Ma 110

ACTA POLYTECHNICA SCANDINAVICA

MATHEMATICS AND COMPUTING SERIES No. 110

Text Mining with the WEBSOM

KRISTA LAGUS

Helsinki University of Technology
Neural Networks Research Centre
P.O.Box 5400
FIN-02015 HUT, Finland

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 11th of December, 2000, at 12 o'clock noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science

ESPOO 2000

Lagus, K., **Text Mining with the WEBSOM**. Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 110, Espoo 2000, 54 pp. Published by the Finnish Academies of Technology. ISBN 951-666-556-X. ISSN 1456-9418. UDC 004.032.26:025.4.03:004.5

Keywords: Self-organizing map, document maps, visual user interfaces, information exploration, text retrieval, large text collections.

ABSTRACT

The emerging field of text mining applies methods from data mining and exploratory data analysis to analyzing text collections and to conveying information to the user in an intuitive manner. Visual, map-like displays provide a powerful and fast medium for portraying information about large collections of text. Relationships between text items and collections, such as similarity, clusters, gaps and outliers can be communicated naturally using spatial relationships, shading, and colors.

In the WEBSOM method the self-organizing map (SOM) algorithm is used to automatically organize very large and high-dimensional collections of text documents onto two-dimensional map displays. The map forms a document landscape where similar documents appear close to each other at points of the regular map grid. The landscape can be labeled with automatically identified descriptive words that convey properties of each area and also act as landmarks during exploration. With the help of an HTML-based interactive tool the ordered landscape can be used in browsing the document collection and in performing searches on the map.

An organized map offers an overview of an unknown document collection helping the user in familiarizing herself with the domain. Map displays that are already familiar can be used as visual frames of reference for conveying properties of unknown text items. Static, thematically arranged document landscapes provide meaningful backgrounds for dynamic visualizations of for example time-related properties of the data. Search results can be visualized in the context of related documents.

Experiments on document collections of various sizes, text types, and languages show that the WEBSOM method is scalable and generally applicable. Preliminary results in a text retrieval experiment indicate that even when the additional value provided by the visualization is disregarded the document maps perform at least comparably with more conventional retrieval methods.

© All rights reserved. No part of the publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author.

PREFACE

This work has been carried out in the Neural Network Research Centre, Helsinki University of Technology during 1995-2000 as part of a project called WEBSOM, led by Academician Teuvo Kohonen. The work was mainly financed by the Academy of Finland with funds directed to research in the Neural Networks Research Centre. Additional support came from Tekniikan Edistämissäätiö and from Jenny and Antti Wihuri foundation.

I am indebted to Academician Teuvo Kohonen for the opportunity to work in the WEBSOM project. The project obtained a solid foundation from his profound and extensive earlier work on the self-organizing map algorithm and its applications. His active participation created a stimulating environment for learning about ambitious scientific research. I wish to thank him also for his general support and guidance over the years.

Warm thanks are due to Academy Professor Erkki Oja for supervising my thesis and graduate studies. His support was invaluable especially in the latest stages of the work.

In starting the whole WEBSOM project and at several points on the way Professor Timo Honkela's vision and belief in the approach were central ingredients in the success of the project. His insight on and enthusiasm about natural language resonated strongly with my interests in the direction. For all that, as well as his encouragement and support, I thank him.

Professor Samuel Kaski has been an important role model as well as a colleague throughout the project. I wish to thank him for his emotional support, his scientific and practical advice, and most valuable of all, his to-the-point criticism, all of which I have greatly appreciated and would have been the poorer without.

I also wish to thank the other co-authors for their input in the publications: Antti Ahonen, Jukka Honkela, Vesa Paatero, Antti Saarela, and Jarkko Salojärvi.

I am grateful for the constructive criticism and comments on various versions of the manuscript given by Teuvo Kohonen, Erkki Oja, Samuel Kaski, and Timo Honkela, as well as for the comments of the reviewers, Professor Pasi Koikkalainen and Dr. Dieter Merkl.

Over several years I have had many thought-provoking discussions about various aspects of cognition and language processing with fellow students and teachers, particularly those involved in the departments of Psychology, General Linguistics, and Finnish and English languages at the University of Helsinki, as well as my study and research environments at HUT—thank you all. The warm, gay, and intellectually inspiring atmosphere in the CIS laboratory of HUT has been an excellent arena for scientific work as well as for interesting discussions. For that, I wish to thank the whole personnel of the laboratory.

Finally, I would like to thank all my friends from various contexts for many enjoyable moments, my parents especially for their encouragement, and my husband Henri for his patience and support during the writing of this thesis.

Espoo, November 2000

Krista Lagus

Contents

LIST OF ABBREVIATIONS

BMU	Best-matching unit (in SOM training)
CGI	Common gateway interface
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol
ICA	Independent component analysis
IR	Information retrieval
KDD	Knowledge discovery in databases
LSI	Latent semantic indexing
MDS	Multidimensional scaling
PCA	Principal component analysis
RP	Random projection
SOM	Self-organizing map
SVD	Singular value decomposition
TM	Text mining
VSM	Vector space model
WCM	Word category map
WWW	World wide web
idf	inverse document frequency
tf	term frequency

1 INTRODUCTION

Large quantities of textual data available for example on the Internet pose a continuing challenge to applications that help users in making sense of the data. Search engines specialize in locating specific documents in answer to well-defined information requests. However, fulfilling a vague information need regarding an unknown domain, or obtaining an overview of a topic or a domain is very hard. Furthermore, when the answers sought relate to a set of documents instead of a single document, or when unexpected patterns or trends should be identified, the information need is better served by methods enabling a combination of visualization and interactive exploration.

In *data exploration* the purpose is to assist the user in familiarizing herself with a large collection of data, for example, by visualizing aspects of the data collection and by enabling browsing and navigation in the data space in some meaningful way. The difference between searching and exploration is much like the one between having to ask in a store for the items from a salesperson, as opposed to walking among the shelves, and picking up whatever seems desirable. Naturally, these means of finding interesting items are complementary: sometimes one wishes to browse, and at other occasions to ask the shop personnel for help.

In recent years neural networks have been successfully applied to a variety of data analysis problems on complex data sets. However, a large portion of the research has concentrated on small or medium sized data sets consisting of numerical data or natural signals. Consequently the methods have been well-studied and developed especially for small-scale problems dealing with low-dimensional data. However, as the computing power has increased it has become possible to tackle much larger problems.

This thesis describes work that has been carried out to develop an automatic method called WEBSOM ([40], [41], [42], [37], [39], [43], [55], [51], [52], [53], [59], [54], [67], [64], [65], [68], [73], [72], Publications 1–8) that enables easy exploration of very large collections of text documents. In WEBSOM an unsupervised neural network algorithm, namely the self-organizing map (SOM), developed by T. Kohonen [61, 62], is applied to automatically organize large collections of text documents onto a two-dimensional display called the map. The method places documents on regularly spaced map grid points where similar documents are generally found near each other. The resulting map can be browsed with a WWW-based exploration interface. Zoom operations can be used to focus on a detailed view of a sub-collection, and further zooming brings to view individual documents. Label words positioned on the map display portray properties of the underlying map area. In addition, a search facility provides a means for describing a specific interesting topic and for finding a suitable starting point for exploration.

The thesis contains a detailed treatment of the visualization and user interaction aspect of the WEBSOM as well as examines the possible ways of utilizing document maps to provide an intuitive user interface for accessing collections of textual data. Furthermore, an overview of the WEBSOM method and the project is presented and an attempt is made to give an introduction to the more general research context of the work, namely the field of text mining, with an emphasis on visual methods.

1.1 Contributions and structure of this thesis

Following are the main contributions of this thesis:

- An overview is presented of the research context of the WEBSOM method, namely text mining, with an emphasis on the visual and exploratory aspects that have received less attention in the mainstream of research, focused on the searching paradigm. It is argued that WEBSOM and similar visual approaches are indeed an improvement over the search-centered mainstream approaches, since they provide a natural framework for all the major text mining tasks by offering a combination of visualization, exploration, and searching.

- The visual text exploration paradigm, especially the map metaphor, is very recent. As a result, neither the possibilities nor the difficulties in navigation of vast text collections using visual landscapes have been fully explored. By discussing choices made in an implemented system various challenges regarding visualization of and navigation in document landscapes can be introduced. Furthermore, the demonstration interfaces provide a common framework for conveying and discussing additional ideas regarding map-based visualization and navigation.

- A method is introduced for utilizing document maps in information retrieval. It is shown with a standard test material that not only can the document maps be used for exploration and visualization, but also for successfully speeding up information retrieval, and for improving retrieval results compared to standard methods on noisy data.

The structure of this thesis is as follows: In Section 2 the field of text mining, including the central tasks and paradigms, is introduced. Section 3 first overviews the adaptive learning approach, then discusses properties of natural language that affect the problem of representing text, and finally presents various document representations used in text mining approaches, with an emphasis on distance-based methods. Section 4 provides a concise overview of the WEBSOM method including its evolution since the start of the project. Section 5 explores the various possible uses of document maps in tasks related to text mining.

1.2 List of publications and the author's contributions

1. Lagus, K., Kaski, S., Honkela, T., and Kohonen, T. (1996). Browsing digital libraries with the aid of self-organizing maps. *Proceedings of the Fifth International World Wide Web Conference WWW5, May 6–10*, Paris, France, pp. 71–79.

2. Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. (1996). Self-organizing maps of document collections: a new approach to interactive exploration. In Simoudis, E., Han, J., and Fayyad, U., editors, *Proceedings of the Second International Conference on Knowledge Discovery & Data Mining (KDD'96)*, pp. 238–243. AAAI Press, Menlo Park, CA.

3. Lagus, K. (1998) Generalizability of the WEBSOM method to document collections of various types. In *Proceedings of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, vol. 1, pp. 210–214, Verlag Mainz, Aachen, Germany.

4. Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. (1998). WEBSOM—self-organizing maps of document collections. *Neurocomputing*, vol. 21, pp. 101–117.

5. Lagus, K. and Kaski, S. (1999) Keyword selection method for characterizing text document maps. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN'99)*, vol. 1, pp. 371–376. IEE Press, London.

6. Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. (1999). WEBSOM for textual data mining. *Artificial Intelligence Review.* vol. 13, issue 5/6, pp. 345–364. Kluwer Academic Publishers.

7. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., and Saarela, A. (2000). Self organization of a massive text document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, vol. 11, pp. 574–585.

8. Lagus, K. (2000). Text retrieval using self-organized document maps. Technical Report A61, Helsinki University of Technology, Laboratory of Computer and Information Science. ISBN 951-22-5145-0.

The WEBSOM method has been developed by a team of several people since the onset of the project in 1995. In particular, two doctoral theses have been published that partially consist of research on the method [37, 52].

The original idea of using a two-stage SOM architecture for organizing document collections was due to Prof. Timo Honkela. Later it became evident that more suitable solutions could be discovered for large collections. The project for developing the new methods was led by Academician Teuvo Kohonen and the software development was supervised by Prof. Samuel Kaski. In particular, the speedups that allowed the application of the method to very large collections are due to T. Kohonen and to S. Kaski. The ideas mainly due to the current author concern the design and implementation of the exploration interface and the public demonstrations, the labeling method described in Publication 5, and the application to information retrieval in Publication 8. Many other ideas and details, the implementation, and the experiments were developed as a team, and it is not possible to give a full account of the detailed contribution of each team member.

Publication 1 presents the exploration interface in detail. The initial form of the WEBSOM method is described in Publication 2 and applied to organizing and exploring a collection of articles from a Usenet discussion group. The current author designed and implemented the exploration interfaces.

The applications of the method to various types of documents and to collections of different sizes are discussed in Publication 3. The experiments regarding the collection of patent abstracts and of the Finnish news articles, as well as creation of the demonstrations were due to the current author.

Publication 4 describes the state-of-the-art of the WEBSOM in 1998, including some speedup methods. The current author carried out the experiments and programming related to the magnification of the maps and the user interface.

Publication 5 introduces an automatic method for characterizing clusters of text and document map areas with descriptive words. An early version of the method was developed jointly with S. Kaski, whereas the final version as well as the experiments were designed and implemented by the current author.

Publication 6 explores the various ways of utilizing document maps in text data mining. The current author created the document map shown in the exploration and search example, implemented the search facility, and designed and carried out the filtering experiment.

Publication 7 describes the results of a team effort on speeding up the methods and programs in order to construct a very large document map of seven million patent abstracts. The speedups in SOM processing were developed by T. Kohonen and S. Kaski. Principal responsibility areas of the current author were the design and creation of the efficient and scalable exploration interface, as well as the design and supervision of the implementation of fast versions of the labeling method, implemented by Antti Saarela, and of the keyword search engine, implemented by Vesa Paatero.

In Publication 8 a method is proposed for using document maps for text retrieval. When comparing retrieval performance on the CISI reference collection the proposed method is found to perform better than either the standard vector space model or the LSI.

2 TEXT MINING

With technological advances our possibilities to collect, generate, distribute and store text data have grown fast. Nowadays virtually anyone or any institution within the technologically developed world can easily and with little cost become an information provider for an unlimited audience. Consequently, instead of rare pieces of valuable texts, we are faced with a vast amount of textual data of unknown value.

As a result, the former methods of managing the texts, such as libraries and hierarchies organized and cataloged by human effort, have become both inadequate and too expensive to perform and to maintain for the majority of the available data. The use of automatic methods, algorithms, and tools for dealing with large amounts of data, especially of textual data, has become necessary. Attempts to solve particular aspects of this general problem can be loosely described as efforts in *text mining*.

Text mining can be viewed as a specific field of *data mining*: "Data mining is the analysis of (often large) observed data sets to find unsuspected relationships and to summarize the data in novel ways which are both understandable and useful to the database owner [28]". General overview of the field can be found in [18, 28]. The data mining field is closely related to exploratory data analysis ([112], for a recent account related to the present work see [52]) and to knowledge discovery in databases[1].

Tasks that a data mining system should help with include

- organizing, clustering and classifying data,
- creating overviews and summaries,
- identifying trends and changes across time,
- identifying dependencies and unsuspected relationships in the data,
- providing other tools and indicators for specific decision-making tasks, and
- visualizing properties of individual data items, of collections of data, and of relationships between data items and collections.

2.1 Information needs and tasks related to texts

The goal of a text mining system is to aid the user in fulfilling his/her *information need*. In some cases, a specific question needs to be answered, or a certain document to be found, whereas in other cases an overview of a topic is desired. At other times, the need is just to merely find "something interesting", or to obtain a general understanding of "what is

[1]Some view data mining as the modeling step of the more comprehensive KDD process (e.g. [18]), whereas others use the terms more or less as synonyms.

out there", or to find unexpected patterns or other new information. Furthermore, the need may be only vaguely understood by the user, and in some cases difficult to express in natural language.

The major tasks related to various information needs could be described as *searching*, *browsing* and *visualization*.

Searching. In the searching approach the user specifies an information request in terms of a *query* and asks the system to locate individual documents that correspond to the query. The Internet search engines [74] are a familiar example of tools that specialize in this task.

In the search paradigm a very modest form of text mining is performed, namely *information access*. It is supposed that the user already knows rather clearly what is to be found, and is well versed in expressing her information need. However, the need may be vague, the domain unknown, and the appropriate, specialized vocabulary hard to come by[2].

Browsing. In browsing, the user navigates in the collection of text, e.g. via links between individual documents like in the WWW (browsing a hypertext or a sparsely connected graph), or via some hierarchical structure such as the contents section of a book or the *Yahoo!* which is a hierarchical, manually constructed directory of WWW sites (structure guided browsing), or via a flat organization such as a points on the display that represent documents (flat browsing).

The browsing approach allows for the information need to be more vague or unconscious, since no explicit description of the need is required. Instead, the need is implicitly communicated via the choices made in browsing, such as the links followed.

In both searching and browsing the background assumption is that the information need of the user can be addressed by individual documents that the user should read. However, when the need is either very vague, or very general, providing access to even the most appropriate individual documents might not fulfill the need. In such cases, summary-like information might be more appropriate and useful.

Visualization. In visualization of information something familiar is used as a means for illustrating something yet unfamiliar. As pointed out, there exist information needs that require assessing and conveying similarities, differences, overlaps, and other relationships *between collections of documents*. As an example, one might wish to find out what is the relationship between a familiar set of documents, e.g. personal files or familiar mailing list, to a yet unfamiliar collection, e.g. a Usenet discussion group. Using suitable visualizations intricate relationships between large collections of items can be communicated fast and intuitively.

Shneiderman identifies the following seven major tasks that a visual and interactive information exploration system should address [108]:

- Gain an overview of the entire collection
- Zoom in on items of interest
- Filter out uninteresting items

[2]This is often called *the vocabulary problem*.

- Select an item or group and get details when needed
- View relationships among items
- Keep a history of actions to support undo, redo and progressive refinement
- Allow extraction of sub-collections and the study of their properties

It is likely that the most useful tools for text mining will in the future encompass all of the above aspects, providing a variety of means to *explore* large collections of text by enabling a seamless alternation between visualization, browsing, and searching.

2.2 Information retrieval

The oldest and most established sub-field of text mining is *information retrieval (IR)* [101, 104, 3] which deals with "the representation, storage, organization, and access of information items" [3]. The research in the field focuses on the search problem, i.e. on the situation where it is assumed that the information need can be described explicitly and adequately.

The core tasks performed by any information retrieval system are indexing text and providing means to search for relevant documents from the text collection. Indexing consists usually of the identification of index keys (e.g. terms) and the construction of a data structure called the *index* that points from the index keys to specific locations in the running text. A typical search consists of formulation of a query based on information obtained from the user after which the IR system attempts to find documents that are relevant to the query, and returns them to the user. The query may be simply the set of words written by the user, or concept-space techniques or relevance feedback may be utilized for query expansion and refinement (the process of query construction and expansion has been studied e.g. in [60]).

2.2.1 Classical retrieval models

An overview of retrieval models can be found, e.g., in [3]. The three classical models are briefly described below.

Boolean model. In Boolean retrieval a document is represented by a set of index terms that appear in the document. A query consists of a set of index terms combined with Boole's operators. The model is binary, i.e. the frequency of a term has no effect. In this model the semantics of the query is well-defined—each document either fulfills the Boolean expression or does not[3]. Due to its uncomplicated semantics and the straightforward calculation of results using set operations, the Boolean model is widely used e.g. in commercial search tools.

However, also the problems of the Boolean model are well-understood: (1) Formulating a suitable query, i.e., the selection of appropriate query terms is difficult, especially if the domain is not well known. (2) The size of the output cannot be controlled: the result set may as easily contain zero or thousands of hits. Furthermore, without a concept of "partial match", one cannot know what was left out of the query definition. (3) Since there is no gradedness of matching, ordering results according to relevance is not possible.

[3]The boolean retrieval model is also called the *exact match* approach.

Vector space model. The vector space model or VSM, introduced by Salton (see e.g. [105, 102]), encodes documents in a way suitable for fast distance calculations. In short, each document is represented as a vector in a t-dimensional space, where t is equal to the number of terms in the vocabulary. In this model the problem of finding suitable documents to a query becomes that of finding the closest document vectors for a query vector, either in terms of distance or of angle. Furthermore, the model allows straightforward application of a number of general data processing methods and algorithms. Variants of the VSM, or more generally the family of distance- or projection-based models underlie the research in many modern information retrieval systems.

Probabilistic models. The probabilistic retrieval model makes explicit the Probability Ranking Principle that can be seen underlying most of the current IR research [80]: For a given query, estimate the probability that a document d belongs to the set of relevant documents and return documents in the order of decreasing probability of relevance[4]. The key question is, how to obtain the estimates regarding which documents are relevant to a given query. One may e.g. start with an initial guess for the sets of relevant and irrelevant documents, and recursively improve this estimate by use of some simplifying assumptions and local optimization. In the original, *Binary Independence Retrieval model*, term occurrences are considered binary and terms are assumed independent. Given these assumptions a probabilistic weighting can be derived for index terms, and utilized similarly as the weighting schemes in the vector space model.

Recently, other kinds of probabilistic models, such as Bayesian Inference Networks have been utilized for information retrieval [7, 1]. In such models, the relationships between documents and queries are described as Bayesian nets. The Bayesian approach enables principled combination of information from various sources during the retrieval process.

2.2.2 Evaluation of retrieval performance

The performance evaluation of an IR system is based on the notion of relevance: if a document matches the *information need* of the user, it is considered to be relevant to the query produced by the user. The quality of retrieval of an IR system can be measured if there is, for some text collection, a set of queries and their respective relevant documents, preferably chosen manually by experts. The basic evaluation measures are the following:

$$\text{precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}} \tag{1}$$

$$\text{recall} = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}} \tag{2}$$

It is not immediately clear what the relationship between a natural language query and a document should be. In some cases, the user would like to ask specific questions such as "What was the lowest price of raw oil last week?", in other cases, specifying an interesting topic, such as "Nokia cellular phones", might be enough. The IR system should therefore determine the user's underlying information need and represent it as a query.

[4]The relevance of a document is considered independent of previously returned documents.

2.2.3 A critique of the search approach

In the search paradigm a very modest form of text mining is performed, namely *information access*. It is supposed that the user already knows rather clearly what is to be found, and is well versed in expressing his/her information need. However, the domain may be unknown, and the appropriate, specialized vocabulary hard to come by.

When the information need is only vaguely understood, even providing access to the most appropriate *individual* documents might not fulfill the need. Presenting summary-like information as well as displaying relationships between sets of documents might then be more appropriate.

The lists of results provided by many IR engines allow displaying only one-dimensional information, e.g. the estimated relevance of each document. As a consequence, thematically similar items may be far apart, making it difficult for the user to form a summary of the results and to discard irrelevant ones. In addition, the lists do not in any way support relating the set of results to the rest of the available collection, or to any other meaningful framework.

Finally, as Hearst [33] points out, information retrieval does not attempt to conclude or summarize existing information, nor to discover unsuspected information – it only provides access to an existing piece of text. Obtaining either more general information such as an overview of a field or surprising information such as unsuspected trends and patterns is better achieved by other approaches, such as text visualization and exploration.

2.3 Text visualization and exploration

In the past years graphical operating systems and color displays have become standard equipment, and programming tools and technologies for building highly graphical and interactive applications, even virtual reality technologies, have been created. Also research on data visualization and exploratory data analysis methods has flourished, providing methods and tools capable of illustrating properties and relationships of complex data sets graphically (e.g. [114]). Moreover, the statistical approach to representing text items has brought text mining problems within the reach of standard data exploration methodology.

The need has been identified for more intuitive and cognitively less demanding methods of dealing with the so-called information overload. Furthermore, the research in cognitive science on the processes of perception in humans has increased our knowledge of the human perceptual apparatus. Although text is read and often also written in a linear fashion, in principle much more information could be communicated rapidly using a more visual medium in which parallel processes automatically group features and select or suppress items. The obvious reason for this is that people are used to interacting with the visuo-spatial world in real time even much before they learn to read and write. Naturally, traits that evolution has enabled humans with should be taken full advantage of in presenting information. The so-called ecological approach to text visualization (see e.g. [117])—part of the more general ecological paradigm—takes an even more extreme position by suggesting that the task of visualization should be turned around: one should start from the visual and spatial metaphors that are natural to the human perceptual system (natural landscapes, stars in the night sky, rivers, etc.) and then try to find out how these metaphors can be used to communicate interesting properties of texts and other data to the user.

Quantitative information has for long been presented using graphical means (see e.g. [111]). Information can be conveyed visually using a combination of points, lines, symbols, words, colors, and intensity of shading. In particular, the use of graphics can help make sense of large and complex data sets that cannot be managed in any other way. If spatial proximity is used to convey similarity of items (e.g. using scatter plots), information regarding clusters, gaps, outliers and patterns is communicated at a glance, and summary-like information is derived automatically by the the perceiver.

2.3.1 Methods and tools

More recently, a paradigm shift has begun in text mining from the search-oriented approach towards a spatial presentation of and interaction with textual information (see e.g. [77, 9, 32], and Publication 6). Properties of large sets of textual items, e.g., words, concepts, topics or documents, can be visualized using one-, two- or three-dimensional spaces, or networks and trees of interconnected objects [108]. Time-related and other dynamic properties may be conveyed using *time-lines*[5] or dynamical changes on the display. Intensity of a property can be depicted e.g. with intensity of shading or with the size of marker signs. Furthermore, in the spatial domain interactive operations such as selection of a subset or a single item for detailed view become cognitively simple tasks that can be performed e.g. by pointing, clicking and dragging with the mouse.

Semantic similarity and other semantic relationships between large numbers of text items have been portrayed using proximity e.g. in visualizations based on the Spire text engine [117, 110], and in document maps organized with the SOM (cf. Sections 4 and 4.5), using colored arcs in Rainbows [29], with color coordination of themes in the ET-map [11, 92] and in colored time-lines of themes in ThemeRiver [110]. The visual metaphor of natural terrain has been used in visualizing document density and clustering e.g. in ThemeView [117], in WEBSOM (Sec. 4), and in a map of astronomical texts [75, 95].

A fish-eye projection[6] has been used for viewing and browsing large graphs with a detailed view of the focus of interest, and simultaneously a less detailed overall view [106].

WWW connectivity has been visualized in Narcissus [34] using clusters connected by lines that form visual graph structures, and as a relief landscape visualization in [26] where the organization of the landscape was obtained with the SOM algorithm (cf. Section 4.1).

Influence diagrams between scientific articles have been constructed based on citations and subsequently visualized as trees or graphs in BibRelEx [6]. Citeseer [5] (www.citeseer.com) offers a browsing tool for exploring networks of scientific articles through citations as well as both citation- and text-based similarity between individual articles. Searching is used to obtain a suitable starting-point for browsing.

Term distributions within documents retrieved by a search engine have been visualized using TileBars [31].

2.3.2 Specialized application domains.

An individual user usually does not wish to simultaneously interface with all the available information in the world. Most users have at least a vague idea of the potential thematic domain (e.g. sports vs. politics) or of the information type or style (scientific articles vs. news stories vs. recreational material) that they are interested in at a specific time. The information needs may also vary according to the type of material. Providing tools, interfaces and applications for accessing specialized collections is thus a feasible way to

[5] In a time-line visualization time is depicted by one axis, and the other is used for conveying some other property, e.g. volume of articles in a certain topical category appearing at different times.

[6] A fish-eye camera lens magnifies the focus of interest while objects further away appear smaller.

attack the information flood problem in a modular fashion. It is also viable from the point of view of information providers, who may for many reasons naturally specialize in a certain domain or type of information. Examples of application areas to which systems have been specifically designed include bibliographical data mining and search (e.g. [6, 5]) and mining of medical texts (e.g. [46, 33]).

3 REPRESENTING TEXT DOCUMENTS

Most problems with text, or with any data, can be stated as that of finding a suitable representation, or a *model*, for the available data using the existing resources for a limited time, so that the subsequent performance of the model meets the requirements of both quality and speed.

3.1 Statistical learning from data

Obtaining a model by learning or estimation from data includes the following steps:

1. Encoding—an initial data encoding is chosen, either based on the intuitions of an expert or by maximizing some objective criteria that reflect interesting properties of the data with regard to the purpose of modeling. This step may contain stages such as *feature selection*, i.e. the selection of a small number of informative features from a large set of possible ones, *weighting or scaling of features* to better reflect estimated importance or "natural scaling" of some measured properties, and *dimensionality reduction*, the application of some preliminary method to reduce the dimensionality of the data encoding. If relevant information is discarded at this stage, it cannot be re-invented later. On the other hand, if the initial data encoding contains too much irrelevant information or *noise* the later search for a good model becomes difficult or time consuming, and interesting properties of the data may be lost amidst the noise.

2. Estimation—a learning algorithm or an estimation method is applied to obtain a model for the data, usually by maximizing some objective criteria. This stage can be viewed as the search for a suitable model from a large family of possible models, called the *model space*. Various learning algorithms differ in the space of models they consider, the search strategy employed, as well as their resource allocation. The success of particular model estimation algorithm is considerably affected by the data encoding used, and vice versa, what is the most suitable encoding may depend on the modeling algorithm.

3. Evaluation—the model is interpreted or evaluated for example by representing previously unseen data or by measuring how well some specific task, such as prediction or classification, can be performed with the model.

A profound textbook account of the statistical and adaptive learning (including neural networks) and statistical learning theory is given in [12]. Various neural networks models in particular are considered e.g. in [30].

3.2 Natural language texts

In natural language understanding and generation at least the following types of knowledge are relevant (see e.g. [2, 8]): (1) *morphological knowledge*—knowledge of word structure, word forms, and inflections, (2) *syntactic knowledge*—the structural knowledge of the roles of words and how words can be combined to produce sentences, (3) *semantic knowledge*—what words mean irrespective of context, and how more complex meanings are construed by combinations of words, (4) *pragmatic knowledge*—the knowledge of language use in different contexts, and how the meaning and interpretation is affected by context, (5) *discourse knowledge*—how the immediately preceding sentences affect the interpretation of the next sentence, and (6) *world knowledge*—the general knowledge of the domain or the world that the natural language communication relates to.

As Allen points out, these levels are more appropriately viewed as characterizations of knowledge rather than separate, distinct classes of knowledge. Solving a single task in natural language understanding or generation often requires knowledge of several different types [2].

Traditional approaches to implementing language representations in computers include rule-based models created manually by experts, and based on theories of linguistics and of artificial intelligence. A famous example in this respect is the so-called "Blocks world" system developed in 1972 by Terry Winograd (described e.g. in [116]). In these approaches not only the model space is defined by a human, but also the particular model is selected manually, based on the expert's intuitions of the data and the problem.

Although the work in this thesis is mainly motivated by applicability of the developed methods to practical problems regarding text mining, it may be interesting and instructive to consider, for a moment, language acquisition also from the cognitive modeling viewpoint. It has been suggested by Noam Chomsky, and to some degree later advocated e.g. by Pinker [94], that humans generate language based on a "universal grammar", an innate mental model for producing an unlimited number of sentences based on a limited set of rules. Moreover, an argument defending a nativist and atomistic view of concepts[7] has been recently described by Fodor [20]. From the point of view of learning systems research, strict claims for one or the other extreme do not seem to be supported by study of human language learning on one hand and on research in adaptive systems on the other.

In all learning systems and their practical implementations the structure of the model space, the search algorithm, and the available data together determine the outcome. In practice, the model space considered is never the space of all imaginable models[8]. The model space available within each human is determined by evolution, defined by the exact genetic makeup, and implemented, among other things, by the limits on neural connectivity and plasticity in the developing brain. The kinds of data available within the environment of the child direct the search for models[9]. It remains the domain of empirical research in computational neuroscience and in child behavior and learning to narrow down the specific mechanisms and the extent to which language is learned in humans. However, adaptive systems research and especially the theories that are developed may offer some useful conceptual tools for addressing the problem.

[7] A concept can be described by analogy as a *term in the vocabulary of thought.*

[8] Finding the best model from such a space given a finite amount of data and finite time simply could not be performed, regardless of whether it is possible in principle.

[9] It should be noted that an infant, by being able to also generate language, at first as mere sounds, and gradually also communication more extensively understood by others, is able to actively test the developing models, and also to obtain ample feedback on the success of such models. This testing, in turn, is directed by the needs and motives of the child, including the desperate need to communicate that Pinker views as underlying "the language instinct"[94].

3.2.1 Levels of natural language analysis

The following table illustrates some possible levels of representation of language data or the kinds of tasks that presumably become possible with features from that level:

Feature level		Task or target of representation
visual/auditory signals	$\Longrightarrow$	speaker identification
phonemes, characters, and n-grams of them	$\Longrightarrow$	language identification
morphemes, words, word n-grams	$\Longrightarrow$	concepts, topics
phrases, sentences	$\Longrightarrow$	propositions, events
strings of phrases, document structure	$\Longrightarrow$	stories, arguments
several documents	$\Longrightarrow$	thematic domains, text types

For example, character trigrams can be used to quite reliably identify the language of a piece of text, whereas word-level estimates may perform poorer [27]. On the other hand, if semantic content is to be represented, it appears appropriate to start from the level of words (or possibly morphemes).

3.2.2 Natural language phenomena

Several phenomena particular to natural language have an impact on the modeling task that text representation systems are faced with. These phenomena may affect the amount of noise to be expected, as well as the dimensionality and the complexity of the learning task. Understanding of the phenomena may aid in considering appropriate features and data encoding as well as in choosing the appropriate model family to be searched by the learning algorithm.

Separation of symbol and meaning. In natural language texts symbols are used as *signs* that refer to meanings. In general, the relationship between a sign and its meaning cannot be determined by the apparent properties of the sign—the relationships must be learned. Consequently, words with similar meaning often do not resemble each other in appearance, and the semantic similarities between words must therefore be learned from their contexts of use.

Discrete symbols vs. continuous representations. Although words themselves can be viewed as discrete symbols, many sets of words have a natural and compact representation in some continuous space that encodes directly the similarities between the concepts: consider, for example color words that denote properties that are very naturally expressed in a 3-dimensional continuous space. This discrepancy is likely to have arisen because discretization is useful in communication for protecting the content from corruption by noise. Unfortunately the discretization also results in losing information about the relatedness of the words. Re-inventing such information from observing only the communicated discretized signal, i.e. words and their contexts, remains a challenging task.

Variation in form. Many different expressions can be used to describe or induce approximately the same idea in the mind of the listener. Examples of such variation are *synonymy*[10], *inflections* or word forms[11], shortenings, acronyms, nicknames and varying writing styles[12], and plain spelling errors. An overview of textual variation found in real world documents is given, e.g., in [80], ch. 4. The problems caused by the variation in form and the fluidity of meaning are often referred to as the *vocabulary problem* [10].

Ambiguity or fluidity of meaning. When used by different people, at different times, or in different contexts the same word can flexibly refer to different meanings called *senses* of the word. Consider, for example, *bank* as a financial institution, or referring to river bank. The phenomenon is commonly called *polysemy* and dictionaries list numerous examples of different meanings of a single word. However, more often the senses are not totally distinct (e.g. the senses of *open* in *opening a door* and in *opening an exhibition*), but rather form a *continuum* in the space of meanings (for general remarks regarding polysemy from the computational point of view see [8, 3]; a detailed treatment of the polysemy of the word "get" can be found in [96]). The fluidity and contextuality of meaning has been modeled e.g. using the self-organizing map in [45]. Polysemy is not a marginal phenomenon of the language, but rather ingrained into it—since a single word may be used flexibly for various purposes depending on the textual or the non-textual context, fewer words are needed for communicating a larger amount of possible messages. Some forms of communication even seem to purposefully utilize the aspect of ambiguity and multiplicity of interpretations, for example, poems and humor that plays on the sudden shifts of interpretation.

Uncertainty of meaning can be caused also by *homography*, where two unrelated words may have some overlapping morphological forms, e.g. *saw* can be either the past these of the verb *see* or the name of a tool. In communication between humans the ambiguities are probably resolved based on structural and semantic cues in the context of the utterance and based on prior knowledge of the world.

Seriality and structure. In speech and in text words appear in serial order, constrained by some implicit structure. The ordering of words also carries meaning: in some languages even the same set of words can be arranged differently to convey very different meanings (e.g. *dog bit man* or *man bit dog*; the first is not surprising but the second would be an item for news). Moreover, knowledge of syntactic constraints and relationships may help in *word sense disambiguation*, i.e. in identifying the particular sense of a given lexical item[13]. Also more complex structural relationships both on the *syntactic* or sentence level and on the *discourse* or story level have an effect on the meaning that is construed by an utterance or a longer text.

A variety of relationship types. Words, concepts, documents and topics can be related to each other in many different ways. The various types of relationships between words

[10] Different words that can have approximately the same meaning, at least in some context

[11] In Finnish a single verb root may have over 18,000 different inflected forms and a noun some 2,000 forms [79].

[12] For example, the SOM has in the literature been referred with numerous ways, including self-organizing map, self-organising map, self organising map, self organised map, Kohonen map, Kohonen network, Kohonen net, SOM or SOFM.

[13] The general idea holds regardless of whether the senses are considered to exist in a discrete or in a continuous space: in a continuous space the disambiguation would correspond to narrowing and heightening of the probability density distribution in some area of the sense space, while in other areas the density would diminish correspondingly.

have been studied in the WordNet project. Human labor has been used to construct a consistent, machine-readable lexical database of English [91] where 95,000 words and their relationships are presented in a psycho-linguistically motivated manner. Relationships between *word forms* include synonymy and antonymy[14] and *between concepts* there are part-whole relationship (meronymy/holonymy) and subset/superset relationship[15]

Languages evolve. As societies change, old words are forgotten or adopted to new purposes and new ones are invented. Specialized fields may evolve their own terminology and particular meanings for commonly used terms, or a special term may be adopted for a somewhat different use by the general public. A model based on the self-organizing map of the development of language through interaction in a community has been considered in [35].

Computational challenges. The richness of textual data, the complexity of problems related to language, and the sparsity of data together pose serious challenges for modeling. Even in collecting data on the level of individual words, problems arise: a truly representative sample has never been seen of all words, and due to unrepresentative samples, over-fitting and incorrect generalizations are possible [8].

In some cases, human labor can be utilized for encoding some of the relevant information such as grammar rules, inflections of words, etc. However, it is generally agreed that the approach is not sufficient as a general solution, for the lack of resources. The approach suffers also from the well-known problem in artificial intelligence: how to extract knowledge from experts of the field and convert it into any machine-usable form. Differences between experts are typical, and even the same person can at different times give a different answer to the same question.

3.3 Document representation models

Currently, in most research in mining of text document collections the documents are viewed as containers for words[16]. This approach, often called the *bag of words* encoding, ignores the order of the words as well as any punctuation or structural information, but retains the number of times each word appears[17].

Based on the discussion of natural language in Section 3.2.2 it is obvious that the bag-of-words encoding is a gross simplification of the wealth of information communicated by a document, merely a fingerprint rather than a faithful description of the content. Developing richer models that are nevertheless computationally feasible and possible to estimate from actual data remains a challenging problem. Facing the challenge will eventually be necessary if harder tasks related e.g. to language understanding and generation are to be tackled seriously.

Although not intricate enough for generating language, the bag-of-words-encoding nevertheless provides a considerable amount of information about associations between words and documents, which is sufficient e.g. for thematic or topical clustering and for information retrieval from large collections.

---

[14] Antonymy—(partial) opposition of meaning, e.g. *rich* and *poor* are antonyms.

[15] Hyponymy/hypernymy, also called ISA-relationship. For example, *tree* is a hyponym of *plant*.

[16] Also *collocations*, pairs of words that occur together more often enough for them to seem connected, are sometimes utilized as features.

[17] A simpler version of the bag-of-words encoding only retains binary information of word appearance.

The following discussion of document representation methods concentrates on models that can be estimated automatically and efficiently based on very large quantities of data at high speeds. They are designed to provide computationally feasible engineering solutions for tasks in which utilizing human labor would be expensive, too slow, or even impossible due to large amounts of data. Typical tasks that the methods are used for in text mining are: accessing a piece of text based on partial or noisy information (e.g. in IR), ordering items based on similarity, summarizing the content of documents or collections, contrasting items and sub-collections, and extracting properties of individual textual items or collections of them.

3.3.1 Vector space model

A straightforward numeric representation for the bag of words -model is to represent documents as points (or vectors) in a t-dimensional Euclidean space where each dimension corresponds to a word (term) of the vocabulary [105, 102]. The i:th component d_i of the document vector expresses the number of times the word with index i occurs in the document, or a function of it. Furthermore, each word may have an associated weight to describe its significance. The similarity between two documents is defined either as the distance between the points or as the angle between the vectors (to disregard document length).

Despite its simplicity, the vector space model (VSM) and its variants are currently the most common way to represent text documents in mining text document collections. One explanation for this is that vector operations can be performed very fast, and efficient standard algorithms exist for performing model selection, dimension reduction and visualization in vector spaces. In part for these reasons the vector space model and its variants have persisted in evaluations of quality e.g. in the field of information retrieval [3].

An obvious problem with the vector space model is the high dimensionality: the number of different words (word types) in a document collection easily rises to hundreds of thousands. The problem is compounded by varying writing styles, spelling errors, etc. Furthermore, in VSM any two words are by definition considered unrelated. However, it is hard to obtain accurate information of semantic relatedness automatically from textual information only.

If one could base the model on some kind of latent variables or conceptual dimensions instead of words, a considerably more concise representation might ensue. In fact, it has been suggested that on the cognitive level of representation the meanings of words are points in some *low-dimensional concept spaces*, which consist of a number of *quality dimensions* each with certain topological or metric properties [23, 24]. Some of the quality dimensions may be grounded rather directly in our perceptual system (e.g. color words) while others may be more abstract. Miikkulainen describes a mental model of the lexicon that utilizes interconnected SOMs for orthographic, phonological and semantic representation levels[89, 90]. When various kinds of damage to the network is simulated, the lexical model is shown to exhibit similar category-specific aphasic impairments as observed in human patients. A model of building a connection between the sensory and the word level in an anticipatory system utilizing SOMs is proposed in [36]. In Gallant's Context vector method [21, 22] a set of *feature words* are used as the grounding or feature dimensions, and other words are encoded in terms of their distances to the feature dimensions. However, the method requires manual insertion of a considerable amount of distance information, and is sensitive to the entered distances and the selected feature words.

Several attempts have been made to obtain a suitable lower-dimensional representation for text in a data-directed manner often starting with the standard vector space model. Some of these methods are briefly discussed next.

3.3.2 Latent Semantic Indexing

Relationships between words can be deduced from their occurrence patterns across documents. This notion is utilized in a method called Latent Semantic Indexing (LSI) [16] which applies singular-value decomposition (SVD) to the document-by-word matrix to obtain a projection of both documents and words into a space referred to as the latent space. Dimensionality reduction is achieved by retaining only the latent variables (projection dimensions) with the largest variance (largest eigenvalues). Subsequent distance calculations between documents or terms are then performed in the reduced-dimensional latent space.

The original LSI algorithms had a high computational complexity, $O(N^3)$, which was problematic for use with large data sets. The computational complexity of the LSI is known to be $\mathcal{O}(Nld)$, where N is the number of documents, l is the average number of different words in each document, and d is the resulting dimensionality. It has recently been suggested that the Random Projection [17] or similar methods [93] could be used for reducing the computational complexity of the LSI as well.

3.3.3 Random Projection

For many applications and methods, the central aspect in document representation is the distance between documents. It has turned out that an initially high-dimensional but sparse data space can be projected onto a randomly selected, much lower-dimensional space so that the original distances are nearly preserved [53]. In effect, the exactly orthogonal basis vectors of the original space are replaced by vectors that are with high probability nearly orthogonal, even with randomly chosen directions if the final dimensionality is sufficiently high. An intuitive reason for this perhaps surprising finding is that in very high-dimensional spaces the number of nearly orthogonal vectors is much larger than the dimensionality of the space.

The advantage of random projection is that it is extremely fast: in an efficient implementation of random projection by pointers, introduced in Publication 7 the computational complexity is only $\mathcal{O}(Nl) + \mathcal{O}(n)$, where N is the number of documents, l is the average number of different words in each document, and n is the original dimensionality of the input space.

Furthermore, it can be applied to any high-dimensional vector representation, and any algorithm that relies merely on vector distances, can in principle be applied *after* the random projection and therefore in a much lower dimensional space. Random projection has been used, e.g., for representing words before averaging in [97], for document encoding in text exploration prior to the application of SOM (see e.g. [52], Publications 4 and 7), as a preprocessing for LSI in document representation [93], and as a preprocessing for SOM in retrieval of spoken documents [71].

3.3.4 Independent Component Analysis

Recently, there have been attempts to apply a method called Independent Component Analysis (ICA) [4] to representing text documents. In the ICA model the data is assumed to be generated as some, typically linear, mixture of a set of independent random variables, also called *sources* (see e.g. [47, 48]): $x = As$, where x is the known data, A is the unknown mixing matrix and s the unknown sources. Various ICA algorithms attempt to estimate the sources as well as the mixing matrix by maximizing a measure of independence of the sources. While PCA (or SVD) look for merely *uncorrelated* latent variables with maximal variance, ICA searches for *independent variables*.

In [49] ICA was utilized for information retrieval; a document was assumed to be generated by a set of independent topics. In the model a single document was represented as a linear combination of the active topics, several of which could be active for a single document. The topics are assumed to differ in their probability density distributions for words.

3.3.5 Word clusters

Clustering methods can be used e.g. for reducing the number of data by grouping together similar items [50]. In document representation clustering methods can be applied to group together similar words, and then represent documents in terms of word clusters rather than individual words. A clustering that is well suited for document representation should reduce the variation of form (cf. Sec. 3.2.2) while losing as little information as possible regarding the semantic content, especially the topics discussed in a document.

An overview of various methods for collecting information of words with the purpose of categorizing or clustering them automatically is presented in [8]. In languages with strict restrictions on word order, such as English, the distribution of words in the immediate context of a word contains considerable amounts of information regarding the syntactic category of the word, and, mostly within the syntactic categories, information about the semantic category as well [19, 44, 120].

Word category map (WCM) English words have been clustered in an unsupervised manner based on the distributions of words in their immediate contexts using the self-organizing map (SOM) by [97, 19, 44, 37] and the subsequent categories have been used for representing documents (see e.g. [41]). Applications of word category maps are considered in [38] and in [37].

3.3.6 Term weighting

In considering the meaning of a piece of text, it seems that some words carry more meaning than others. In addition to a basic division to function words (e.g. *which*, *of*, and *and*) and content words (e.g. *table*, *sun-burned*, and *to shout*) some content words seem to target the theme of discussion much more precisely than others. Consider, for example, the words *astronomer* and *book*.

Regardless of which method is used for dimension reduction or for deducing latent dimensions, it is possible to assign weights to the words which attempt to describe how important the word is for the document representation. The weights may be based on a word distribution model, e.g., the Poisson distribution [13], or an estimate of informativeness such as entropy across topics (Publications 4 and 7 and [115]).

A commonly used weighting scheme is the $tf \times idf$ -family [103], where tf stands for term frequency within the document, and *idf* stands for the inverse of the number of documents in which the term appears. The scheme is based on the notion that words that occur frequently in documents are often less significant for meaning, and rare words probably carry more meaning. Many variations of the general scheme exist. For example, the weight W_{ij} of a word w_i occurring in document d_j can be calculated as follows: $W_{i,j} =$

$(1 + \log tf_{i,j}) \log \frac{N}{df_i}$ where $tf_{i,j}$ is the frequency of the term i in document j, and df_i is the document frequency, i.e. the number of documents in which the term i appears. The weighting assigns maximum weight to words that appear only in a single document. Since in the vector space model term weights directly affect the distances between documents, the results greatly depend on the weighting of terms.

The above global weighting schemes attempt to describe the importance of a word irrespective of its particular context, such as nearby words, or the location of the word in the document structure. Prior information about the structure of documents may be utilized as well, for example to emphasize title words or words that appear in the beginning of the document.

3.3.7 Probabilistic modeling

Probabilistic modeling allows answering questions in terms of probabilities, e.g. "how probable is this document in this model" or, if the model has been constructed for document classification, "what is the most probable class for this document". Furthermore, if appropriate prior information about the task or the data exists it can be encoded explicitly using a rigorous mathematical framework. Two models that employ the assumption of independence between words in a document are briefly described below (for a detailed description of their usage with text see [81] or in general [25]).

The binary encoding of documents that disregards word occurrence counts in a document is captured by the *Multivariate Bernoulli Independence model*[18]. In other words, a document is assumed to be generated by a collection of discrete, independent random variables, one for each word in the vocabulary. If a certain word appears in the document, the value of the respective random variable is 1, otherwise 0, thus disregarding the frequency of occurrence.

The bag-of-words representation is captured by the *Multinomial model*, in which each word in a document is assumed to be drawn from a multinomial distribution of words with as many independent trials as the length of the document counted in words [81].

4 WEBSOM DOCUMENT MAPS

In the WEBSOM method, the self-organizing map algorithm (see Sec. 4.1) is used for projecting documents from an initially very high-dimensional space onto a two-dimensional map grid, so that nearby locations on the map contain similar documents. Subsequently the map can be used for visually conveying information about the document collection, for exploring the collection, and for performing searches on the documents.

4.1 Self-organizing map (SOM) algorithm

The self-organizing map (SOM) [61, 62, 66] is an unsupervised neural network [30] algorithm that is able to arrange complex and high-dimensional data so that alike inputs are in general found close to each other on the map. The organized map avails itself readily to visualization, and thus the properties of the data set can be illustrated in a meaningful manner.

[18] also called the *Binary Independence model*.

The SOM algorithm places a set of reference vectors—also called model vectors—into the input data space so that the data set is approximated by the model vectors. The model vectors are constrained to a (usually two-dimensional) regular grid that in effect forms an "elastic network" which, by virtue of the learning algorithm, follows the distribution of the data in a nonlinear fashion. The SOM algorithm obtains simultaneously *a clustering* of the data onto the model vectors and a *nonlinear projection* of the data from the high-dimensional input space onto the two-dimensional, ordered lattice formed by the model vectors.

In the original, stochastic version of the SOM the data samples are presented to the algorithm in random order, possibly several times. At each step the best-matching model (*winner*, also called *best-matching unit* or BMU) for the current data sample is searched. Subsequently, the winner model and its neighbors on the lattice are updated. Given a data sample $x(t)$ at iteration step t the model vector $\mathbf{m}_i(t)$ with index i is adapted as follows:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{c(\mathbf{x}),i}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \ , \tag{3}$$

where the index of the "winner" model for the current data, $c(\mathbf{x})$, is

$$c(\mathbf{x}) = \arg\min_i \{\|\mathbf{x} - \mathbf{m}_i\|\} \ . \tag{4}$$

$h_{c(\mathbf{x}),i}(t)$ is called the *neighborhood function*, which acts like a smoothing kernel over the grid, centered at the "winner" model $\mathbf{m}_{c(\mathbf{x})}(t)$ of the current data sample. The neighborhood function is often taken as the Gaussian

$$h_{c(\mathbf{x}),i}(t) = \alpha(t) \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_{c(\mathbf{x})}\|^2}{2\sigma^2(t)}\right) \ , \tag{5}$$

where $0 < \alpha(t) < 1$ is the learning-rate factor which decreases monotonically with the iterations. A finite-width approximation of the Gaussian can be used for reducing the number of calculations. The width $\mathbf{r}$ of the neighborhood function is decreased monotonically during the learning process. In effect, initially a large number of models are updated for each data sample, and later only few models are slightly adjusted. In the final stage the distribution of the SOM reference vectors in the input space roughly approximates the density of the input data [62]. Note that the density approximation and the ordering of the data are competing goals between which the algorithm makes a compromise that depends e.g. on the final width of the neighborhood function.

The model vectors perform an implicit clustering of the data where each data point belongs to the cluster of the closest map vector. The clustering is said to be implicit because the number of clusters need not be the same as the number of map units—several neighboring units may form a cluster. In a stable state, each model vector expresses a weighted average of the data points in that map region, particularly of data points mapped to the unit associated with the reference vector. The neighborhood function defines the size and the shape of the weighting function. If the neighborhood width is zero the algorithm is equal to the so-called K-means clustering algorithm.

A faster convergence is often obtained by the batch computation of the map (Batch Map), similar to the batch version of the K-means algorithm (e.g. [50]). In Batch Map the changes to the models are collected over the whole data set before the models are updated. For large maps a fast parallelized implementation can be utilized (Publication 7).

4.1.1 Applications of the SOM

The SOM is one of the most widely used neural network algorithms. Studies in which the SOM has been used or analyzed have been reported in over 4000 scientific articles (for an older collection, see [56]). Most of the early applications were in the fields of engineering but nowadays a very diverse range of applications is covered, from medicine and biology to economics and natural language analysis. Overviews of the applications are given in [62, 69]. A collection of recent works has been published in [118].

The usefulness of the SOM stems from its two properties: (1) It creates models or *abstractions* of different types of data in the data set, and (2) it *organizes* the abstractions onto a usually two-dimensional lattice which can be used to generate an illustrative graphical display of the data set. The latter property makes the SOM especially suitable for data mining and exploratory data analysis (for a detailed treatment, see e.g. [52]). Some examples of such applications of the SOM include construction of overviews of socio-economic data sets [57] and financial analyses [15].

4.1.2 Accelerated computation of the SOM

Each iteration step of the original SOM algorithm consists of winner search and updating a neighborhood of the winner. The complexity of the winner search is $\mathcal{O}(dN)$ where d is the dimension of the vectors, and N the number of map units within the neighborhood. The updating step is always of at most the same complexity. The number of iterations should be a multiple of N to ensure sufficient statistical accuracy. To sum up, an upper limit of the complexity of the traditional iterative SOM algorithm is given by $\mathcal{O}(dN^2)$.

When very large maps are created of high-dimensional data such as documents, the requirements on main memory space, disk space and CPU time in the SOM teaching are considerable, and the standard algorithm is not feasible. However, by introduction of various computational tricks and a careful implementation, the requirements of both space and processing time can be considerably reduced. Initial speedups were described in [63]; for a detailed description of the latest methods and of experimental results, see Publication 7. In brief, the major speedups consist of the following:

- Rough initial ordering of the map. Faster convergence is achieved if an approximate initial ordering of the models is obtained e.g. by applying PCA [66].

- Estimation of larger maps based on carefully constructed smaller ones.

- Parallelized Batch Map. A parallelized implementation of the Batch Map can be used by dividing data among different processors in a shared-memory computer. As the map is changed only between epochs, the map can be shared read-only during the BMU search.

- Local search of best-matching units. Instead of full search of the BMUs for each data point at each epoch, the previous winner can be used as a starting-point and a local search performed in the neighborhood.

By using the speedup methods for the winner search and for the estimation of large maps based on smaller ones the complexity can be reduced considerably, to $\mathcal{O}(dM^2) + \mathcal{O}(dN) + \mathcal{O}(N^2)$, where M is the number of units in the smaller map.

4.2 Creation of large document maps

An overview of the creation and use of large document maps in the current WEBSOM system is depicted in Fig. 1. In the following section the various processing stages are briefly described. For a complete description, see Publication 7.

4.2.1 Document encoding

Preprocessing. The document headers, signatures, and all non-textual information is removed. Mathematical symbols, numbers, URLs, and email addresses are replaced with one of the special dummy tokens. Common, uninformative words that are listed as stop words are discarded. Also the rarest words are removed (with large collections a cutoff frequency of 50 has been used). In our later experiments with both English and Finnish the morphological variation has been reduced by replacing words with their base forms using the TWOL analysis package [70] of Lingsoft Inc.

Weighting. For weighting of words two methods have been applied, a $tf \times idf$ weighting (see Section 3.3.6) and an entropy-based weighting (see e.g. Publication 4). The latter can be utilized whenever some topical categorization of the material is available.

Dimensionality reduction. Although the preprocessing stage reduces the initial vocabulary, with large collections the remaining number of words is still very high, say in tens or hundreds of thousands of words. In experiments described e.g. in [41] and Publication 2 dimension reduction was achieved by categorizing words based on their contexts using word category maps (see Sec. 3.3.5), and encoding documents as word histograms. In Publications 4 and 7 random projection (Sec. 3.3.3) was used to reduce the dimensionality of the document vectors.

4.2.2 Fast creation of an organized document map

Large SOMs of very large document collections can be created rather fast with the methods described in Sec. 4.1.2. With textual data, further speedup is achieved due to the sparsity of the document vectors. If the vectors are normalized to unity and distances are measured using inner products, distance computations can be performed very efficiently by skipping components with zero values.

It is estimated that taking into account the speedup obtained by random projection of document vectors, the total speedup factor in construction of very large document maps is of the order of the dimensionality of the original input vectors (for details, see Publication 7).

4.2.3 Adding new documents

New documents can be inserted onto an existing map simply by locating the best-matching map unit for each document. However, in a non-stationary document collection where new topic areas and terms are introduced, after a while the map may not be such a good representation of the collection. An intuitive reason for this is that in a very high-dimensional and very sparse space an unseen document will not be near any area of the two-dimensional map unless its own topical domain is discussed by documents that contributed

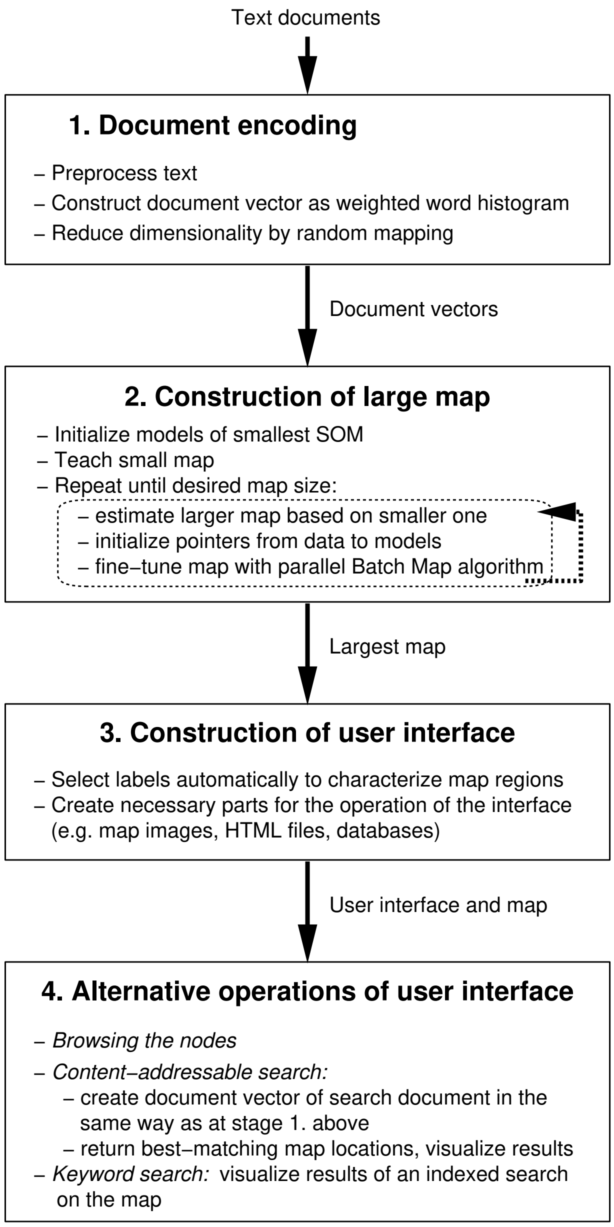


Figure 1: Overview of the construction and operation of the WEBSOM system as described in Publication 7.

to the map construction. In a non-stationary collection the map should therefore either be incrementally adapted or fully re-calculated after a time[19].

4.2.4 Evaluation of document maps

Good evaluation methods for measuring the quality of visualization, exploration and navigation are very difficult to define. Ultimately user studies are required, but also more direct, automatically applicable measures are clearly needed—without such measures, improvement of methods is extremely difficult and the research is bound to be constructive rather than analytic.

For evaluating document maps we have utilized an indirect measure that is based on an external topical classification of documents, best described as the *purity* of map nodes, defined as the proportion of documents that fall into a map unit where their own class forms a majority. The measure can be utilized when comparing methods using an single document collection (see e.g. Publication 7). However, the absolute values obtained are collection dependent, since the number of topic categories affects the results. Moreover, the measure evaluates only the *local coherence* of the individual map units, not the overall organization of the map.

The overall organization has been studied by observing visually the displays of the distributions of various topics on the document map (see Fig. 2). The measure is subjective, but nevertheless useful in practical situations. Furthermore, the visualized class distributions offer more information regarding the organization of a map, e.g. in the presence of prior information concerning dependencies between topics, than could be achieved with a straightforward automatic measure.

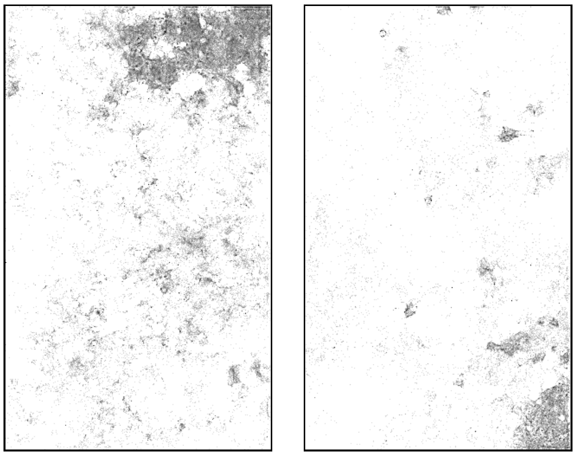
A fundamental problem with evaluating maps according to an existing categorization is that the categorization itself may not be accurate, for example, the categories may be overlapping, the borders fuzzy, or the same article should perhaps belong to several categories. It is even possible that the automatic methods could produce better categorizations than the original one, but a relative measure can never identify such a situation.

An additional, subjective means of evaluation is obtained by providing an interface by which the organization obtained by the maps can be explored. Although no quantitative comparisons are achieved this way, the experiences obtained from browsing the maps have throughout the project guided research intuitions and suggested hypotheses for further rigorous study. The fact that the models are visual and explorable thus not only aids in the eventual user tasks but also in researching the models as well.

4.3 User interface for document maps

The practical purpose of developing the WEBSOM document map interface was to set up public demonstrations of document maps that anyone could conveniently explore. The visualization was to offer a view of the collection that would help in forming a general understanding of the domain, as well as to guide exploration towards potentially interesting particular areas. In addition, a convenient and efficient strategy of moving from general view to specific details and back had to be designed, including methods of interaction and a strategy for providing the user with a sense of location and of context. Suitable visual and textual means for conveying information about the content of the map had to be developed as well.

[19]The matter is similar to the so-called *folding-in* of new documents in LSI.



Chemistry Engines or pumps

Figure 2: Distribution of two sample subsections of the IPO patent classification system on the document map of 7 million patent abstracts. The gray level indicates the logarithm of the number of patents in each node.

For the implementation of an exploration interface to be used in a public demonstration the following goals were identified: (1) to make use over long distances and slow Internet connections sufficiently comfortable, (2) to enable use regardless of user's software or hardware, and (3) to conserve CPU time, disk space, and main memory at the WWW server side. To fulfill these goals various optimizations were necessary, including minimizing the sizes of transmitted material by reducing the number of colors in images. In addition, user interaction strategies that were considered either too expensive for the WWW server of a popular demonstration, or too costly for the user in terms of transmission time, were discarded[20].

4.3.1 Navigation interface

While navigating in an information space the user is constantly faced with the following problems: *Where do I want to go next and how do I get there?*, *Where am I?* and *Where have I been already?* A good navigation tool aids the user in making informed decisions about future actions and in bookkeeping the information needed for making the decisions.

The WEBSOM navigation interface (see Fig. 3) consists of an image of the whole document map, a hierarchy of zoomed pieces of the map at various zooming levels, and a set of HTML pages, imagemap files, and CGI scripts. *Zooming in* is achieved by pointing and clicking with the mouse at the desired location on the map image. *Horizontal movement* to nearby areas as well as *panning out* is carried out by clicking on a compass image. On the more detailed zoom levels white dots mark units of the regular, hexagonal map grid (see Fig. 5). By clicking near a dot the list of documents associated with the map unit is accessed. From the list, individual documents can be selected for reading by clicking on the title.

[20]It is likely that as the WWW environment develops, at least some operations previously considered too costly or awkward to implement could be implemented efficiently and elegantly.

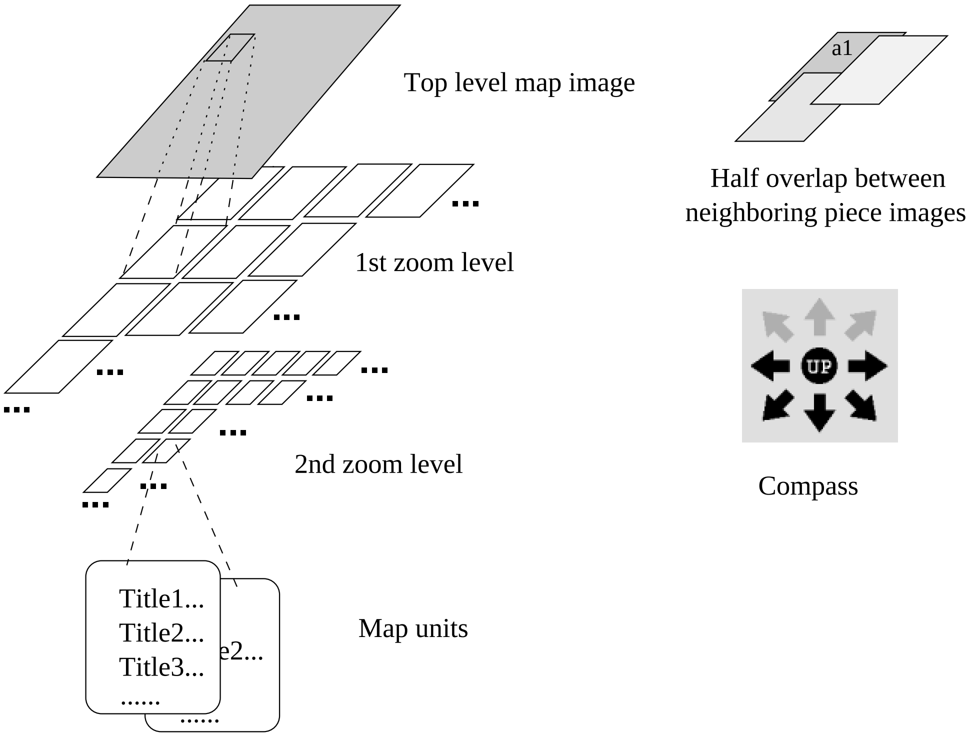


Figure 3: The images of the navigation interface form a hierarchy of zoom levels with increasing magnifications with respect to the original image. Clicking a map image performs a "zoom in" operation. For horizontal movement a "compass" is provided: clicking on a black arrow causes movement to the neighboring area with a half overlap between neighboring images (the arrows leading out of the edge of the map turn grey). A click on the center of the compass performs a "pan out" operation.

The initial exploration interface was designed and applied for small collections of 5,000–20,000 documents. However, in scaling to larger collections it soon became apparent that in addition to the computational aspect of scalability also the *cognitive support of navigation* in large spaces had to be considered. For example, with large information spaces the zoom operation is necessarily more than just an image transformation: additional details such as more specific labels and a detailed view of the landscape should appear at lower zoom levels. However, a balance is required between increasing the amount of detail and maintaining a sense of continuity across zooms. In horizontal movement the sense of continuity is endorsed by an overlap of half an image across movements to each direction (see Fig. 3). Another balancing is necessary between adding more zoom levels for the sake of continuity and keeping the number of levels at a minimum in order to provide efficiency and convenience of exploration.

The following properties of a system offering spatial navigation were identified as useful:

- **Continuity.** Providing continuity across movements helps the user in maintaining an internal sense of location and in keeping track of history.

- **Context.** Offering explicit information of the user's location with reference to the overall information space and to nearby areas at all times reduces the cognitive load of the user.

- **Navigation history.** Keeping track of navigation history for the user and visualizing it as needed may be very helpful. For example, in browsing the WWW the visited links appear in a different color than unvisited ones.

Implementation details. A maximum zoom ratio of 1:4 between consecutive levels has been used on the largest map, but generally a smaller zoom is found more comfortable. Furthermore, on the lowest zoom level the dots marking map units should be separated by a sufficient distance so that selection of individual units is possible—we have used distances between 13–50 pixels. Given these constraints, the largest map of 7 million documents and one million units required two zoom levels between the overall view and the level of map units. The map images were created in advance and stored as static files to minimize calculations while using the interface. In the current implementation both the articles and the contents of the map units are stored in databases, accessed at request by CGI scripts that also generate the HTML layout. The clickable map images are operated using server-side and client-side imagemaps.

Additional ideas for improving navigation. Currently the context of browsing can be viewed by panning out, then zooming back in. However, visualizing current location and context in a straightforward manner could be implemented e.g. by showing at each page an additional miniature image of the whole map and the current location on it. This resembles the use of the fish-eye view, where a detailed image of the focus of interest is shown, and an overall image visualizes the context of the focus point at a less detailed scale. Recent navigation history could be visualized on map displays as sequential paths marking locations already visited. Such facility could be implemented using HTTP cookies and dynamic visual overlays on the static map images.

4.3.2 Visualized document map

The WEBSOM document maps have been visualized using two methods, a smoothed version of the U-matrix [113], and a smoothed document density diagram. In the U-matrix visualization dark color corresponds to a considerable difference between the model vectors of neighboring map units, whereas a bright color signals similarity between neighbors. In contrast, in the density diagram light color denotes a large number of similar documents and dark color an emptier area. Due to the relationship in SOM between density of model vectors and density of documents, both methods visualize the cluster structure to some degree, although U-matrix more faithfully. The density visualization was finally chosen because (1) it was faster to compute and did not require the original document vectors, (2) the visualizations obtained are sufficiently similar, and (3) explaining the significance of the colors to users of the demonstrations is considerably easier in terms of numbers of documents than of the more abstract similarities between map areas.

Smoothing is applied in either visualization to achieve a pleasant, varying display with little detail, resembling a landscape with hills and valleys. The landscape can then be used as a background for presenting various details and dynamic information.

As pointed out e.g. by [111] the use of multi-functioning graphical elements can make visualizations more compact and informative. The smoothed document map landscape carries out the following functions: (1) it *describes the document density* at each area and

(2) it *provides texture* that can help maintaining a sense of location and context across movements and across dynamic visualizations.

4.3.3 Labeling the map display

Interpretation of a document map display can be aided by labeling the display with a selection of descriptive words that characterize regions of the map. The labels can be utilized for multiple functions: (1) to *describe the underlying area*, contrasting it against the rest of the map, (2) collectively to *summarize aspects of the collection*, (3) and in navigation to *act as landmarks* or anchor points that help orientation by providing reference points during transitions across views that have different resolutions.

An automatic method has been introduced in Publication 5 for selecting descriptive terms suitable for characterizing textual clusters, individual map units, and document map regions. The method was validated against human-assigned descriptors on a map of 10,000 INSPEC articles. The method has been applied for labeling several WEBSOM demonstration maps since 1997[21]. A speeded-up approximation was used for labeling the map of 7 million patent abstracts described in Publication 7.

Characterizing textual clusters and map units. A good descriptor of a cluster can be said to characterize some outstanding property of the cluster in relation to the rest of the data set. This can be formalized in the following way: in a measure that compares the word's frequency to other word frequencies in the cluster, and also to its own relative frequency generally in the collection. The measure of goodness $G(w, j)$ for a word w to characterize cluster j is defined as

$$G(w,j) = F_j(w) \frac{F_j(w)}{\sum_i F_i(w)}, \tag{6}$$

where $F_j(w)$ is the proportion of the word w in cluster j.

Sometimes cluster borders are not known exactly. In such cases, a *neutral zone* can be left between the cluster and the rest of the collection, which does not affect the measure for word w (see figure 4). Furthermore, instead of individual map units j, the measure can be defined for groups of neighboring units. The measure for the goodness of word w for labeling the map area A_0^j centered at map unit j thus becomes

$$G^2(w,j) = [\sum_{k \in A_0^j} F_k(w)] \frac{\sum_{k \in A_0^j} F_k(w)}{\sum_{i \notin A_1^j} F_i(w)}, \tag{7}$$

$$\text{where } k \in A_0^j \text{ if } d(j,k) < r_0, \text{ and}$$
$$i \in A_1^j \text{ if } r_0 < d(j,i) < r_1,$$

where r_0 is the inner radius of the area and r_1 the outer radius, and the "neutral zone" in between is disregarded.

Labels for map areas. With large maps it is neither possible nor desirable to label every map unit on the display. Often there is not enough space on the graphical map display,

[21] Demonstrations are available at `http://websom.hut.fi/websom/`.

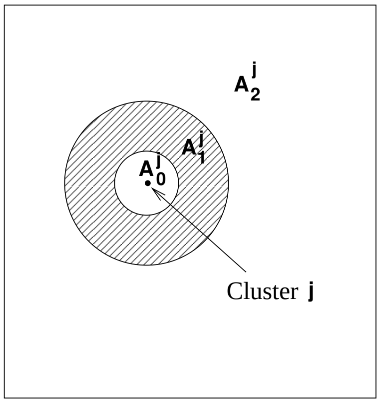


Figure 4: Determining the goodness value G^2 for words in map unit j. The shaded area (A_1^j) is disregarded when calculating the goodness values for each word in the area centered around unit j.

and even if there were, cramming the display with masses of words should not be called visualization. Therefore, to obtain the final labeling of a map, we need to place labels on a subset of the map units so that the resulting labeling is as good as possible. If the total goodness of the labeling is defined as sum of the goodnesses of all labels, there are $\binom{N}{M}$ possible combinations of N labels out of M candidates. Obviously all combinations cannot be evaluated in most practical situations. Fortunately, in this type of selection problems a greedy approach usually produces a near-optimal approximation.

With WEBSOM document maps that have several zooming levels, we have used the following procedure for selecting labels for each level l, starting with the topmost level. First, decide for each display level l the desired labeling density expressed in terms of minimum distance on map grid between two labeled units i and j, $d_l(i,j)$. Then, perform the following steps for each level:

1. Order map units according to goodness of the best word in the unit.
2. Repeat: accept the best word from the best unit on the map if it is separated at least by distance d_l from all already accepted labels.
3. When no more labels can be added for level l, increment l.

We have obtained good results by choosing the radius parameters r_0 and r_1 used in calculating G^2 so that half of the desired labeling density, $d/2$, lies between r_0 and r_1.

It should be noted that the method assumes that labels on the display are independent. Naturally, such an assumption is not strictly correct, and it should be possible to define a method that takes into account this dependence. However, the independence assumption allows for a faster selection.

Comparison with other labeling methods. In [78] each map unit is labeled with the word that is most similar to the model vector. This results in consecutive areas labeled with the same word. However, some of the cluster areas achieved in this way are relatively large, and a further description of the large areas seems important, as was confirmed in a user study in [9]. Roussinov[98] offers a solution by letting users dynamically change the labeling used. Despite the possible usefulness of user-controlled modifications, obtaining an initial labeling that is as good as possible remains important. Merkl and Rauber have proposed a method called LabelSOM [87] for labeling individual SOM units. The method analyzes the components of SOM model vectors and selects as labels the terms corresponding to components that show smallest deviation. The method has only been defined for providing labels for individual map units, not for larger map areas.

4.3.4 Search facility

Especially when browsing large document maps it may be difficult to decide where to start browsing the map. A search facility has been implemented by which suitable starting points for exploration can be located. The description of interest written by the user—either a whole document or a few words—is encoded as a document (see Section 4.2.1), and a number of best-matching map units are marked on the display with circles the radius of which conveys the goodness of the match (see Figure 5). It should be noted that this facility does not perform *document retrieval*, i.e. return the best-matching documents, but only returns the best-matching map units (however, the facility could be extended to perform document retrieval as described in 5.1.1).

The facility is implemented using a client-server architecture: A search server holds the map reference vectors in memory, and when a search is initiated, a client program encodes the query as a document vector, passes it to the server, and asks for a number of the best-matching map units. Upon receiving the results the client draws them on existing static map images, constructs an appropriate HTML page and returns it to the WWW browser. HTTP Cookies are used for keeping track of users and the performed searches so that the zoomed images can be marked with appropriate search results upon requests for browsing. It is feasible to use the search facility—in the current implementation performing a search takes 4–10 seconds on the largest map of 7 million documents, and considerably less on smaller maps. In exploration of the results there is no noticeable delay compared to browsing without a search.

4.4 Evolution of the WEBSOM project

The initial experiments and public demonstrations on the WEBSOM consisted of maps of articles in Usenet discussion groups (see e.g. [41, 40, 67, 55], Publications 2, 1 and 4). Further experiments with various materials included maps of scientific abstracts (in [72] and in Publication 5), a map and a public demonstration on Finnish news articles (Publication 3) including a search facility, and maps of patent abstracts (a small experiment in Publication 3; a very large map in [68] and in Publication 7). Table 1 provides an overview of sizes of the largest collections handled each year. Experiments with the various data sets

Table 1: Sizes of document collections organized by the WEBSOM method at each year.

Year	Num. of documents	Num. of SOM units
1996	5,000-131,000	768-50,000
1997	1,000,000	100,000
1999	7,000,000	1,000,000

show that the method can be successfully applied to organizing both very small and very large collections, to colloquial discussions and to carefully written scientific documents, as

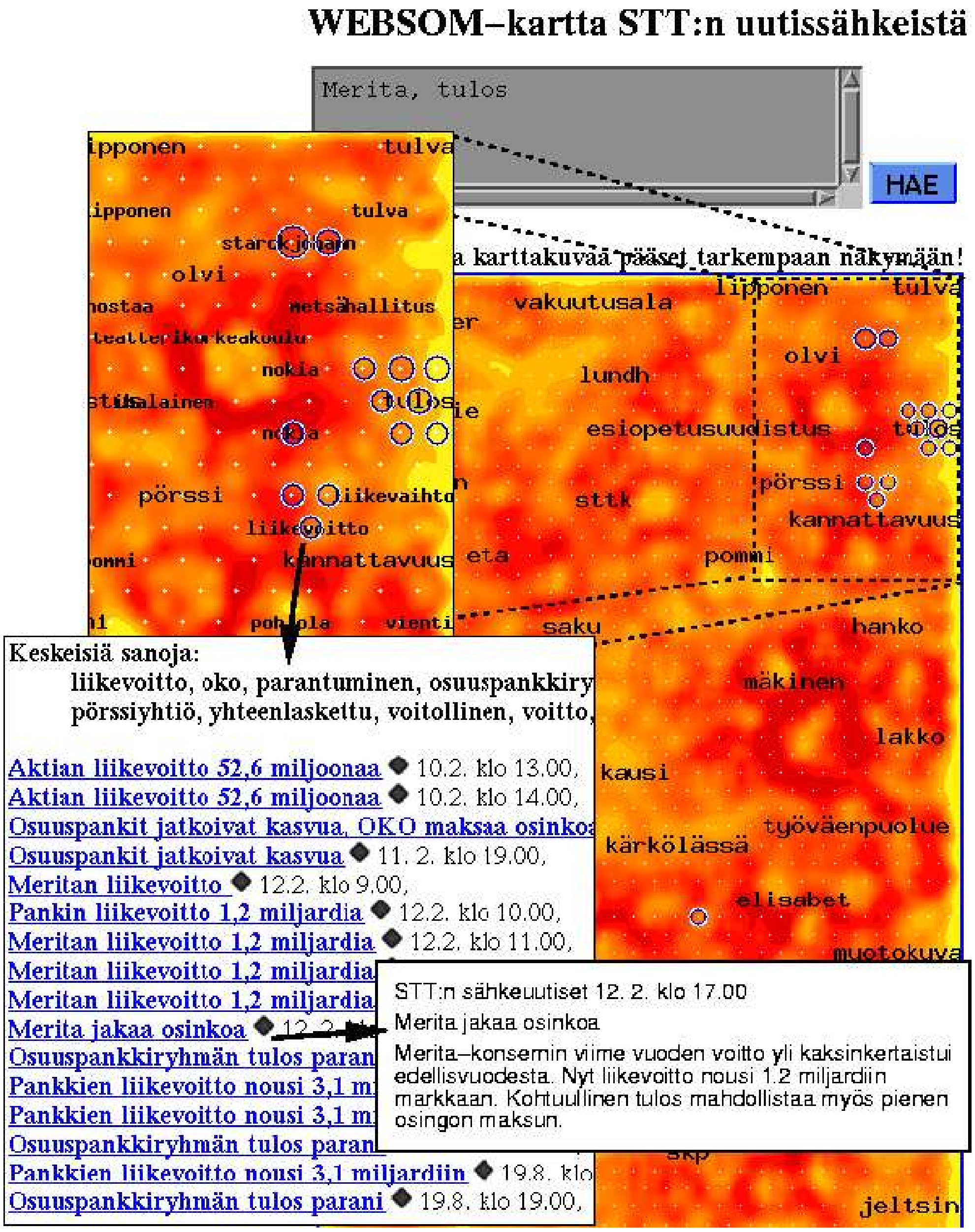


Figure 5: The user was interested in the financial situation of the Merita bank and typed in the query "Meritan tulos" (approximate translation: Merita's financial status). Three main clusters of results appear on the map: one containing news about the financial status of Merita and other banks, another discussing Merita buying and selling other companies, and a third cluster where announcements of financial results of many large companies appear.

well as to texts both in Finnish (see Publication 3) and in English. Various applications of the document maps were described in [39] and [43] and the different kinds of user tasks were explored further in Publication 6.

The methodological improvements were developed as follows: the word category map was used for document encoding in all of the earlier articles, including [41, 40, 67, 55, 73], Publications 2, 1. The word category map was examined in detail in [37]. The vector space model was used for document encoding in [72]. Various speedups for larger collections have been described in [67], [64], [51], [59], and Publication 4. Random projection was studied for dimensionality reduction in document encoding in [53] and applied to projecting word categories e.g. in [65] and in Publication 4 and to projecting document vectors e.g. in Publication 7. The entropy-based weighting was described in [67] and in Publication 4. The labeling method was first described in Publication 5 and applied to labeling several maps reported e.g. in Publication 6, Publication 3, [68], and Publication 7. The speeded winner search was described in [54]. The search facility for locating starting-points for exploration has been utilized since 1997, for a description, see e.g. Publication 7. The method for utilizing document maps for performing speeded document retrieval and the experiments were reported in Publication 8.

4.5 Related work on document maps

In an early study Lin formed a small map of scientific documents based on the words that occurred in the titles [78, 76] and later extended the method to full-text documents [77]. Scholtes has utilized the SOM in constructing a neural document filter and a neural interest map [107]. Merkl has organized software library components [82, 83, 84] and studied hierarchical document map formation [85, 86, 88]. Document maps have also been created by Zavrel [119]. A system rather similar in appearance to the WEBSOM has been used to organize collections of scientific articles in the field of astronomy [75, 95]. The Arizona AI group has utilized the SOM for categorizing Internet documents to aid in searching and exploration in a system called the ET-Map [11, 92], for adaptive visualization of search results [100, 99], and as part of a specialized application for medical data mining on the Internet [46]. Recently a commercial system was described in [109] that applies the SOM for creating document maps.

Hierarchical maps versus a single, large map. Instead of constructing a single, large map, the document collection may be organized on a hierarchy of layers that consist of distinct, smaller SOM:s, as has been done e.g. in [11, 85, 86, 88]. The hierarchical approach can be used to reduce computational complexity compared to the standard SOM algorithm (however, on large, continuous maps such reductions can be achieved with various speedups as described in Sec. 4.1.2). It may also be argued that the hierarchical directory structure is easier and more intuitive for users that are used to such structures. On the other hand, since in the hierarchical approach the data is partitioned into distinct clusters, the connections between documents found on different maps are lost. The impact of such partitioning is likely to depend on the amount of relevant, lost connections across partitions in the document collection. Finally, when moving across the different levels the lack of continuity might be confusing to the user. In any case, these are topics that require further study using controlled experiments as well as the design of appropriate evaluation criteria that are suitable for the visual exploration task.

5 USING DOCUMENT MAPS IN TEXT MINING

The possible uses of document maps are discussed below in terms of the three major tasks in text mining that were introduced in section 2: *searching, browsing*, and *visualization*.

5.1 Performing searches

5.1.1 Clustering documents for improving retrieval

It is important to develop methods that can speed up the search process while maintaining high perceived quality. In Section 4.3.4 a search facility was described for locating suitable starting-points for exploration of the document map. Moreover, a recent experiment described in Publication 8 shows that document maps can be utilized successfully for speeding up information retrieval, and even improved precision may be obtained on some collections.

Users typically look at only a small number of best documents returned by a search engine. The range of high precision and low recall is therefore most important in the evaluation and design of information retrieval methods. If the task is to find a small number of best hits, and to return them in ranked order, the document maps can be utilized for searching as follows: (1) *pre-select:* locate the *best-matching model vectors* for a query, and (2) *refine:* perform a full search among the K first documents found in the units corresponding to the best-matching models.

The retrieval performance of the method was evaluated on the CISI reference collection [14] of 1460 documents and 76 queries using *average precision*, a measure which takes into account the order of relevant and non-relevant documents in ranked result lists. The results indicate an equal or improved retrieval performance when compared to VSM and to LSI.

As pointed out in Section 4.1 the SOM model vectors form an approximate model of the data distribution density in the input space, performing a clustering of the document space, that acts as a smoothing on the documents: the initial search does not locate individual documents but *topical clusters*. The clusters may contain relevant documents that would not rank very high if compared directly with the query, but may nevertheless be relevant, and after the pruning carried out by pre-selection, these documents may rank rather high on the result lists. It is likely that this property of the document map improves searches especially when the queries share only few words with some of the desired documents, i.e., in the presence of noise.

Moreover, a speedup is achieved, since the total number of vector comparisons is always considerably smaller than the number of documents. Furthermore, with large collections it may be possible to keep the model vectors in main memory even when the full collection of document vectors could not fit in memory, causing even larger actual speedups than the differences in the complexity suggest. However, in the preliminary experiment the test collection was very small, and various speedups, including random projection of document vectors and magnification of maps, were not applied. The retrieval performance of the suggested search method should therefore be confirmed and the actual speedups in computation be measured in experiments on large collections. Alternative ways of performing the search could be studied as well. In a related study Kurimo [71] utilized SOM for smoothing of document vectors in retrieval of spoken documents.

It is claimed that still further improvement, perceived by a user but not readily measurable within the IR framework, would be achieved by visualizing the locations of the best hits on the document map display and by enabling exploration of the document map.

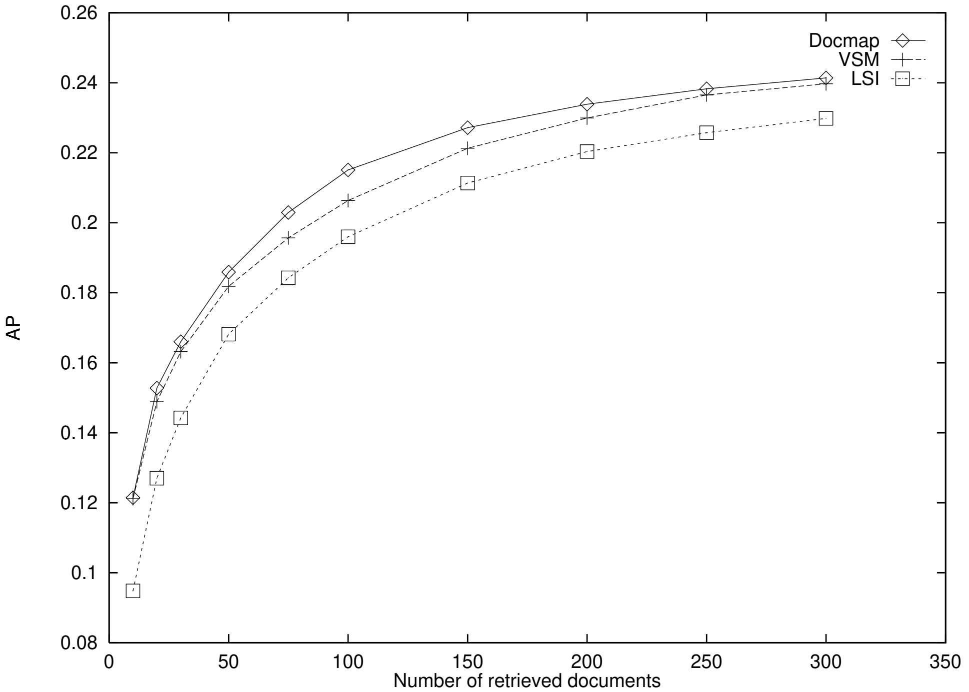


Figure 6: The retrieval performance of the document map method is compared to that of the VSM and the LSI. LSI was calculated using 150 latent dimensions. The size of the document set requested at the pre-selection stage was 300; the number was chosen based on prior comparison of different values using random selection of half of the queries from the test set of 76 queries. The rest of the queries were used to compare the methods. The methods were compared using *non-interpolated average precision (AP)* and plotted for several amounts of retrieved documents.

In such an environment the user may visually confirm the results and interactively refine the search further by concentrating on some hit, and by locating similar documents on the document map. This refinement operation is of complexity $O(1)$ only, since the map and the exploration interface have been created off line, independent of a particular search.

5.1.2 Maps of search results

Often the hits of Internet search engines fall into several domains, e.g. according to various meanings of each polysemous word, or according to various types of data, e.g. personal home pages, institutional pages, company advertisements, scientific articles, etc. However, the various categories are not readily apparent from the result lists organized by relevance.

Results of searches performed on Internet search engines can be organized dynamically using the SOM onto a document map. In this way, results falling in various domains or styles may be clustered together, making browsing easier. Furthermore, the visualization can aid in forming an overall view of the topic.

Roussinov has applied SOM for organizing document maps of search results: several hundreds of search results are obtained from an Internet search engine, the texts are encoded as vectors, and the SOM is applied to construct a document map [100, 99]. Dynamic and adaptive operations are included by which users can control the map creation, e.g. by removal of individual terms from the document representations. Each interactive operation results in re-building the map.

In principle, this approach is feasible, and potentially very valuable for the user. However, when the results of external search servers are mapped, obtaining and transmitting the result lists takes considerable time (several minutes for 400 documents in Roussinov's example) although the subsequent map construction itself is fast. Possible solutions could be (1) the construction of the maps at the search server site, or (2) the formation and visualization of an initial map based on the first set of results and an incremental refinement of the map as subsequent results are obtained from the search server. However, these strategies have not been explored in detail.

5.2 Text exploration

Document maps of large collections could be utilized by owners of large, public databases for offering world-wide access to their collections. Databases with large amounts of potentially valuable data which is difficult to make sense of using off-the-shelf methods include articles in the medical sciences and patent abstracts. An exploration example of the map of 7 million patent abstracts is presented in Figure 7. As shown in the figure, a suitable starting-point for exploration can be searched using the document search facility described in Section 4.3.4.

Alternatively, any search engine can be used for information retrieval, and the results can be depicted on the map. By visualizing the distribution of the search results in a topically ordered display, the map offers additional information regarding the results that cannot be conveyed by the one-dimensional list of results. An example of visualizing the results of an external search engine on the map is presented in Figure 8.

5.3 Visual domain models

In visualization of information something familiar is used to illustrate something yet unfamiliar. With a document map of an unknown collection the source of familiar information are the label words and the visual metaphor of portraying landscape with a map. Also locating map areas related to an individual query or a document fall into this category.

However, if the map is already known intimately, properties of new items and collections can be illustrated a bit like in the *scatter-plot*, by dots marking individual items on the map display (for an example, see Fig. 9). In the ordinary scatter-plot the two or three axes offer meaningful information against which the new data is portrayed. However, on the map display, instead of a small number of meaningful dimensions there may be *several meaningful clusters* which form a basis for interpreting the new data.

5.3.1 Depicting new information on a familiar map

Publication 6 describes a small experiment of utilizing a familiar topical map to studying visually the properties of an incoming stream of email from a similar subject area. The map was based on a collection of discussion articles from the Usenet discussion group `comp.ai.neural-nets`[22] and the emails were taken from the `connectionists` email list.

[22] The map can be explored at
`http://websom.hut.fi/websom/comp.ai.neural-nets_new/html/root.html`.

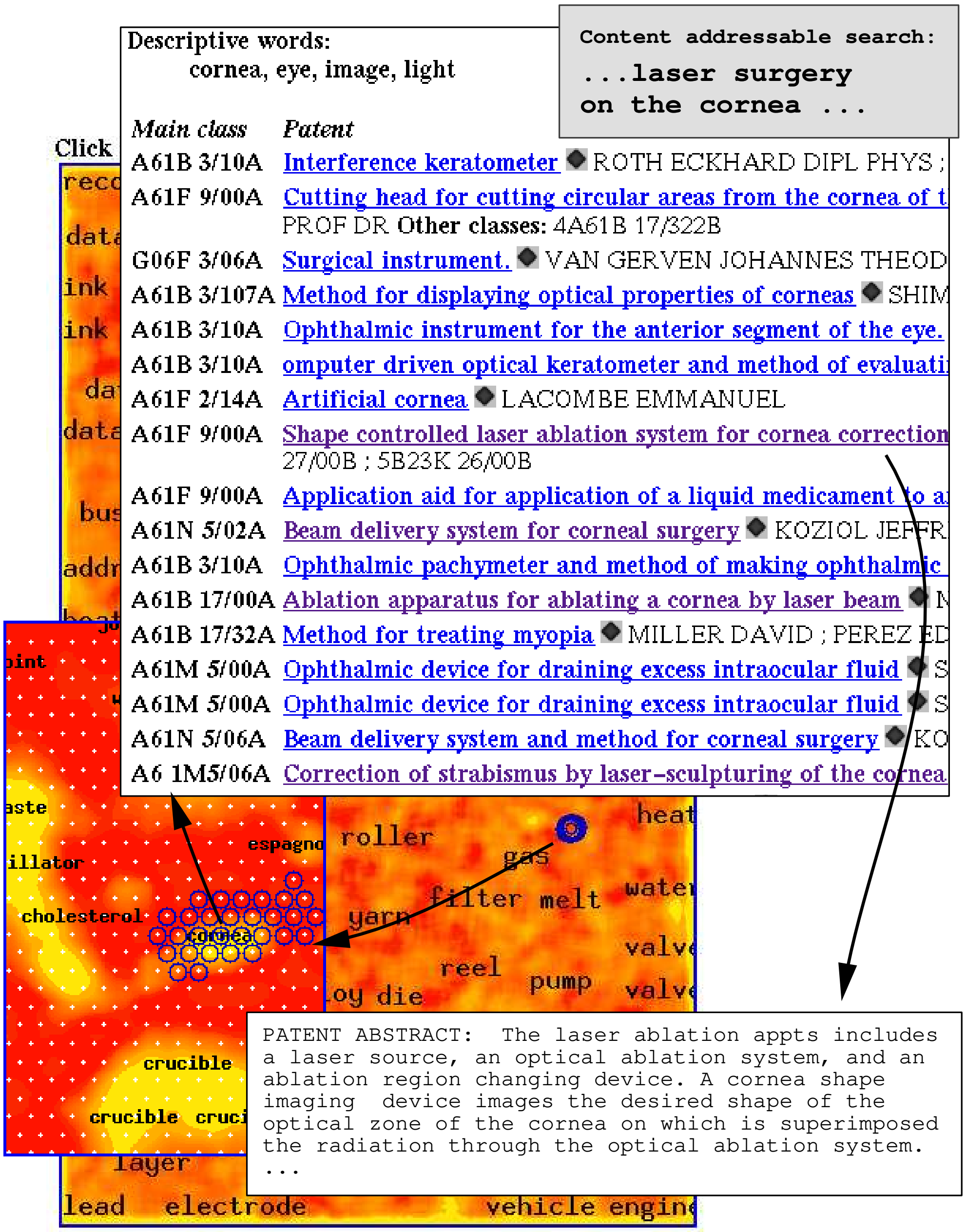


Figure 7: Content addressable search was utilized to find information on laser surgery on the cornea of the eye. The best-matching locations are marked with circles. Zooming on the area reveals a small cluster of map units that contains patent abstracts mostly about the cornea of the eye, and of surgical operations on it. Several abstracts concerned with the description of interest, i.e. laser surgery on the cornea, are found in the best-matching units.

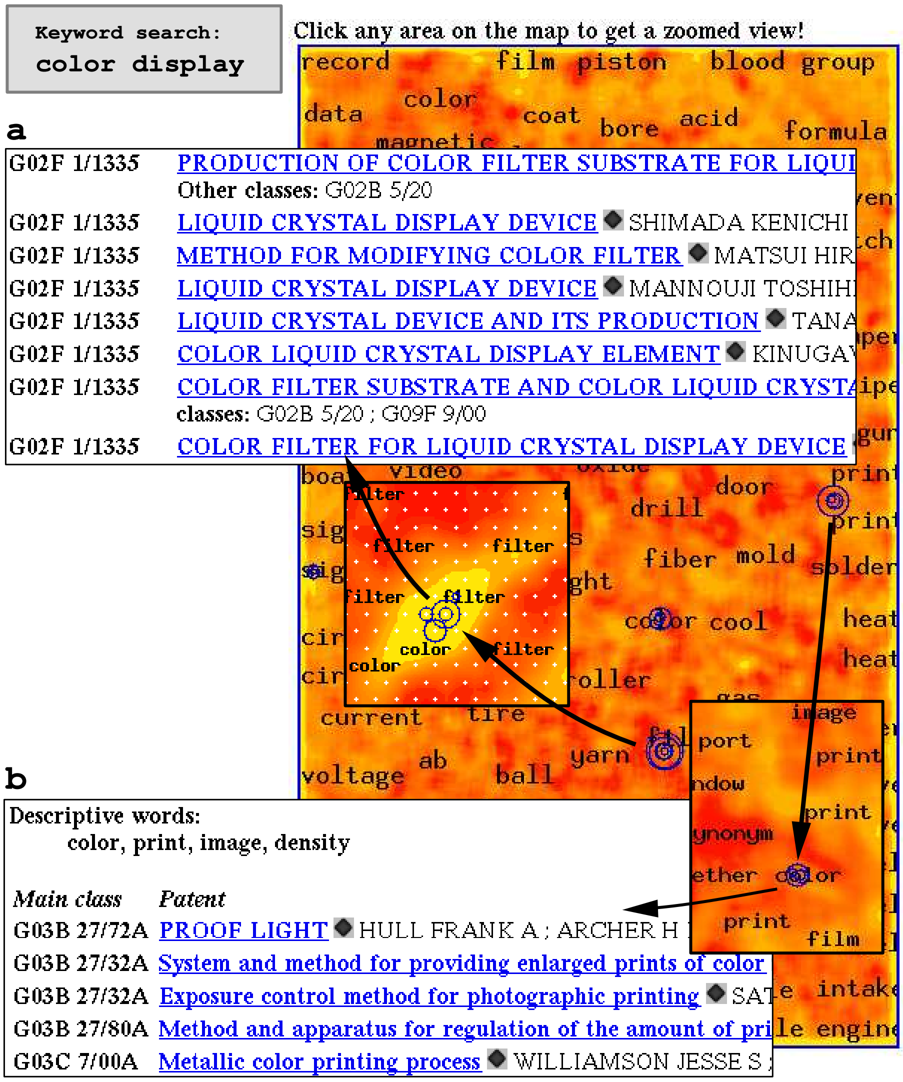


Figure 8: The keyword search mode was utilized to find information on color displays. 30 best-matching units were marked on the display with circles the size of which indicates the goodness of the match. As seen from the map display, the matches are distributed into several tight “clusters” found in different regions of the map. From two of these clusters the partial contents of a matching unit are shown in the insets. Closer inspection of the units reveals different aspects of color and displays. Unit **a** features a considerable number of abstracts about color filters used in building LCD displays, whereas in **b** one finds technology related to displaying colors when printing documents (the “Descriptive words” lists were found for each map unit using the automatic keyword selection method described in 4.3.3). The user, who probably did not have printing in mind when formulating the query, can then concentrate on the other clusters.

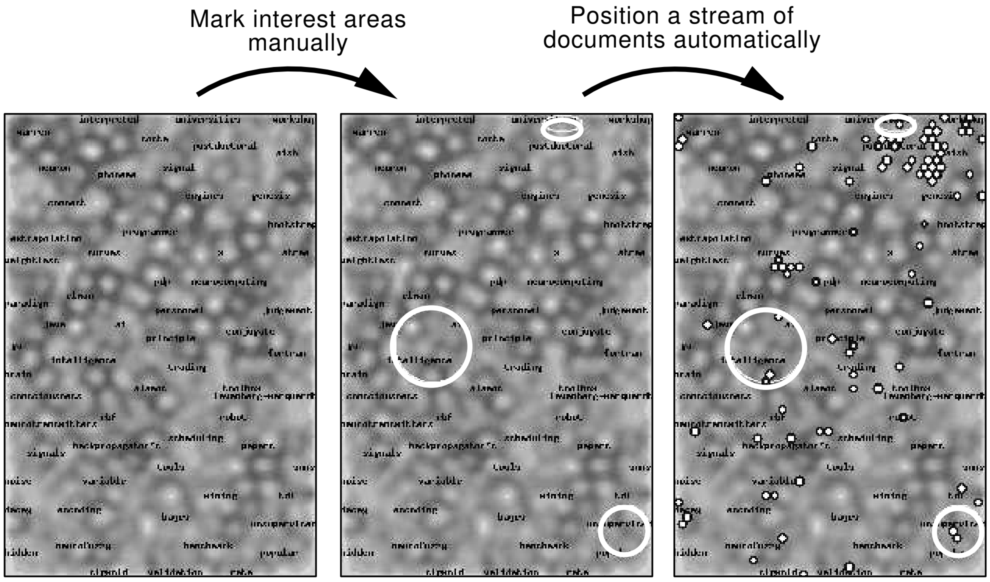


Figure 9: A sketch of an interactive, visual tool for constructing filters for handling streams of incoming documents. First the user selects the document map areas to be used as the filter (large white ellipses) and next new documents are projected on the map (small white circles). Finally, either the system explicitly filters out documents located outside the specified areas, or a quick visual filtering is performed by the user looking at the map display. In the latter case the filter borders can be tentative, and the user may choose to read a new document outside a filter area after all, for example, if no documents appear inside the filter borders.

It was known before that the topical domains of the collections were similar. However, by visualizing the emails on the familiar document map a more detailed idea of the similarities and differences could be obtained: for example, conference announcements seemed to be prominent in the email list whereas in the discussion newsgroup they have a smaller role. In contrast, the map of the Usenet group has an area of philosophical discussions on neural networks and artificial intelligence, but only a few documents from the email list excerpt were found to fall near this area.

5.3.2 Visual filter construction

A familiar document map display could also be used as a tool for *constructing graphical filters*. As depicted in Figure 9, interesting (or uninteresting) map areas can be selected, and new incoming documents such as new emails can be added on the map with further processing dictated by the filtering need. The document map can either be used as an easy interface for constructing filters, after which the filtering is performed automatically, or the map can act as an *implicit filter* by visualizing the similarity of the incoming articles with the interesting (or non-interesting) map areas[23].

[23]The interactive filter construction capability is currently not included in the WEBSOM browsing tool. However, the documents were positioned on the map display using standard WEBSOM programs.

5.3.3 Personal interest maps

The documents accumulated by an individual provide a description of the person's interests that is extremely familiar and relevant to the individual. A visual model of personal interests could be used as an utterly familiar frame of reference on which large amounts of new information can be rapidly conveyed to the individual.

Such compact and abstract models of interest could be utilized as *user profiles* e.g. for identifying similarly inclined people, as well for detailed evaluation of similarities and dissimilarities between the interests of two people (the similarity of the models could be compared e.g. using a method described in [58]). Unlike the often sensitive text material in the texts themselves, the more abstract and compact interest profiles formed by the maps could be more freely distributed by the individual. The level of abstraction achieved with a document map can be controlled by the map size.

Lin [77] has utilized the SOM for arranging a personal document collection onto a document map to provide a convenient exploration interface to the collections, and to explore the interests.

5.4 Comparison of document maps with other approaches

5.4.1 Comparison to similar visualization tools

In Galaxies (project SPIRE; for an overview, see e.g.[117]) the documents are expressed in terms of distances to a small number of cluster centroids, and then PCA is applied to project the documents onto two dimensions. The visualization of individual documents thus places each document on the display so that the position reflects its similarity *to every cluster centroid* on the display), not to each other document. The method thus attempts to preserve the global distances rather than the local ones. In contrast, the document maps organized with the SOM attempt to preserve local distances between similar documents, which in part accounts for the promising results obtained in utilizing document maps for information retrieval (see Sec. 5.1.1).

5.4.2 Comparison to manually organized hierarchies

Chen et al. [9] performed a usability study comparing browsing the entertainment subcategory of the hierarchical *Yahoo!* organized by manual effort, or the same document collection organized automatically with the SOM (the ET-map containing 110,000 documents [11]). The 34 subjects were asked to browse for "something of interest to you", the task was described as "window shopping", and the users were advised to start without a specific goal in mind. The users were asked to think out loud, describing the reasoning behind their choices. The authors found that the document map was best suited to browsing tasks that were very broad, and to situations in which subjects liked to skip between topic categories. In particular many subjects especially liked the visual and graphical aspects of the map. The unfamiliar associative mental model of the SOM was troubling to some subjects, especially to those who started with *Yahoo!* and then proceeded to the ET-map—on the other hand some other subjects found the differing model interesting or useful. Based on success rates in browsing and a detailed analysis of user feedback the authors conclude that their SOM-based browsing prototype compares rather favorably with browsing the hierarchical, manually organized *Yahoo!*. The fact that an automatic organization such as a document map can offer even nearly as good results as an organization obtained with considerable utilization of manual labor is indeed noteworthy.

5.4.3 Comparison to search-oriented approaches

Implementation of a text mining tool can be viewed as a compromise in how limited resources are allocated between searching, browsing, and visualization. Previously it has been argued why considerable allocation of resources to the visualization and exploration aspects is indeed useful (cf. Section 2), and how the usefulness manifests itself specifically in document maps (cf. Section 5). In particular, it has been demonstrated that also in the search task a WEBSOM document map performs at least comparably to other state-of-the art search approaches, even when the visualization ability of the map is not utilized.

Especially with very large collections that push the limits of computer systems in every way, it is an advantage if a single set of data structures or indices can be utilized for all of the identified tasks related to text mining. In the WEBSOM method, visualization, exploration and search are all implemented using the same framework.

When compared to tools exhibiting only search capability the combination of visualization, exploration and search offered by WEBSOM provides most help in an environment where the system is expected to service both vague and clearly defined information needs, as well as both specific and broader needs. In information needs that are very specific and well-understood the best methods that specialize in the search problem alone may outperform the general-purpose document map approach. In contrast, the most additional utility from document maps is likely to be experienced in the vague and broader information needs.

6 CONCLUSION

It has been shown that document maps can be utilized for visualization and exploration of text collections of various sizes, various text types, different languages, as well as many different kinds of user tasks in text mining. Subjective evaluation of the maps indicates that the WEBSOM method is able to uncover a kind of topical organization of the text material in an unsupervised manner. Based on the experiments described in this thesis and related research it can be concluded that the method is scalable and can be successfully used for visualizing and exploring very large document collections. In addition, its performance in the search task has been evaluated using a small, standard document collection, and found to be comparable with the performance of state of the art search approaches.

The visual metaphor of maps and landscapes seems to offer an intuitive framework for performing the major tasks studied in the field of text mining, namely searching, exploration, and visualization. Furthermore, the visual landscapes can be utilized for tasks that have not received much attention before, possibly for lack of suitable methods. Examples of such tasks are: relating collections of text to other collections, and visualizing various kinds of unfamiliar information on familiar landscapes. However, the possible uses of document maps have so far only been touched on and even less research has been carried out with actual users in real situations.

Directions for future research include the development of suitable evaluation principles for the various user tasks described in this thesis, as well as the application of such principles in evaluating document maps and text visualization models in general. A continuing challenge is to develop richer and yet practically computable models for representing the content of natural language utterances.

References

[1] J. Allan, J. Callan, W. B. Croft, L. Ballesteros, D. Byrd, R. Swan, and J. Xu, "INQUERY does battle with TREC-6," in *Proceedings of the Sixth Text Retrieval Conference (TREC6)*, (Gaithersburg, MD), National Institute of Standards and Technology (NIST), 1998.

[2] J. Allen, *Natural Language Understanding.* The Benjamin/Cummngs Publishing Company, 1995.

[3] R. Baeza-Yates and B. Ribeiro-Neto, eds., *Modern Information Retrieval.* Addison Wesley Longman, 1999.

[4] A. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[5] K. Bollacker, S. Lawrence, and C. L. Giles, "CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications," in *2nd International ACM Conference on Autonomous Agents*, pp. 116–123, ACM Press, May 1998.

[6] A. Brüggemann-Klein, R. Klein, and B. Landgraf, "BibRelEx—exploring bibliographic databases by visualization of annotated content-based relations," *D-Lib Magazine*, vol. 5, November 1999.

[7] J. P. Callan, W. B. Croft, and S. M. Harding, "The inquery retrieval system," in *Proceedings of the 3rd International Conference on Database and Expert Systems*, 1992.

[8] E. Charniak, *Statistical Language Learning.* MIT Press, 1993.

[9] H. Chen, A. L. Houston, R. R. Sewell, and B. R. Schatz, "Internet browsing and searching: User evaluations of category map and concept space techniques," *Journal of the American Society for Information Science (JASIR)*, vol. 49, no. 7, pp. 582–603, 1998.

[10] H. Chen, B. Schatz, T. Ng, J. Martinez, A. Kirchhoff, and C. Lin, "A parallel computing approach to creating engineering concept spaces for semantic retrieval: The Illinois digital library initiative project," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1006.

[11] H. Chen, C. Schuffels, and R. Orwig, "Internet categorization and search: A self-organizing approach," *Journal of the Visual Communication and Image Representation*, vol. 7, pp. 88–102, 1996.

[12] V. Cherkassky and F. Mulier, *Learning from Data—Concepts, Theory, and Methods.* John Wiley & Sons, 1998.

[13] K. W. Church and W. A. Gale, "Inverse document frequency (IDF): A measure of deviations from Poisson," in *Proceedings of the Third Workshop on Very Large Corpora* (D. Yarowsky and K. Church, eds.), pp. 121–130, Massachusetts Institute of Technology, 1995.

[14] CISI-collection, "The CISI reference collection for information retrieval. 1460 documents and 76 queries." http://local.dcs.gla.ac.uk/idom/ir_resources/test_collections/cisi/, 1981.

[15] G. Deboeck and T. Kohonen, eds., *Visual Explorations in Finance and Investments Using Self-Organizing Maps.* Springer, 1998.

[16] S. Deerwester, S. T. Dumais, G. W. Furnas, and T. K. Landauer, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391–407, 1990.

[17] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering in large graphs and matrices," in *Proc. of the 10th ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA*, pp. 291–299, ACM, 1999.

[18] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Knowledge discovery and data mining: Towards a unifying framework," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (E. Simoudis, J. Han, and U. Fayyad, eds.), pp. 82–88, AAAI Press, 1996.

[19] S. Finch and N. Chater, "Unsupervised methods for finding linguistic categories," in *Artificial Neural Networks, 2*, pp. II–1365–1368, North-Holland, 1992.

[20] J. A. Fodor, *Concepts—Where Cognitive Science Went Wrong.* Oxford University Press, 1998.

[21] S. I. Gallant, "A practical approach for representing context and for performing word sense disambiguation using neural networks," *Neural Computation*, vol. 3, pp. 293–309, 1991.

[22] S. I. Gallant, "Methods for generating or revising context vectors for a plurality of word stems." U.S. Patent number 5,325,298, 1994.

[23] P. Gärdenfors, "Meaning as conceptual structures," Tech. Rep. LUCS 40, Lund University Cognitive Studies, 1995. ISSN 1101-8453.

[24] P. Gärdenfors, *Conceptual Spaces.* MIT Press, 2000.

[25] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis.* Chapman & Hall/CRC, 1995.

[26] L. Girardin, "Mapping the virtual geography of the world-wide web," in *Proceedings of the Fifth International World Wide Web Conference WWW5, May 6-10, Paris, France*, vol. Poster Proceedings, pp. 131–139, EPGL, 1996.

[27] G. Grefenstette, "Comparing two language identification schemes," in *JADT 1995, 3rd International conference on Statistical Analysis of Textual Data, Dec 11-13, 1995*, 1995.

[28] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining.* MIT Press, 2000.

[29] S. Havre, B. Hetzler, and L. Nowell, "ThemeRiver(TM): In search of trends, patterns, and relationships," in *Proc. IEEE Symposium on Information Visualization, InfoVis '99*, (San Francisco CA), October 1999.

[30] S. Haykin, *Neural Networks—A Comprehensive Foundation.* Prentice Hall, 2nd ed., 1999.

[31] M. A. Hearst, "Tilebars: Visualization of term distribution information in full text information access," in *Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems*, (Denver, CO), pp. 56–66, ACM, May 1995.

[32] M. A. Hearst, *Modern Information Retrieval*, ch. 10. User Interfaces and Visualization, pp. 257–324. Addison Wesley Longman, 1999.

[33] M. A. Hearst, "Untangling text data mining," in *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics*, 1999.

[34] R. Hendley, N. Drew, and A. Wood, "Narcissus: Visualizing information," in *Proc. IEEE Information Visualization '95*, (Los Alamitos, CA), pp. 90–96, IEEE Computer Press, 1995.

[35] T. Honkela, "Neural nets that discuss: A general model of communication based on self-organizing maps.," in *Proceedings of International Conference on Artificial Neural Networks (ICANN-93)*, pp. 408–411, Springer-Verlag, 1993.

[36] T. Honkela, "Learning to understand - general aspects of using self-organizing maps in natural language processing," in *Proceedings of the CASYS'97, Computing Anticipatory Systems*, pp. 563–576, American Institute of Physics, Woodbury, 1997.

[37] T. Honkela, *Self-Organizing Maps in Natural Language Processing.* PhD thesis, Helsinki University of Technology, Espoo, Finland, 1997.

[38] T. Honkela, "Self-organizing maps of words for natural language processing applications," in *Proceedings of International ICSC Symposium on Soft Computing, Nimes, France, September 17-19, 1997*, (Alberta, Canada), pp. 401–407, ICSC Academic Press, 1997.

[39] T. Honkela, S. Kaski, T. Kohonen, and K. Lagus, "Self-organizing maps of very large document collections: Justification for the WEBSOM method," in *Classification, Data Analysis, and Data Highways* (I. Balderjahn, R. Mathar, and M. Schader, eds.), pp. 245–252, Springer, 1998.

[40] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "Exploration of full-text databases with self-organizing maps," in *Proceedings of the ICNN96, International Conference on Neural Networks*, vol. I, pp. 56–61, IEEE Service Center, 1996.

[41] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "Newsgroup exploration with WEBSOM method and browsing interface," Tech. Rep. A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.

[42] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "WEBSOM—self-organizing maps of document collections," in *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pp. 310–315, Helsinki University of Technology, Neural Networks Research Centre, 1997.

[43] T. Honkela, K. Lagus, and S. Kaski, "Self-organizing maps of large document collections," in *Visual Explorations in Finance with Self-Organizing Maps* (G. Deboeck and T. Kohonen, eds.), pp. 168–178, Springer, 1998.

[44] T. Honkela, V. Pulkki, and T. Kohonen, "Contextual relations of words in Grimm tales analyzed by self-organizing map," in *Proceedings of ICANN-95, International Conference on Artificial Neural Networks*, vol. II, (Paris), pp. 3–7, EC2 et Cie, 1995.

[45] T. Honkela and A. M. Vepsäläinen, "Interpreting imprecise expressions: Experiments with Kohonen's self-organizing maps and associative memory.," in *Proceedings of ICANN-91, International Conference on Artificial Neural Networks*, vol. I, pp. 897–902, North-Holland, 1991.

[46] A. L. Houston, H. Chen, S. M. Hubbard, B. R. Schatz, T. D. Ng, R. R. Sewell, and K. M. Tolle, "Medical data mining on the internet: Research on a cancer information system," *Artificial Intelligence Review*, 1999.

[47] A. Hyvärinen, "Independent component analysis: A neural network approach," *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 88*, October 1997. DTech Thesis, Helsinki University of Technology, Finland.

[48] A. Hyvärinen, "Survey on independent component analysis," *Neural Computing Surveys*, no. 2, pp. 94–128, 1999.

[49] C. L. Isbell and P. Viola, "Restructuring sparse high dimensional data for effective retrieval," in *Conference on Neural Information Processing (NIPS'98)*, pp. 480–486, 1998.

[50] A. K. Jain and R. C. Dubes, *Algorithms for clustering data.* Prentice Hall, 1988.

[51] S. Kaski, "Computationally efficient approximation of a probabilistic model for document representation in the WEBSOM full-text analysis method," *Neural Processing Letters*, vol. 5, pp. 139–151, 1997.

[52] S. Kaski, "Data exploration using self-organizing maps," *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82*, March 1997. DTech Thesis, Helsinki University of Technology, Finland.

[53] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," in *Proceedings of IJCNN'98, International Joint Conference on Neural Networks*, vol. 1, pp. 413–418, IEEE Service Center, 1998.

[54] S. Kaski, "Fast winner search for SOM-based monitoring and retrieval of high-dimensional data," in *Proceedings of ICANN99, Ninth International Conference on Artificial Neural Networks*, vol. 2, pp. 940–945, IEE, 1999.

[55] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, "Creating an order in digital libraries with self-organizing maps," in *Proceedings of WCNN'96, World Congress on Neural Networks, September 15-18, San Diego, California*, pp. 814–817, Lawrence Erlbaum and INNS Press, 1996.

[56] S. Kaski, J. Kangas, and T. Kohonen, "Bibliography of self-organizing map (SOM) papers: 1981–1997," *Neural Computing Surveys*, vol. 1, no. 3&4, pp. 1–176, 1998. Available in electronic form at http://www.icsi.berkeley.edu/~jagota/NCS/: Vol 1, pp. 102–350.

[57] S. Kaski and T. Kohonen, "Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world," in *Neural Networks in Financial Engineering. Proceedings of the Third International Conference on Neural Networks in the Capital Markets, London, England, 11-13 October, 1995* (A.-P. N. Refenes, Y. Abu-Mostafa, J. Moody, and A. Weigend, eds.), pp. 498–507, World Scientific, 1996.

[58] S. Kaski and K. Lagus, "Comparing self-organizing maps," in *Proceedings of International Conference on Artificial Neural Networks (ICANN96), Lecture Notes in Computer Science*, vol. 1112, (Berlin), pp. 809–814, Springer, 1996.

[59] S. Kaski, K. Lagus, T. Honkela, and T. Kohonen, "Statistical aspects of the WEBSOM system in organizing document collections," *Computing Science and Statistics*, vol. 29, pp. 281–290, 1998. (Scott, D. W., ed.), Interface Foundation of North America, Inc.: Fairfax Station, VA.

[60] J. Kekäläinen, *The effects of Query Complexity, Expansion and Structure on Retrieval Performance in Probabilistic Text Retrieval.* PhD thesis, Dept. of Information Studies, University of Tampere, 1999.

[61] T. Kohonen, "Self-organizing formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[62] T. Kohonen, *Self-Organizing Maps.* Springer, 1995.

[63] T. Kohonen, "The speedy SOM," Tech. Rep. A33, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.

[64] T. Kohonen, "Exploration of very large databases by self-organizing maps," in *Proceedings of ICNN'97, International Conference on Neural Networks*, pp. PL1–PL6, IEEE Service Center, 1997.

[65] T. Kohonen, "Self-organization of very large document collections: State of the art," in *Proceedings of ICANN98, the 8th International Conference on Artificial Neural Networks* (L. Niklasson, M. Bodén, and T. Ziemke, eds.), vol. 1, pp. 65–74, Springer, 1998.

[66] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "SOM_PAK: The Self-Organizing Map program package," Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, Jan. 1996.

[67] T. Kohonen, S. Kaski, K. Lagus, and T. Honkela, "Very large two-level SOM for the browsing of newsgroups," in *Proceedings of ICANN96, International Conference on Artificial Neural Networks, Bochum, Germany, July 16-19, 1996* (C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, eds.), Lecture Notes in Computer Science, vol. 1112, pp. 269–274, Springer, 1996.

[68] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive text document collection," in *Kohonen Maps* (E. Oja and S. Kaski, eds.), pp. 171–182, Elsevier, 1999.

[69] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE*, vol. 84, pp. 1358–1384, 1996.

[70] K. Koskenniemi, *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production.* PhD thesis, University of Helsinki, Department of General Linguistics, 1983.

[71] M. Kurimo, "Indexing audio documents by using latent semantic analysis and som," in *Kohonen Maps*, pp. 363–374, Elsevier, 1999.

[72] K. Lagus, "Map of WSOM'97 abstracts—alternative index," in *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pp. 368–372, Helsinki University of Technology, Neural Networks Research Centre, 1997.

[73] K. Lagus, T. Honkela, S. Kaski, and T. Kohonen, "WEBSOM – a status report," in *Proceedings of STeP'96, Finnish Artificial Intelligence Conference* (J. Alander, T. Honkela, and M. Jakobsson, eds.), pp. 73–78, Finnish Artificial Intelligence Society, 1996.

[74] S. Lawrence and C. L. Giles, "Searching the world wide web.," *Science*, vol. 280, pp. 98–100, Aprin 1998.

[75] S. Lesteven, P. Ponçot, and F. Murtagh, "Neural networks and information extraction in astronomical information retrieval," *Vistas in Astronomy*, vol. 40, p. 395, 1996.

[76] X. Lin, "Visualization for the document space," in *Proceedings of Visualization '92*, (Los Alamitos, CA, USA), pp. 274–81, Center for Comput. Legal Res., Pace Univ., White Plains, NY, USA, IEEE Comput. Soc. Press, 1992.

[77] X. Lin, "Map displays for information retrieval," *Journal of the American Society for Information Science*, vol. 48, pp. 40–54, 1997.

[78] X. Lin, D. Soergel, and G. Marchionini, "A self-organizing semantic map for information retrieval," in *Proceedings of 14th Ann. International ACM/SIGIR Conference on Research & Development in Information Retrieval*, pp. 262–269, 1991.

[79] K. Linden, "Language applications with finite-state technology," *International Journal of Corpus Linguistics*, vol. 2, pp. 1–7, 1997.

[80] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts, 1999.

[81] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *Working Notes of the 1998 AAAI/ICML Workshop on Learning for Text Categorization*, pp. 41–48, AAAI Press, 1998.

[82] D. Merkl, "Structuring software for reuse - the case of self-organizing maps," in *Proceedings of IJCNN-93-Nagoya, International Joint Conference on Neural Networks*, vol. III, (Piscataway, NJ), pp. 2468–2471, JNNS, IEEE Service Center, 1993.

[83] D. Merkl, "Content-based software classification by self-organization," in *Proceedings of ICNN'95, IEEE International Conference on Neural Networks*, vol. II, (Piscataway, NJ), pp. 1086–1091, IEEE Service Center, 1995.

[84] D. Merkl, "Exploration of text collections with hierarchical feature maps," in *Proceedings of SIGIR'97, 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (New York), ACM, 1997.

[85] D. Merkl, "Lessons learned in text document classification," in *Proc. of Workshop on Self-Organizing Maps 1997 (WSOM'97)*, (Espoo, Finland), pp. 316–321, Helsinki University of Technology, Neural Networks Research Centre, 1997.

[86] D. Merkl, "Document classification with self-organizing maps," in *Kohonen Maps*, pp. 183–195, Elsevier, 1999.

[87] D. Merkl and A. Rauber, "Automatic labeling of self-organizing maps for information retrieval," in *Int'l Conference on Neural Information Processing (ICONIP'99), November 16-20.Perth, WA.*, 1999.

[88] D. Merkl and A. Rauber, "Uncovering the hierarchical structure of text archives by using an unsupervised neural network with adaptive architecture," in *Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD'2000)*, 2000.

[89] R. Miikkulainen, *Subsymbolic natural language processing: an integrated model of scripts, lexicon, and memory.* MIT Press, 1993.

[90] R. Miikkulainen, "Dyslexic and category-specific aphasic impairments in a self-organizing feature map model of the lexicon," *Brain and Language*, vol. 59, pp. 334–366, 1997.

[91] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: an on-line lexical database.," *International Journal of Lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[92] R. Orwig, H. Chen, and J. F. Nunamaker, "A graphical, self-organizing approach to classifying electronic meeting output," *Journal of the American Society for Information Science*, vol. 48, no. 2, pp. 157–170, 1997.

[93] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," in *Proc. 17th ACM Symposium on the Principles of Database Systems, Seattle*, ACM Press, 1998.

[94] S. Pinker, *The Language Instinct.* Penguin Books, 1994.

[95] P. Poinçot, S. Lesteven, and F. Murtagh, "A spatial user interface to the astronomical literature," *Astronomy and Astrophysics*, vol. 130, pp. 183–191, 1998.

[96] J. Raukko, "An "intersubjective" method for cognitive-semantic research on polysemy: The case of *get*," in *Cultural, Psychological and Typological Issues in Cognitive Linguistics* (M. K. Hiraga, C. Sinha, and S. Wilcox, eds.), John Benjamins B.V., 1999.

[97] H. Ritter and T. Kohonen, "Self-organizing semantic maps," *Biological Cybernetics*, vol. 61, pp. 241–254, 1989.

[98] D. Roussinov, *Information Foraging through Clustering and Summarization: a Self-Organizing Approach.* PhD thesis, University of Arizona, 1999.

[99] D. Roussinov, "Internet search using adaptive visualization," in *Conference on Human Factors in Computing Systems, Doctoral Consortium (SIGCHI'99)*, ACM, 1999.

[100] D. Roussinov and M. Ramsey, “Information forage through adaptive visualization,” in *The Third ACM Conference on Digital Libraries, June 23-26*, (Pittsburgh), pp. 303–304, 1998.

[101] G. Salton, *Automatic Information Organization and Retrieval.* McGraw-Hill, 1968.

[102] G. Salton, *Automatic Text Processing.* Addison-Wesley, 1989.

[103] G. Salton and C. Buckley, “Term weighting approaches in automatic text retrieval,” Tech. Rep. 87-881, Cornell University, Department of Computer Science, Ithaca, NY, 1987.

[104] G. Salton and M. J. McGill, *Introduction to modern information retrieval.* McGraw-Hill, 1983.

[105] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[106] M. Sarkar and M. H. Brown, “Graphical fisheye views,” *Commumications of the ACM,*, vol. 37, no. 12, pp. 73–84, 1994.

[107] J. C. Scholtes, *Neural Networks in Natural Language Processing and Information Retrieval.* PhD thesis, Universiteit van Amsterdam, Amsterdam, Netherlands, 1993.

[108] B. Shneiderman, “The eyes have it: A task by data type taxonomy for information visualizations,” in *Proceedings IEEE Visual Language*, (Boulder, CO), pp. 336–343, September 1996.

[109] S. Shumsky, “Navigation in databases using self-organizing maps,” in *Kohonen Maps*, pp. 197–206, Elsevier, 1999.

[110] J. Thomas, K. Cook, V. Crow, B. Hetzler, R. May, D. McQuerry, R. McVeety, N. Miller, G. Nakamura, L. Nowell, P. Whitney, and P. C. Won, “Human computer interaction with global information spaces - beyond data mining,” tech. rep., Pacific Northwest National Laboratory,Richland, WA 99352, 2000.

[111] E. R. Tufte, *The Visual Display of Quantitative Information.* Graphics Press, 1983.

[112] J. W. Tukey, *Exploratory Data Analysis.* Addison-Wesley, 1977.

[113] A. Ultsch, “Self-organizing neural networks for visualization and classification,” in *Information and Classification*, (London, UK), pp. 307–313, Springer, 1993.

[114] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, “Som toolbox for matlab 5,” Report A57, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, April 2000.

[115] S. Wermter, G. Arevian, and C. Panchev, “Recurrent neural network learning for text routing,” in *Ninth International Conference on Artificial Neural Networks (ICANN99)*, vol. 2, (London), pp. 898–903, IEE, London, 1999.

[116] P. H. Winston, *Artificial Intelligence.* Addison-Wesley, 1984.

[117] J. A. Wise, “The ecological approach to text visualization,” *Journal of the American Society for Information Science*, vol. 50, no. 13, pp. 1224–1233, 1999.

[118] *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, (Espoo, Finland), Helsinki University of Techology, Neural Networks Research Centre, 1997.

[119] J. Zavrel, "Neural navigation interfaces for information retrieval: are they more than an appealing idea?," *Artificial Intelligence Review*, vol. 10, no. 5-6, pp. 477–504, 1996.

[120] J. Zavrel and J. Veenstra, "The language environment and syntactic word-class aquisition," in *Proceedings of the Groningen Assemly on Language Aquisition (GALA95)*, pp. 365–374, 1995.