Method and Browsing Interface, technical report*, Helsinki University of Technology, Neural Networks Research Center, Espoo, Finland, 1996a.

Honkela, T., Kaski, S., Lagus, K., and Kohonen, T., Exploration of full-text databases with self-organizing maps, in *Proc. International Conference on Neural Networks (ICNN'96)*, Washington, 1996b.

Kohonen, T., Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, *43*, 59–69, 1982.

Kohonen, T., *Self-Organization and Assoziative Memory*, Springer-Verlag, Berlin, 1984.

Kohonen, T., Hynninen, J., Kangas, J., and Laaksonen, J., *SOM_PAK: The Self-Organizing Map Program Package. Technical Report A31*, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.

Lagus, K. and Kaski, S., Keyword selection method for characterizing text document, in *Proc. International Conference on Artificial Neural Networks (ICANN'99)*, Edinburgh, UK, 1999.

Lochbaum, K. E. and Streeter, L. A., Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval, *Information Processing and Management*, *25(6)*, 665–676, 1989.

Porter, M., An algorithm for suffix stripping, *Program*, *14(3)*, 130–137, 1980.

Ritter, H. and Kohonen, T., Self-organizing semantic maps, *Biological Cybernetics*, *61(4)*, 241–254, 1989.