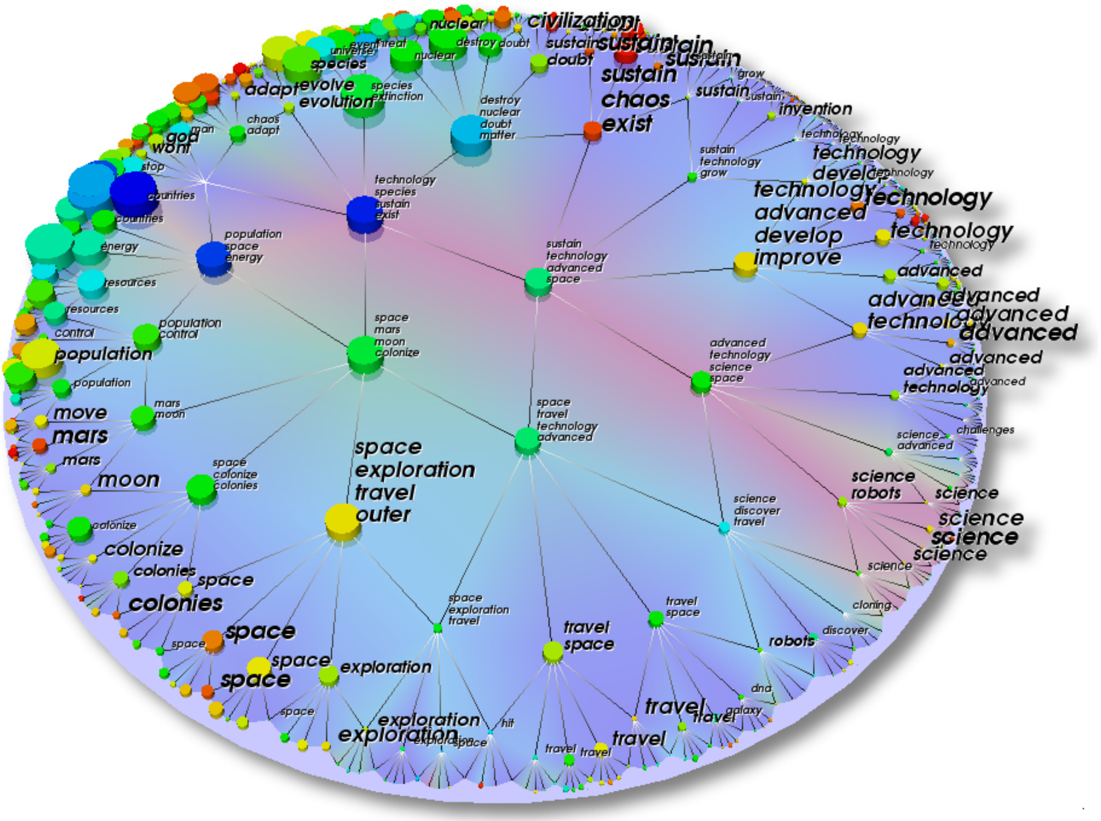
Semantic Visualization with Hyperbolic Self-Organizing Maps

A novel approach for exploring structure in large data sets

Jörg Ontrup



Der Technischen Fakultät der Universität Bielefeld vorgelegt zur Erlangung des akademischen Grades *Doktor der Ingenieurwissenschaften*

Abdruck der genehmigten Dissertation zur Erlangung des akademischen Grades *Doktor der Ingenieurwissenschaften (Dr.-Ing.)*

Der Technischen Fakultät der Universität Bielefeld

- am 06.02.2008 vorgelegt von Jörg Ontrup
- am 06.06.2008 verteidigt und genehmigt

Gutachter:

- Prof. Dr. Helge Ritter, Universität Bielefeld
- Prof. Dr. Samuel Kaski, Helsinki University of Technology

Gedruckt auf alterungsbeständigem Papier nach ISO 9706

To my family

Acknowledgments

This work would not have been possible without the support and help of many people. It was carried out in the Neuroinformatics Group, headed by Prof. Helge Ritter, at the Faculty of Technology, University of Bielefeld.

First of all, I thank Helge not only for his constant support but also for many of his truly inspiring ideas he gave throughout all stages of my work. His strong enthusiasm and optimism had an invaluable impact: Also when I was foraying into the business world and chose an entrepreneurship his encouragement has led to the completion of this work. The Neuroinformatics Group was an excellent and very enjoyable workplace and I thank Olli for his maintenance of the technical infrastructure and for his unbureaucratic way of fixing things. A very special thanks goes to Tim. Not only were our scientific collaborations very fruitful, but in many of our discussions he became a trusted friend with his truly remarkable sense of humor. Likeways I have always enjoyed the company of my former colleagues Axel, Bert, Matthias and Thorsten. Not only they assisted with many useful tips, but also were always ready for a supporting laugh. Special credits go also to Ingo for proofreading the manuscript - any remaining mistakes are truly my own fault. To Nils and Carola go my warm thanks for their indispensable input to the “Hawking” and “aid” project. I am also very much obliged to Uli - as my business partner he had mostly to do without me while I was writing up the thesis - I am very grateful for that. Likeways many thanks go to Andreas, Manfred and Peter at HOTSPLOTS. Thank you all!

On the personal side, my profound thanks go to my parents for all the support they gave to me throughout my life. Finally, I thank my wife Gesa, for all her emotional support, and what is most important, for her love.

Contents

Publications

Parts of this thesis have been published in:

- Ontrup, J. and Ritter, H. (2001a). Hyperbolic self-organizing maps for semantic navigation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 1417–1424.
- Ontrup, J. and Ritter, H. (2001b). Text categorization and semantic browsing with self-organizing maps on non-Euclidean spaces. In *Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 338–349. Springer, LNAI 2168.
- Ontrup, J., Nattkemper, T., Gerstung, O., and Ritter, H. (2003). A mesh term based distance measure for document retrieval and labeling assistance. In *Proceedings of the 25th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society (EMBS)*. Cancun, Mexiko.
- Walter, J., Ontrup, J., Wessling, D., and Ritter, H. (2003). Interactive visualization and navigation in large data collections using the hyperbolic space. In *Proceedings of the Third IEEE International Conference on Data Mining*. IEEE.
- Ontrup, J., Wersing, H., and Ritter, H. (2004). A computational feature binding model of human texture perception. In *Cognitive Processing*, 5(1).
- Ontrup, J. and Ritter, H. (2005a). *Clinical Knowledge Management: Opportunities and Challenges*, chapter Interactive Information Retrieval as a Step Towards Effective Knowledge Management in Healthcare, pages 52–71. Idea Group Publishing.
- Ontrup, J. and Ritter, H. (2005b). A hierarchically growing hyperbolic self-organizing map for rapid structuring of large data sets. In *Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM 05)*. Paris, France.
- Saalbach, A., Ontrup, J., Ritter, H., and Nattkemper, T. W. (2005). Image fusion based on topographic mappings using the hyperbolic space. In *Information Visualization*, 4(4):266–275.
- Wagner, R., Ontrup, J., and Scholz, S. W. (2005). Innovative technologies for clustering, monitoring, and evaluation of up-and-coming topics in business information. In *Japanese-German Symposium on Classification*. Tokyo, Japan.
- Ontrup, J. and Ritter, H. (2006). Large-scale data exploration with the hierarchically growing hyperbolic SOM. In *Neural Networks*, 19(6):751–761.
- Martin, C. and Diaz, N.N. and Ontrup, J. and Nattkemper, T.W. (2008). Hyperbolic SOM-based clustering of DNA fragment features for taxonomic visualization and classification. In *Bioinformatics*.
- Ontrup, J. and Ritter, H. and Scholz, S.W. and Wagner, R. (2008). Detecting, Assessing and Monitoring Relevant Topics in Virtual Information Environments. In *IEEE Transactions on Knowledge and Data Engineering*.

Chapter 1

Introduction

In 1597 the philosopher Francis Bacon has coined the aphorism “knowledge is power”. Today we have access to information sources, literally at the touch of our fingers, Bacon would probably never have dreamed of: For example, Google News indexes more than 10,000 news sources worldwide on a continuous basis, the MEDLINE catalogue contains more than 17 million references to biomedical publications to which every hour 80 new publications are added (on average). These two examples are only a tiny snapshot of the data we can access from almost everywhere anytime. It poses a tough challenge to cope with this tidal wave of information. In order to gain valuable knowledge from this vast amount of data four directions have emerged from the perspective of modern information processing:

First, the discipline of classical information retrieval is mainly concerned with efficient indexing techniques and the discovery of data items meeting a specific information need.

Second, the field of machine learning is concerned with algorithms which enable the computer to learn from large amounts of data. Either the application of classification systems which are able to predict the semantics of documents based on their word statistics, or the ability to detect structural relationships within large collections help the human operator to scan through huge quantities of information.

From a third perspective, data might be transformed into visual form permitting the viewer to look and browse through information in order to obtain a better understanding.

And last, additional layers of semantics might be added to information sources either by the incorporation of hand-crafted ontologies or by the collaborative input of thousands of Internet users.

The main contribution of our work is a novel semantic visualization approach aiming at the integration of all four aforementioned perspectives: Based upon the self-organizing principle we show how the peculiar geometric properties of non-Euclidean hyperbolic space can be exploited to generate hierarchical structurings of large datasets. The benefits of the hyperbolic hierarchically self-organization are twofold: First, the computational complexity offers a speed-up of several orders of magnitude for large datasets. And second, the hyperbolic space offers a natural *focus & context* environment allowing the user to browse through data at arbitrary semantic resolutions. In the following we lay out the organization of the manuscript and the further contributions of our work.

1.1 Organization of the Manuscript

In Chapter 2 we discuss the challenges arising from an ever increasing amount of available information and review some of the key methods which have been proposed to overcome this information overload. We present standard techniques from classical information retrieval to search for information and examine methods for their evaluation. There are several directions from which more recent solutions to information overload emerge: One from a machine learning perspective targeting the computational power of modern machines to uncover hidden data structures. One from a data visualization perspective trying to optimize the presentation of information such that human-computer interaction is most effective. And last, but not least, from a viewpoint which seeks to imprint an additional layer of semantics to the data - either by the application of hand-crafted semantic networks or by the collaborative input of thousands of users through the Web 2.0.

Chapter 3 introduces the hyperbolic self-organizing map (HSOM). We briefly review the concept of the biologically motivated self-organizing map and give a review on the history of Non-Euclidean geometry. By combining both we show how the employment of the hyperbolic plane as the main canvas for laying out the nodes of the HSOM offers a unique *focus & context* approach to self-organizing maps. Based on two artificial datasets and one standard benchmark from classical information retrieval we compare the performance of the HSOM to its standard Euclidean counterpart.

In Chapter 4 we present a hierarchically growing extension to the HSOM, the H^{2}SOM. First, we show how the intrinsically “uniformly hierarchical” structure of the hyperbolic grid can be exploited to drastically reduce the computational complexity of the self-organizing algorithm. Second, we motivate an extension to the standard model of text representation in information retrieval and expand the flat bag-of-words by applying the *WordNet* lexical database in order to construct the hierarchically organized *pyramid-of-words*. In combination with a neural *word sense disambiguation* model we show the benefits of those extensions with respect to classification tasks for the Reuters-21578 benchmark dataset.

In Chapter 5 we lay out the overall software architecture of an interactive text visualization system. It consists of three main modules responsible for text storage, self-organizing text structuring and interactive visualization. We demonstrate how the combination of hyperbolic space, hierarchically growing self-organizing maps and interactive visualization together with standard techniques from classical information retrieval can be used to achieve a novel data display methodology. On the basis of three real world datasets we show how the proposed framework allows for an intuitive analysis and understanding of large amounts of text data.

Chapter 6 discusses the design of two user studies to corroborate the findings from the previous chapters. The first study is addressing the question how effective the hyperbolic *focus & context* interface is with respect to navigation tasks in large data structures. Thirtysix students took part in the study. The results are highly significant and show that for complex navigation tasks users achieve a higher score in less time when they are using the H^{2}SOM framework as compared to a classical tree browser. In the second study five food and health experts evaluated the H^{2}SOM framework in a recommendation system setting to answer the question: Are hierarchical semantic maps able to retrieve knowledge from unstructured text data which is comparable to expert knowledge? The results show that the artificial recommendation system is indeed able to produce helpful results for Internet users.

The concluding Chapter 7 gives a short summary and discusses the achievements of the thesis.

Chapter 2

Background: Information Overload

2.1 Exponentially Growing Information Spaces

Since the invention of movable type printing in the 15th century mankind is observing a steadily increasing pace with which new information becomes available. It was Vannevar Bush in 1945 who was the first to suggest a mechanical solution to cope with the amount of in-flowing data. He recognized that *"The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of square-rigged ships"* (Bush, 1945). Fig. 2.1 shows a sketch[1] of the device which was published by Life Magazine in 1945.

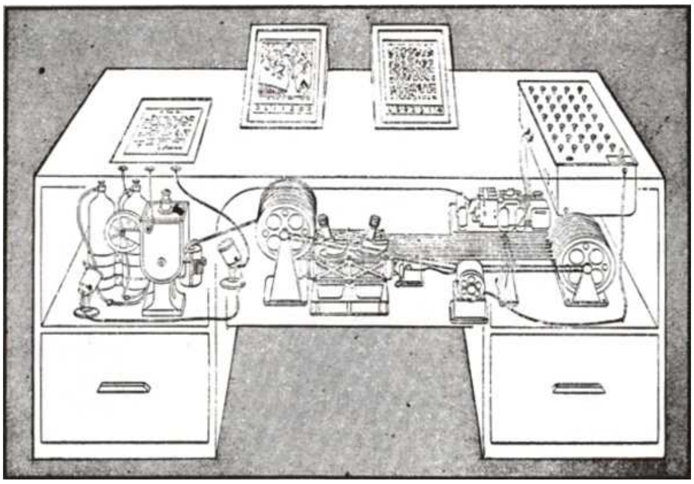


Figure 2.1: The mechanical device proposed by Bush. The original caption read: *"Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by coded numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference."*

Bush considered the capability of an associative indexing the most essential feature of the memex: *"The process of tying two items together is the important thing"*. Because of this statement, the memex has been regarded as the precursor of today's world wide web. Without doubt, the web has eased the access to any kind of information, but the

[1] The image was taken from http://www.kerryr.net/pioneers/gallery/bush.htm at 27th. January 2008 and is believed to be public domain.

sheer plethora of news portals, Internet forums, literature databases and other sources has not made it necessarily more *useful*. Since 1965 when Gordon E. More - a co-founder of Intel Corp. - has formulated "Moore's Law" the capabilities of electronic devices has doubled approximately every two years. This lead to an exponential increase of data storage capacities which is easily taken up by the information produced today.

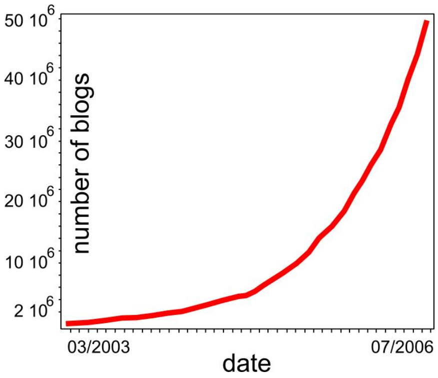


Figure 2.2: Development of the blogosphere: The number of weblogs has doubled in size about every six months.

We illustrate the consequences by two examples. First, consider MEDLINE, a literature database indexing about 5,000 selected publications covering bio-medicine and health. The amount of citations as of the end of 2007 totals to some 17 million references. On average there are about 80 scientific publications added every hour. Clearly it is out of the scope of any scientist to digest that kind of information load.

Second, consider the so-called *blogosphere* - a term describing the world of weblogs and their interconnections. Blogs have been identified as having a similar influence to their consumers as traditional mainstream media. The number of links to popular weblogs is similar to that of well established online portals of media companies such as the "BBC" or "USA Today". Technorati[1] has been tracking the blogosphere since November 2002. Its growth figure is shown in Fig. 2.2, conveying a doubling in approximately every six months. Consequently, a single individual is not able to track all topics and trends set by the blogs and affecting an increasing amount of people. For the average user this is not harmful. However, in the context of business management the situation is different: The ability to effectively scan large amounts of data in order to quickly identify key information is one of the most crucial skills for business managers today (Decker et al., 2005). Without that ability executives might pursue wrong strategic plans resulting in total business failure. We therefore strongly believe that the ability to deal effectively with incoming information streams is an important prerequisite to be successful in our modern information environment.

2.1.1 Cognitive Aspects

During its evolution mankind has adapted in a fascinating way to the challenges brought upon us by nature. All of our senses are highly optimized to survive in the environment we are living in. Our brain is able to organize the tremendous amount of information supplied by these senses in such a way that the world is not perceived as a chaotic stream of impressions, but as a well structured set of entities. In everyday life we do not have to think about how to keep balance while walking, how to interpret high resolution images from our retina, or how to separate continuously changing air pressures into words uttered by different people. The attempt to mimic these abilities by means of computational power has lead to a just beginning understanding how complex the achievements of our brain really are.

The problem is: In the era of the information society we are living in now, we directly experience that our brain is not optimized to scan, interpret and act upon the level of data which is hitting us by the tidal wave of information overload. Our brain is simply not able to integrate the incoming stream into a set of coherent entities as it does with its sensory

[1] http://technorati.com

input. Herbert A. Simon has formulated a key phrase describing the resulting cognitive consequences: *"What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the overabundance of information sources that might consume it"* (Simon, 1971). In other words - in order to consume and effectively digest all the information which is reaching us, we need more and more of our high level brain capacities to cope with it. It cannot simply be done concurrently. From a psychological point of view the ever increasing load of information can even lead to a mental disorder, Lewis (1996) coined "information fatigue" raising stress levels and manifesting itself in a number of ways, such as "inabilities to make decisions", "irritability and anger", or "loss of energy". In their literature review Edmunds and Morris (2000) observed that in recent years the problem has become more widely recognized and experienced.

We therefore seek help from instruments we are building to bear with the difficulties arising. From a computational point of view there should be no fundamental difference between high level data such as video streams from cameras, or message postings in Internet forums. So, if we can build computer vision systems which are able to achieve at least some of the human vision capabilities, why should we not be able to build computerized systems able to edit information into a set of coherent entities more easily accessible for us. In the following sections we review some of the key methods which have been proposed as contributing building blocks to achieve this goal.

2.2 Searching for Information

The first ingredient for handling the flood of information is a well structured way of searching within information spaces. We here briefly lay out the fundamental principles of classical information retrieval which has culminated in today's web search engines such as Google.

2.2.1 Classical Information Retrieval

Classical information retrieval (IR) has its roots in librarianship. It is mainly concerned with finding information for a specific *information need* usually defined by a user query. Broadly speaking, an IR system compares the user query with all stored documents and returns a set of matching documents sorted by their relevance to the query. Generally, IR systems are considered to be one of the following model types:

- Set-theoretic models representing documents as sets of index terms.
- Algebraic models representing documents as vectors.
- Probabilistic models assuming probabilistic hypotheses for the retrieval.

For a detailed overview the reader is referred to the standard literature, e.g. Manning and Schütze (1999); Baeza-Yates and Ribeiro-Neto (1999).

The Bag-of-Words Model

The one name most closely connected to the field of information retrieval is that of Gerard Salton. His pioneering work paved the way for a formal mathematical approach on information retrieval. In one of his most influential papers he drew a 3D coordinate system with a set of orthogonal basis vectors depicted as index terms. Documents were plotted as dots in the resulting space and the first paragraph read *"Consider a document space..."* (Salton et al., 1975). Since then the so-called *vector space model* has become the premier model used in IR. In the vector space model each document d is represented as a vector $\mathbf{f}_d$ given by

$$\mathbf{f}_d = (w_1^d, \cdots, w_N^d), \tag{2.1}$$

where w_1^d to w_N^d are weights for each of the N *terms* occurring in all documents stored in the retrieval system. The definition of a "term" depends on the actual implementation of the indexing system. Common choices are tokens resulting from a simple white-space parsing or more elaborate linguistic procedures such as word stemming (Manning and Schütze, 1999). Since the vectorial representation loses the word order within documents, the model is commonly referred to as the *bag-of-words* model. Similarities of documents can be easily computed by means of the *cosine angle* between their two vectors

$$sim(d_1, d_2) = \frac{\langle \mathbf{f}_{d_1}, \mathbf{f}_{d_2} \rangle}{|\mathbf{f}_{d_1}|\ |\mathbf{f}_{d_2}|}, \tag{2.2}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product, and $|\cdot|$ the norm of the document vectors. Consequently, the distance between two documents d_1 and d_2 is given by

$$dist(d_1, d_2) = 1 - sim(d_1, d_2). \tag{2.3}$$

There has been extensive work on the choice of optimal weighting schemes for the weights in Eq. 2.1. Salton and Buckley (1988) have proposed several variants, and an overview is given by e.g. Manning and Schütze (1999). Throughout this theses we have applied a standard *term frequency* × *inverse document frequency* (tf-idf) weighting scheme given by

$$w_t^d = N_t^d \, log \left(\frac{N_D}{N_t^D} \right), \tag{2.4}$$

where N_t^d is the number of times term t occurs in document d, N_D is the total number of documents in the database and N_t^D is the number of times term t appears in the whole collection of all documents. The tf-idf weighting scheme is motivated by two assumptions:

1. The more often a term occurs in a document, the more important it is for the document. This is reflected by the tf-part of Eq. 2.4.

2. Terms which occur in every document have no discriminative power. I.e., the word "the" is not of much use if searching for information. Hence, the idf-part of Eq. 2.4.

2.2.2 Retrieval Performance Evaluation

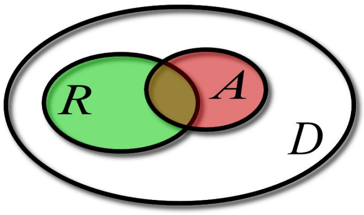


Figure 2.3: Illustrating retrieval performance evaluation.

For any computational system it is of importance to be able to measure its successfulness. In IR the predominant performance measures are *precision* and *recall*. Consider the situation depicted in Fig. 2.3 on the left: We have a collection of documents D, a user query q, a set of documents R relevant to the query q, and an answer set A which is returned by the IR system.

Given the above definition we can compute the following measures:

- The *precision* $Prec$ is the fraction of retrieved documents relevant to the user query, i.e.,

$$Prec = \frac{|R \cap A|}{|A|} \tag{2.5}$$

- The *recall* Rec is the fraction of relevant documents which have been retrieved, i.e.,

$$Rec = \frac{|R \cap A|}{|R|} \tag{2.6}$$

Note, that the above measures of precision and recall assumes that the whole answer set A is evaluated. They compute the precision a system is able to achieve at its maximum recall. From a practical point of view, it is more convenient to sort the result set according to the relevance of its elements to the query. The ranked list is then presented to the user who is starting to examine the results from the top of the list. We therefore compute a *precision-recall curve* based on the ranked result list.

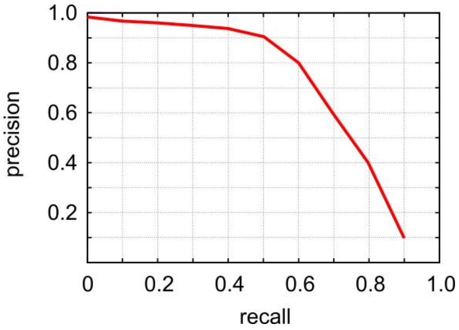


Figure 2.4: Typical precision-recall curve: The ranked list starts at a high precision and as the user retrieves more and more documents the precision declines.

Fig. 2.4 shows a typical example of a precision-recall curve. If the first document in the ranked list is relevant to the query, the precision equals to one while the recall is low. As the user proceeds to examine the ranked list, more and more documents which are relevant are retrieved. Hence, the recall level is increasing, however usually at the cost of precision: More and more documents turn up which are not relevant to the original query.

Note, that the ranking function used to sort the retrieval set has a large influence on the quality of the IR system. One of the most famous ranking methods is the *PageRank* algorithm of Brin and Page (1998) - the founders of Google. The high quality of search results achieved by *PageRank* and a very simple user interface are believed to be two key elements contributing to the success of Google. For a more thorough discussion of retrieval performance evaluation the interested reader is referred to the standard literature, e.g., Baeza-Yates and Ribeiro-Neto (1999); Manning and Schütze (1999).

2.3 Exploring Information Spaces

Classical IR as briefly reviewed above is able to provide helpful tools to dig out relevant items from a large amount of data. However, it is only helpful when pursuing a specific information need. If confronted with large volumes of information, IR is not able to provide condensed overviews nor is it able to detect hidden structures or connections within the data. In order to meet this challenge we need to turn to other methods.

2.3.1 Machine Learning Approaches

The term *machine learning* describes the field in computer science which is concerned with computational methods enabling computer systems to learn from data. It is out of the scope of this thesis to review a whole discipline. We therefore concentrate on central aspects and premier examples of the application of machine learning approaches to the field of unstructured text data - also called *text mining*.

Text Categorization

In text categorization - sometimes also called text classification - the task is to assign categories or topics to documents based on their content. For this setting usually *supervised* algorithms are considered. In a first step the machine is trained with a set of labelled documents from which an internal representation is learned - as illustrated in Fig. 2.5.

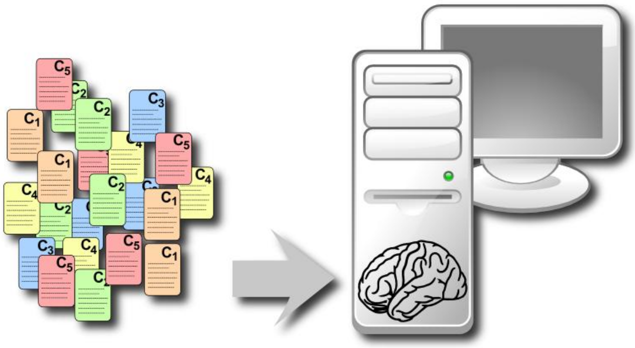


Figure 2.5: Step one: During a training phase the machine learns an internal representation from a set of labelled documents.

After the system has learned how the document content affects its semantics, the machine can be used in an application setting in which new documents are presented to the system. It uses the "knowledge" obtained from the training phase and can assign category labels to previously unknown documents - as illustrated in Fig. 2.6.

Typical application settings are routing tasks in customer support which forward user questions directly to the responsible employee, alert systems which flag news messages depending on user interest, or email classification systems sorting incoming mail directly

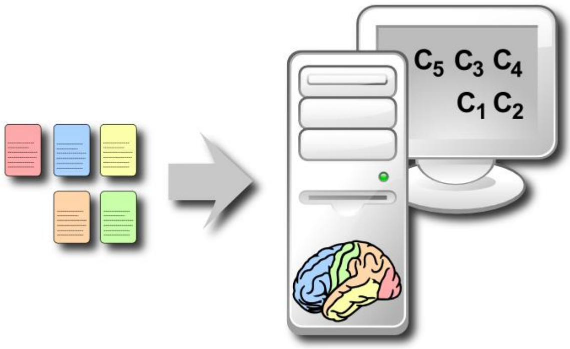


Figure 2.6: Step two: After the system has learned how the content of the documents determine their categories, it can be used to assign category labels to new previously unknown documents.

into their designated folders.

Previous work on machine learning algorithms for text categorization tasks is as multi-faceted as the discipline of machine learning itself. Important directions are the works of Joachims (1998) who applied *support vector machines* (SVM) for classification of very high dimensional bag-of-words feature vectors, Yang (1999) who evaluated statistical approaches such as the *naive Bayes classifier*, or Sebastiani et al. (2000) using boosting techniques to increase classification performances. Craven and Kumlien (1999) and Marcotte et al. (2001) applied statistical classifiers to extract knowledge from biomedical publications. For an extended review on machine learning approaches on the task of text categorization the interested reader is referred to Sebastiani (2002).

Text Clustering

In text clustering the task is to partition a collection of documents into subsets or clusters. Most of the corresponding machine learning algorithms operate in an unsupervised way, i.e., they do not require any labelling of the data in advance. Fig. 2.7 illustrates the main principles: A collection of (unlabelled) documents is fed into the system and the system learns an internal representation usually on the basis of some distance measure between the documents. The clustering system then assigns each document to one of several clusters.

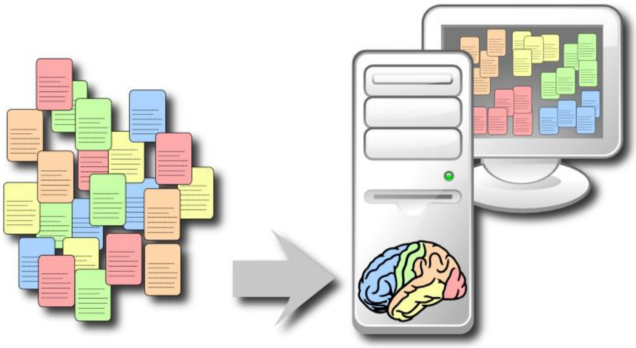


Figure 2.7: Text Clustering: In an unsupervised manner the machine learns from the input data and partitions a collection of documents into disjoint clusters.

In the context of information overload, clustering approaches allow the user to group the incoming data into coherent groups - similar to the perceptual grouping the brain does concurrently for visual sensory information (Ontrup et al., 2004). This might significantly reduce the overload since the examination of only a few representatives for each detected cluster allows to assess the topic distribution within a whole collection of documents.

Generally, there exist two ways of clustering. The first organizes the data in a hierarchical manner - either by partitioning the data in a top down approach (*divisive clustering*) or by merging items bottom up (*agglomerative clustering*). The second variant, partitional clustering, directly assigns the data items to one element of a set of disjoint clusters.

In the context of text processing, the self-organizing map (SOM) - a member of the second group and introduced by Kohonen (1982) more than 20 years ago - combines several advantages: Its computational complexity allows its application to large real world datasets as the WEBSOM project impressively has demonstrated (Kaski et al., 1998b; Kohonen et al., 2000; Lagus et al., 2004). Furthermore, the generated map might be used for visualization purposes as well as for classification tasks (Merkl, 1997; Ontrup and Ritter, 2001b; Rauber et al., 2002; Hung et al., 2004).

2.3.2 Information Visualization

The field of *information visualization* offers a further perspective to handle the information overload. Generally, information visualization is concerned with the transformation of data into a visual form which permits the user - or better: the viewer - to look at and browse through information in order to obtain a better understanding. A somewhat beaten phrase expounds its ambition more clearly: *"A picture says more than a thousand words"*. Two of the reasons why good visualization systems are capable to transport a tremendous amount of information are:

1. From an information theoretic point of view, the human eye and the visual cortex provide the highest-bandwidth "information highway" into the cognitive center of our brain. First, this is supported by neurological findings (Borst and Theunissen, 1999). Second, in technical systems the channels for encoding image data generally need most of the bandwidth, indicating that the visual elements carry the most information which cannot be compressed any more.

2. A visual presentation might be aesthetically appealing. Thus, making the user like to inspect and explore the data.

Map Displays

A common visualization metaphor many people are used to is that of a *map*. For centuries humans have used maps to depict and convey geographical information. More recently the map metaphor has also been used as a general data canvas for drawing information patches on it. In combination with artificially created mappings it becomes a powerful tool to transport large amounts of information.

The group of methods which achieves a mapping of data samples from a high-dimensional feature space to a low-dimensional mapping canvas is usually referred to as *multidimensional scaling* (MDS) (Sammon, Jr., 1969; Cox and Cox, 1994).

As already mentioned above, the self-organizing map (SOM) does not only achieve a clustering of data points, but is also a prominent example of how low-dimensional mappings can be created algorithmically. In the context of text visualization an early milestone is the WEBSOM project (Honkela et al., 1996). Fig. 2.8 shows a screenshot from a SOM trained with 12088 articles taken from the comp.ai-neural-nets usenet newsgroup[1]. The image shows an overview of the map with an imposed color

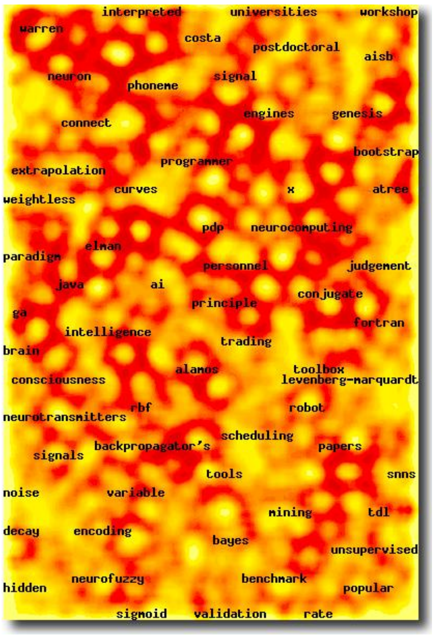


Figure 2.8: WEBSOM (1997): 12088 articles from a neural network newsgroup from June 1995 to March 1997 (http://websom.hut.fi/websom/comp.ai.neural-nets-new/html/root.html 02/01/08).

[1] Note, that this and the other images below were taken from the corresponding project's webpage and are believed to be in the public domain. URL and time of the snapshots are given in the corresponding figure caption.

scale visualizing node distances. For details on the construction of the map see Kohonen et al. (2000).

More recently Skupin (2004) has produced aesthetically very pleasing maps motivated by geographic metaphors. Fig. 2.9 shows a mapping of scientific abstracts to a canvas painted with colors borrowed from geography. Other SOM-based approaches have been published by e.g., Merkl (1998); Rauber and Merkl (2001); Rauber et al. (2002) or Hung et al. (2004).

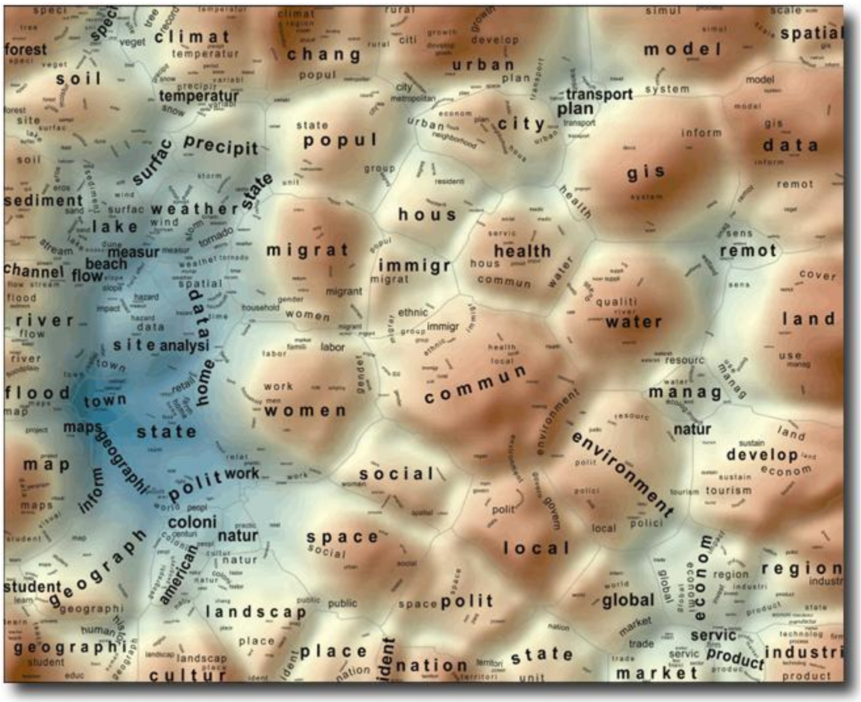
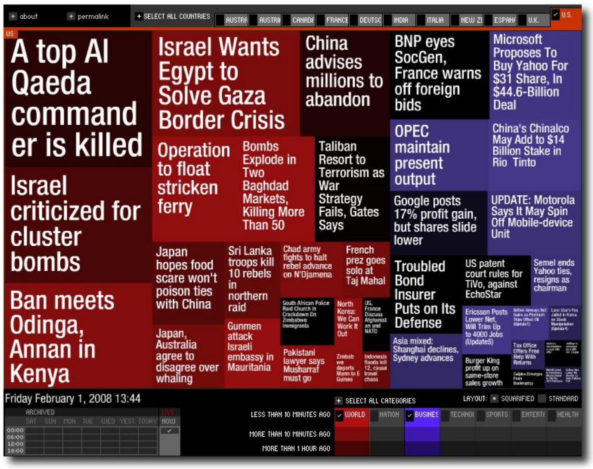


Figure 2.9: Cartographic approach (2004): A SOM-based visualization of abstracts submitted to the Annual Meetings of the AAG held between 1993 and 2002. (http://geography.sdsu.edu/People/Pages/skupin/ 02/01/08).

A different approach for creating two dimensional mappings is the *Treemap* algorithm by Shneiderman (1992); Bederson et al. (2002). Fig. 2.10 shows an application of the treemap layout to display news headlines. The data is aggregated by Google News which crawls major news sites at short time intervals and clusters similar messages. Since many messages are written by journalists on the basis of news agency postings they share a common word statistics. This allows for a relatively easy clustering which identifies groups of news items describing the same incident. The treemap then allocates for each cluster an area on the canvas which is proportional to the number of items within the cluster. The resulting



A top Al Qaeda commander is killed

Israel Wants Egypt to Solve Gaza Border Crisis

China advises millions to abandon

BNP eyes SocGen, France warns off foreign bids

Microsoft Proposes To Buy Yahoo For $31 Share, In $44.6-Billion Deal

Israel criticized for cluster bombs

Operation to float stricken ferry

Bombs Explode in Two Baghdad Markets, Killing More Than 50

Taliban Resort to Terrorism as War Strategy Fails, Gates Says

OPEC maintain present output

China's Chinalco May Add to $14 Billion Stake in Rio Tinto

Google posts 17% profit gain, but shares slide lower

UPDATE: Motorola Says It May Spin Off Mobile-device Unit

Ban meets Odinga, Annan in Kenya

Japan hopes food scare won't poison ties with China

Sri Lanka troops kill 10 rebels in northern raid

Chad army fights to halt rebel advance on N'Djamena

French prez goes solo at Taj Mahal

Troubled Bond Insurer Puts on Its Defense

US patent court rules for TiVo, against EchoStar

Semel ends Yahoo ties, resigns as chairman

South African Police Raid Church In Crackdown On Zimbabwe Immigrants

North Korea: We Can Work It Out

US, France Discuss Afghanistan and NATO

Ericsson Posts Lower Net, Will Trim Up to 4000 Jobs (Update5)

Japan, Australia agree to disagree over whaling

Gunmen attack Israeli embassy in Mauritania

Pakistani lawyer says Musharraf must go

Zimbab we deports Mann to E Guinea

Indonesia floods kill 12, cause travel chaos

Asia mixed: Shanghai declines, Sydney advances

Tax Office Offers Free Help With Returns

Burger King profit up on same-store sales growth

Friday February 1, 2008 13:44

Figure 2.10: Newsmap (2004): A treemap showing news headlines compiled by the *Google News* news aggregator (http://marumushi.com/apps/newsmap/newsmap.cfm 02/01/08).

map offers an intuitive view on the current news situation. In addition, the occupied map area gives a very quick impression on the importance the journalists ascribe to each news.

The above examples show that maps are able to transport information very effectively. However, for very large datasets we face the challenge to either reduce the information in order to "squeeze" it on a limited map space, or to provide zooming facilities allowing for a closer inspection of certain areas of interest on demand.

Focus & Context Displays

By using zooming techniques for the visualization of very large datasets, users might lose the context of surrounding data when focusing on specific aspects. Therefore, the notion of *focus & context* or *fisheye* displays has been proposed to offer both simultaneously (Furnas, 1986; Sarkar and Brown, 1994). Lamping and Rao (1994) used a projection of the hyperbolic plane $I\!H^2$ (which is reviewed in larger detail in Chapter 3) for the embedding of large hierarchical structures. Their hyperbolic tree browser allows a focus & context endowed navigation within very large hierarchical data sets. Schaffer et al. (1998) and Pirolli et al. (2003) have shown that focus & context navigation can significantly accelerate "information foraging". A screenshot of a 3D hyperbolic viewer for browsing web content is shown in Fig. 2.11.

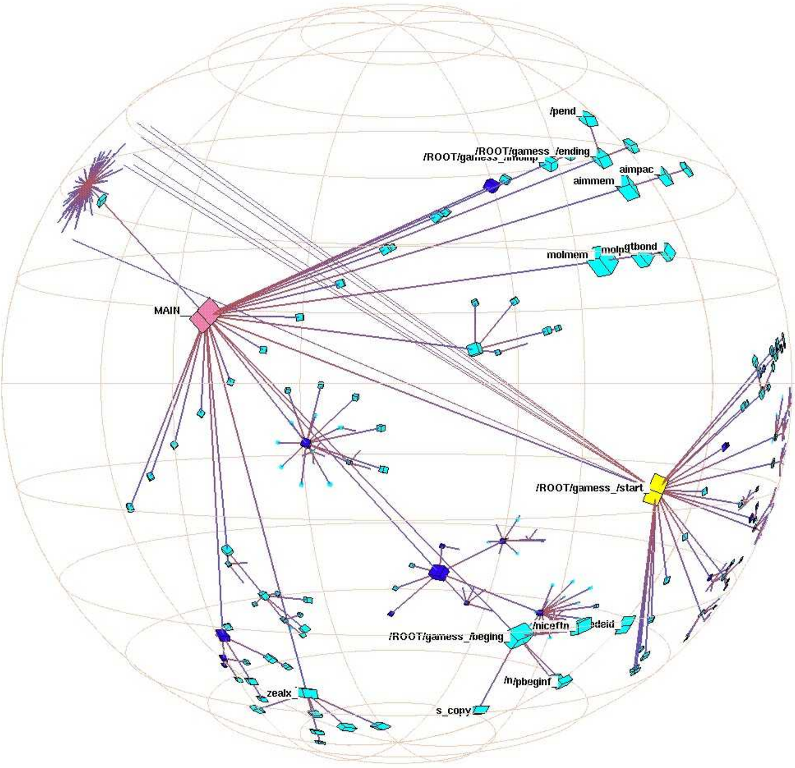


Figure 2.11: A 3D hyperbolic viewer showing the tree-like graph of a web site (http://graphics.stanford.edu/papers/h3draw/ 02/01/08).

Note, that the majority of focus & context enabled viewers expects the data to be organized in a hierarchical manner. They are not able to exploit the peculiar properties of hyperbolic space to allow for a navigation within unstructured data sets such as document databases.

2.4 Semantic Networks

A third perspective on remedies for information overload is the notion of *semantic networks*. It was Tim Berners-Lee who was the first to formulate the vision of a *Semantic Web* (Berners-Lee et al., 2001): *"The Semantic Web will enable machines to* ***comprehend*** *semantic documents and data, not human speech and writings"*. According to his original idea, a variety of formal descriptions of concepts and relationships from a given knowledge domain should enable software agents to gain meaningful access to web contents. Technically, the semantics should be expressed by *resource description frameworks* (RDF), standardized formats such as XML, or a *web ontology language* (OWL). However, in a recent contribution his co-authors stated that *"this simple idea [...] remains largely unrealized"* (Shadbolt et al., 2006).

The extra layer of machine-understandable content necessary for the fulfillment of the Semantic Web poses a barrier which up to date has not been crossed. The attempt to force web content into a strict framework understandable by "classical" information processing has not yet produced substantial results to reduce the information overload.

2.4.1 The Web 2.0

A different, but successful, approach to generate semantic content has emerged from the *Web 2.0* - a term coined by Tim O'Reilly at the "Web 2.0 Conference 2004". It describes a trend in web development where the web is understood as a business platform for services giving users control over their own personal data. In the following, many services and social-networking sites have attracted large user bases. A common concept in many Web 2.0 applications is the idea of *tagging*: Users are able to tag data such as photos, pieces of music, or favored movies with keywords. Giving the large statistics of the substantial user bases this results in a *collaborative tagging* or *social classification* creating an additional layer of information offering a semantic access to web contents. A method to visualize user-generated tags with their corresponding weights is the notion of a *tag cloud* as illustrated in Fig. 2.12.

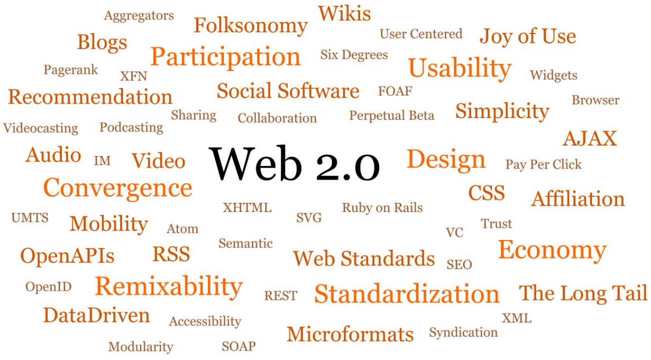


Figure 2.12: A tag cloud with terms describing concepts related to the Web 2.0. The image was taken from Wikipedia (http://en.wikipedia.org/wiki/Tag_cloud 02/02/08).

Recently, Grahl et al. (2007) have shown that by clustering of collaborative tags, hierarchies can be constructed which allow for a conceptual retrieval of data within social bookmarking systems, thus providing a semantic access to large information sources.

2.4.2 WordNet: a Semantic Lexicon of the English Language

In contrast to the semantic networks generated by millions of users in the context of the Web 2.0, the WordNet lexical database was handcrafted by a small amount of highly specialized experts (Fellbaum, 2001). WordNet can be considered as a semantic lexicon of the English language which groups synonymous words into sets called *synsets*. Semantic connections between synsets describe various relationships of their corresponding words depending on their morphology. For nouns the most important relation is the *hypernym/hyponym* relation: A thing X is a hypernym of another thing Y, if Y is a kind of thing X, i.e., a computer is a kind of machine, thus the word "machine" is a hypernym of "computer". The same relationship holds for verbs, i.e. x is a hypernym of y, if y is a way of doing x. As an example, "walking" is a way of "travelling". These connections within the lexical database resemble a valuable knowledge base. For example, the look-up of the word "terrier" yields the following hierarchical hypernym tree:

```
terrier
  => hunting dog
    => dog, domestic dog, Canis familiaris
      => canine, canid
        => carnivore
          => placental, placental mammal, ...
            => mammal, mammalian
              => vertebrate, craniate
                => chordate
                  => animal, animate being, beast, ...
                    => organism, being
                      => living thing, animate thing
                        => object, physical object
                          => physical entity
```

We shall see later in Section 4.2.1, how this knowledge database might used to extend the bag-of-words model from classical information retrieval to semantically enriched *pyramid-of-words*.

2.5 Summary

In this chapter we have given a short introduction to the problems evoked by information overload. We have discussed four perspectives which offer a remedy:

- **Information Retrieval:** Efficient indexing techniques allow for a direct search of information relevant to a specific user query.
- **Machine Learning:** Either the application of classification systems which are able to predict the semantics of documents based on their word statistics, or the ability to detect clusters or other structures within large collections help the human operator to scan through huge quantities of information.
- **Information Visualization:** By transforming document data into visual presentations, the abilities of the human visual system might be utilized to recognize hidden data structures.
- **Semantic Networks:** Either by the application of hand-crafted semantic networks or by the collaborative input of thousands of users, additional semantic layers might be generated.

Ideally, a system aiming at the alleviation of information load addresses all four perspectives. In the next chapters we present the hyperbolic self-organizing map (HSOM) and its hierarchically growing extension. Throughout the work we add several semantic extensions to the hyperbolic models and propose an approach which addresses all of the four aspects within a single framework.

Chapter 3

Semantic Maps in Hyperbolic Space

3.1 Topographic Feature Maps

As discussed in the previous chapter, the employment of a map metaphor is a popular option to present data in an easy "digestible" way to the human observer. In order to answer the question, how these maps might be generated algorithmically, we turn to biologically inspired models.

Our brain is constantly receiving a plethora of high dimensional input patterns from our senses. It is somehow organizing this data in such a way that we do not drown in the information stream but instead perceive a coherent and well structured world. In their classic work, Hubel and Wiesel (1959) addressed the question of how the visual cortex of the brain is morphologically organized. They analyzed the sensitivity of neurons to differently orientated gratings and found that when moving gradually through the cortex, cells are tuned to continuously varying gratings. That is, cells neighbored on the cortex generally respond to input stimuli neighbored in the input space. Thus, the orientation tuning over the surface of the cortex forms a kind of map where similar input stimuli are close together. Due to this reason, these maps are commonly referred to as *Topographic Feature Maps*. An illustrative example for the case of input stimuli from our sense of touch is the homunculus[1] which was first systematically mapped using electrical stimuli by Penfield and Boldrey (1937) as shown in Fig. 3.1.

The neuron's response properties which form a mapping from the high dimensional input space to the low dimensional space of the cortex are not genetically hardwired into the brain, but rather shaped by experience (Blakemore and Cooper, 1970; Dalva and Katz, 1994). For the orientation selectivity of cells in striate cortex a first computational model for such a self-organizing process was proposed by von der Malsburg (1973). But it was Kohonen (1982) who formulated the generalized method of the Self-Organizing Map (SOM), also commonly known as the *Kohonen Map*.

3.1.1 Kohonen's Algorithm

The main objective of Kohonen's SOM algorithm (Kohonen, 1982, 2001) is to achieve a mapping from an arbitrary N-dimensional input space $\mathcal{X} \subset \mathbb{R}^N$ to a set of formal neurons $\mathcal{A}$ placed within a low-dimensional map space $\mathcal{M} \subset \mathbb{R}^M$ (typically with a dimension of $M = 1$ or $M = 2$). In order to reflect the biological archetype, this mapping should *(i)* be obtained during a learning phase in a completely *self-organizing* fashion; and *(ii)* the resulting map

[1] Latin for "little man"

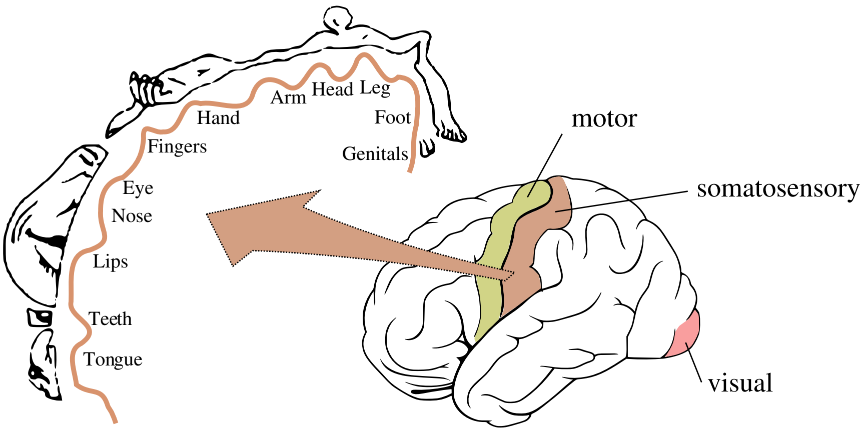


Figure 3.1: On the right hand side, the primary motor, primary somatosensory and primary visual cortex of the brain are highlighted. On the left, the magnification of the primary somatosensory cortex shows that the neurons are topographically ordered; thus, representing an "image" from the body, the so called *homunculus*.

should be *topologically ordered*, i.e., it should translate data similarities from the input space $\mathcal{X}$ into spatial relations on the map space $\mathcal{M}$. Algorithmically, the SOM is build in the following fashion:

1. The neurons A of the network are placed along the vertices of a regular lattice $\mathcal{L}$ in the map space $\mathcal{M}$. Most commonly used are a 2D rectangular grid as Kohonen (1982) proposed in his original work and as shown in Fig. 3.2, or a 2D hexagonal grid (Kohonen, 2001).

2. To each neuron $a \in \mathcal{A}$, a *reference vector* $\mathbf{w}_a \in I\!R^N$ is attached, projecting into the input data space $\mathcal{X}$. It might be convenient to initialize the reference vectors on a 2D map along the two principal components of the training data distribution. By doing so, the map is roughly ordered before the learning phase begins. Thus, allowing for a reduction of the initial learning phase in which the coarse global structure of the map is determined. However, the computation of the principal components are computationally expensive, especially for large data sets. If such initialization of the reference vectors is prohibitive, an initialization with random values is also feasible.

3. During a learning phase, the distribution of the reference vectors $\mathbf{w}_a$ is iteratively adapted by a sequence of training vectors $\mathbf{x}_t \in \mathcal{X}$: After finding the so-called "best-match" neuron a^*, i.e. the node which has its prototype vector $\mathbf{w}_a$ closest to the given input $\mathbf{x}$,

$$a^* = \operatorname{argmin}_a \; \|\mathbf{w}_a - \mathbf{x}\| \tag{3.1}$$

all reference vectors are updated by the adaptation rule

$$\Delta\mathbf{w}_a = \epsilon(t)\; h(a,a^*)\; (\mathbf{x} - \mathbf{w}_a), \tag{3.2}$$

where $h(a,a^*)$ is a neighborhood function centered at the "winner" node a^* and decaying with increasing distance $d_{\mathcal{M}}(a,a^*)$ in the map space spawned by the neuron lattice $\mathcal{L}$.

Common examples of the neighborhood function h are the bell shaped Gaussian with

$$h(a,a^*) = \exp\left(-\frac{d^2_{\mathcal{M}}(a,a^*)}{2\sigma^2(t)}\right) \tag{3.3}$$

and a simpler so-called *bubble* function defined as

$$h(a, a^*) = \begin{cases} 1 & \text{if } d_{\mathcal{M}}(a, a^*) < \sigma(t) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where σ is a given radius specifying the neighborhood around the winner node in which adaptation takes places.

Depending on the chosen lattice structure $\mathcal{L}$ the distance $d_{\mathcal{M}}$ in the map space might either be given by the Euclidean distance of the corresponding node coordinates in $I\!R^M$ or by the length of the connecting path on the lattice. For the computation of the distance in the input space (Eq. 3.1) usually the Euclidean metric is used, although other options such as the dot product are possible as well. Note, that the adaption rule (Eq. 3.2) has to reflect the metric used in Eq. 3.1, i.e., for the dot product a renormalization step to unit length is mandatory.

During the course of learning, the width $\sigma(t)$ of the neighborhood function and the learning step size $\epsilon(t)$ are continuously decreased in order to allow more and more specialization and fine tuning of the then increasingly weakly coupled neurons. A common choice for $\sigma(t)$ and $\epsilon(t)$ is an exponentially decreasing function with respect to t.

Despite the relatively large number of parameters for the SOM, the algorithm has been found to be robust against variations for different parameter settings (Kiang and Kumar, 2001). For a more thorough discussion on the aspects of different grid topologies, initialization schemes, neighborhood functions and learning rates, the interested reader is referred to the book of Kohonen (2001).

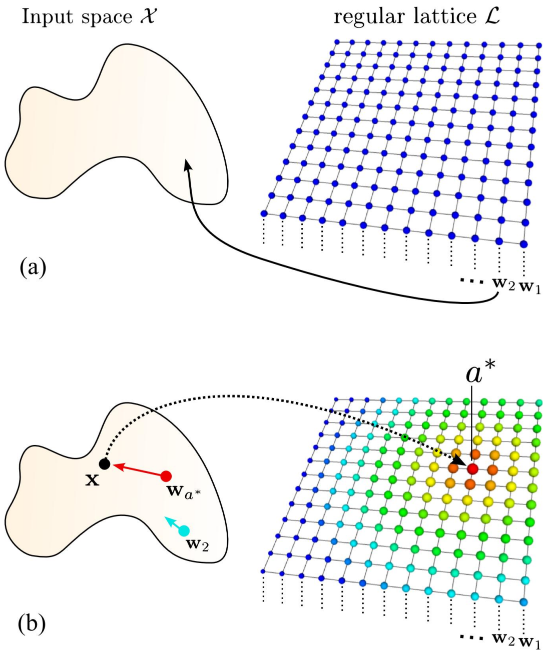

Input space $\mathcal{X}$
regular lattice $\mathcal{L}$
(a)
$\cdots\ \mathbf{w}_2\,\mathbf{w}_1$
a^*
$\mathbf{x}$
$\mathbf{w}_{a^*}$
$\mathbf{w}_2$
(b)
$\cdots\ \mathbf{w}_2\,\mathbf{w}_1$

Figure 3.2: Illustration of the SOM algorithm. (a) shows a rectangular lattice structure $\mathcal{L}$ of formal neurons. The feature vectors $\mathbf{w}_a$ attached to each neuron project back into the input space $\mathcal{X}$. In (b) an input stimulus $\mathbf{x}$ selects the winner neuron a^*. Depending on the neighborhood function and the distance from a^* on $\mathcal{L}$ all reference vectors are adapted towards $\mathbf{x}$.

3.1.2 Previous Work on Self-Organizing Maps

The number of publications on the SOM is remarkable. Since its first presentation in the 80s (Kohonen, 1982), there has been on average one new publication per day during the last 25 years, resulting in more than 6000 published papers up to now. A corresponding bibliography with a thematic and keyword index has appeared in two journal articles (Kaski et al., 1998a; Oja et al., 2003). The diversity in applications of the SOM is extremely widespread. But until recently, the overwhelming majority of these works had one common ground: virtually all approaches use the *flat Euclidean space* as the geometrical substrate for the creation of a regular lattice structure $\mathcal{L}$ of formal neurons. By far the most approaches utilize the 2D Euclidean plane as their chief canvas for the generated mappings (such as shown Fig. 3.2) mainly because for the ease of visualizing the resulting mappings on a flat computer screen, although there has also been work on higher dimensional Euclidean lattice structures (Bradburn, 1989; Ritter et al., 1992; Kiviluoto, 1998; Gaskett and Cheng, 2003).

3.2 Non-Euclidean Geometry

Though the choice of a Euclidean map space $\mathcal{M}$ might seem very natural to us, it is by no means a mandatory choice. In 1999 for the first time Ritter (1999) has suggested to use non-Euclidean spaces in conjunction with the SOM. Others have adopted this idea and have presented spherical SOMs utilizing the 2D surface of a sphere for the SOM's neuron lattice (Sangole and Knopf, 2003; Wu and Takatsuka, 2005). In the following section we briefly describe the historical development of hyperbolic space and point out its major geometrical benefits as a motivation for using the hyperbolic plane $\mathbb{H}^2$ as an alternative map space for the SOM algorithm.

3.2.1 Historical Development of Hyperbolic Space

Most of our spatiotemporal thinking is deeply rooted in the world of Euclidean geometry as it is still taught in schools today. Basically all of our school-knowledge on geometry can be found in Euclid's *Elements*, a series of books written around 300 BC. It is considered to be one of the most widely read books, "and is second only to the Bible in number of editions published"[1]. Euclid's basic assumption consist of five "common notions" (called axioms today) and the following five postulates (taken from Coxeter (1957)):

1. *A straight line may be drawn from any one point to any other point.*
2. *A finite straight line may be produced to any length in a straight line.*
3. *A circle may be described with any center at any distance from that center.*
4. *All right angles are equal.*
5. *If a straight line meets two other straight lines, so as to make the two interior angles on one side of it together less than two right angles, the other straight lines will meet if produced on that side on which the angles are less than two right angles.*

It was the fifth postulate that was long eyed suspiciously. It is not of the same elegant simplicity as the first four postulates and does contain no explicit statement of where the meeting point of the two lines could be. In essence that point could be infinitely far away, which bothered geometers 750 years after Euclid so much, that they tried very hard to get rid of this (in their view) objectionable postulate and to deduce it as a proposition. To all avail, they did not succeed and the best they could do, was to replace it with an equivalent, more simple assumption. The most common example is probably Playfair's postulate (Playfair, 1861):

5a. *Through any point, there is at most one line parallel to a given line.*

[1] http://en.wikipedia.org/wiki/Euclid's_Elements

Gauss, Lobachevsky and Bolyai (ca. 1830)

It was in the beginning of the 19th century when independently from each other the mathematicians Gauss, Lobachevsky and Bolyai denied Euclid's 5th Postulate and discovered the notion of what we today call *hyperbolic geometry*[1]. It was Lobachevsky (1829) who was the first to publish this idea of a non-Euclidean geometry - in Russian in the "Kazan Messenger". Naturally, this newspaper article did not reach far into the mathematical community and was rejected for publication when submitted to the "St. Petersburg Academy of Sciences". At about the same time, J. Bolyai discovered the same results which were published as an appendix to his father's book (Bolyai, 1832).

More than 10 years after the original discoveries of Lobachevsky his French article (Lobachevsky, 1837) and his German book (Lobachevsky, 1840) produced more attention to disseminate the thoughts on non-Euclidean geometry. Gauss was impressed by Lobachevsky's work and it came apparent, that Gauss himself has been working on the issue of non-Euclidean geometry, but was reluctant to publish any of his findings. Only in a personal communication he wrote in 1824: *"The assumption that the sum of the three angles in a triangle is less than* 180° *leads to a curious geometry, quite different from ours, but thoroughly consistent, which I have developed to my entire satisfaction."* Despite the attention of a mathematician such as Gauss, non-Euclidean geometry was still a mere obscurity and not accepted within the mathematical world at that time. As it turned out, a new mathematical perspective on geometry was necessary to pave the way for the non-Euclidean world.

Riemannian Geometry (1854)

It was the work of Bernhard Riemann whose "Habilitationsvortrag" on the 10th of June in 1854 founded the field of *Riemannian geometry*. It is not in the scope of this thesis to provide a complete introduction to the subject. Instead we briefly lay out the fundamental ideas and refer the interested reader to the standard literature, e.g. Morgan (1993), do Carmo (1976) or Chavel (1994). Riemannian geometry is a part of *differential geometry* dealing with the study of geometry using calculus. Riemann's revolutionary idea was to concentrate on the properties of space itself, rather than the objects and their attributes within that space. Consider the following example: When thinking of the curvature of an object's surface, e.g. the surface of a coffee mug, we see it as an extrinsic property, i.e. from the way its surface bends in the 3D space around it. Riemann shifted this extrinsic perspective to an intrinsic point of view. He took the view of an ant traveling along the surface of the coffee mug not being able to perceive the space from outside. Nevertheless with Riemann's help the ant would be able to deduct all geometric properties from local observations:

To describe a surface S in three-dimensional space, we define the convex region Ω in a parameter space u, v as shown in Fig. 3.3. The surface S is then given by the *mapping functions*

$$\begin{aligned} x &= f_x(u,v) \\ y &= f_y(u,v) \\ z &= f_z(u,v), \end{aligned} \tag{3.5}$$

with $u, v \in \Omega,\ x, y, z$ being the Cartesian coordinates in 3-space and f_x, f_y, f_z the continuous and differentiable functions. The parameters u, v are called *curvilinear coordinates* on

[1] The term *hyperbolic geometry* was first mentioned by Klein (1871).

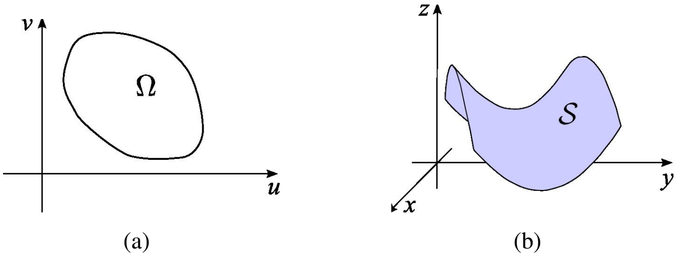


Figure 3.3: (a) The curvilinear coordinates u and v from the convex parameter space Ω represent the surface $\mathcal{S}$ in 3-space (b).

the surface $\mathcal{S}$. For example, the mapping functions

$$\begin{aligned} f_x(\phi,\theta) &= r\sin(\phi)\ \cos(\theta) \\ f_y(\phi,\theta) &= r\sin(\phi)\ \cos(\theta) \\ f_z(\phi,\theta) &= r\cos(\phi) \end{aligned} \tag{3.6}$$

with $\Omega : 0 < \phi < \pi,\ 0 < \theta < 2\pi$, $r = \text{const.} > 0$ and the two spherical polar coordinates ϕ and θ describe the surface of a sphere with radius r.

Under the assumption that the mapping functions in Eq. 3.5 are differentiable, two points (u, v) and $(u + \Delta u, v + \Delta v)$ that are close together in Ω get mapped to two points (x, y, z) and $(x + \Delta x, y + \Delta y, z + \Delta z)$ that are close together on the surface $\mathcal{S}$. The Euclidean distance between them is given by

$$\Delta s = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}, \tag{3.7}$$

with

$$\Delta x = \frac{\partial x}{\partial u}\Delta u + \frac{\partial x}{\partial v}\Delta v,\ \ \Delta y = \frac{\partial y}{\partial u}\Delta u + \frac{\partial y}{\partial v}\Delta v \text{ and } \Delta z = \frac{\partial z}{\partial u}\Delta u + \frac{\partial z}{\partial v}\Delta v. \tag{3.8}$$

By defining the 2×2 matrix G

$$G = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \tag{3.9}$$

with

$$\begin{aligned} g_{11} &= \left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial y}{\partial u}\right)^2 + \left(\frac{\partial z}{\partial u}\right)^2 \\ g_{12} = g_{21} &= \frac{\partial x}{\partial u}\frac{\partial x}{\partial v} + \frac{\partial y}{\partial u}\frac{\partial y}{\partial v} + \frac{\partial z}{\partial u}\frac{\partial z}{\partial v} \\ g_{22} &= \left(\frac{\partial x}{\partial v}\right)^2 + \left(\frac{\partial y}{\partial v}\right)^2 + \left(\frac{\partial z}{\partial v}\right)^2 \end{aligned} \tag{3.10}$$

we can rewrite Eq. 3.7 quite elegantly as

$$ds^2 = \mathbf{v}^T G \mathbf{v}, \quad \text{with } \mathbf{v} = \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}. \tag{3.11}$$

The expression ds is called the *line element*, and the matrix G the *Metric Tensor* of the surface $\mathcal{S}$ with respect to the coordinates u and v. G depends on the particular choice of u and v on the surface $\mathcal{S}$. For example, if $\mathcal{S}$ is the 2D plane and u, v are the Cartesian coordinates x, y, the line element and metric tensor are given by

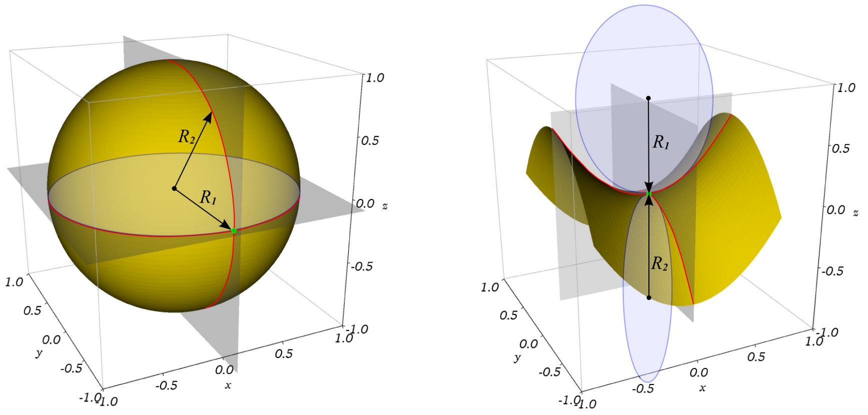
$$ds^2 = dx^2 + dy^2, \quad \text{and} \quad G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{3.12}$$

respectively. The metric tensor G can be roughly interpreted as a function which tells us how to compute the distance between any two points in a given space. It describes the geometry of a space in a *local* way. By means of integrating over the line element ds we are able to obtain geometric properties such a *length*, *angle*, or *area* (do Carmo, 1976; Chavel, 1994). Thus, the entire intrinsic geometry of a space $\mathcal{S}$ can be obtained from its metric tensor G.

An important property of space (independent of the used coordinate system) is the *Gaussian curvature*. In case of a regular surface in $I\!R^3$ we can extrinsically compute the curvature at point p as

$$K(p) = \frac{1}{R_1 R_2}, \tag{3.13}$$

where R_1 and R_2 are the radii of the *principal curvatures* k_1 and k_2. Roughly speaking, the principal curvatures describe the maximal and minimal curvature of the surface, their directions are perpendicular to each other as indicated by the intersecting planes in Fig. 3.4.



(a) (b)

Figure 3.4: By determining the radii R_1, R_2 of the extrinsic principal curvatures (indicated as the red curves) we can compute the Gaussian curvature at a point p (green point) by $K(p) = 1/R_1R_2$. (a) In case of the sphere, both radii are 1 and point into the same direction, resulting in a curvature of $K = 1$. (b) In case of the saddle, the radii point into opposite directions. Hence, the curvature at p is $K(p) = -1$.

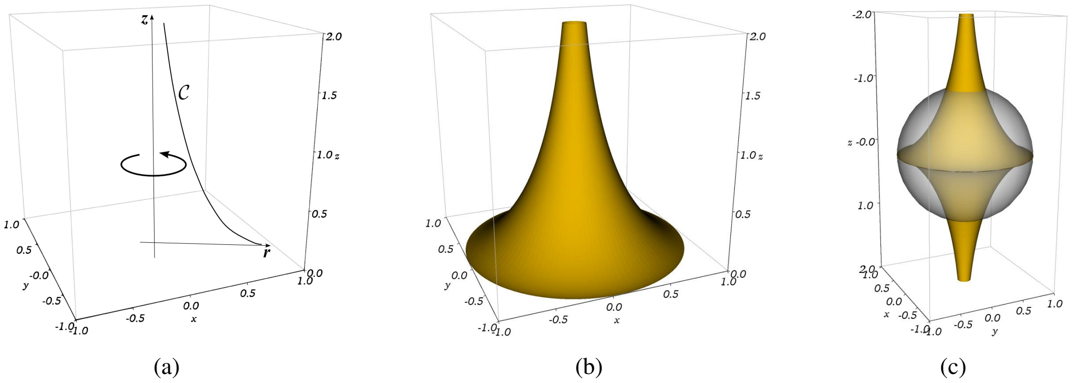
An intrinsic method to compute the curvature can be obtain via the application of the *Gauss-Bonnet theorem* (do Carmo, 1976). Imagine an ant tied to the point p with a short thread of length r. By running around p, the ant can measure the circumference $C(r)$ of the corresponding area. The Gaussian curvature is then given by:

$$K(p) = \frac{3}{\pi} \lim_{r \to 0} \frac{2\pi r - C(r)}{r^3} \tag{3.14}$$

Spaces with constant curvature, i.e. $K(p) = \text{const.}\ \forall p$, are called *homogeneous*: geometric figures do not change their properties when moved within such a space.

Beltrami's Pseudosphere (1868)

Up until 1868 there was still no mathematical proof for the consistency of Lobachevsky's description of hyperbolic geometry. Thus, mathematicians were reluctant to accept a non-Euclidean beside the well established Euclidean geometry. The breakthrough for the acceptance of hyperbolic geometry was the publication "Essay on the interpretation of non-Euclidean geometry" from Beltrami (1868). He produced the first 3D Euclidean model of a 2D hyperbolic space which he called the *pseudosphere*. Thus, he made use of the familiar Euclidean space to "visualize" hyperbolic space. The pseudosphere is the surface obtained by revolving the curve of the *tractrix*[1] around its asymptote as indicated by Fig. 3.5.



(a) (b) (c)

Figure 3.5: (a) The curve $\mathcal{C}$ of the tractrix in parametric form is given by $r = 1/\cosh t$, $z = t - \tanh t(0 \leq t \leq \infty)$. (b) By revolving the curve around the $z-$axis, the surface of the *pseudosphere* is obtained. In (c) a sphere is superimposed on the pseudosphere. Both share the same volume and surface area, and both surfaces have a constant curvature with magnitude 1. They differ only in the sign of their curvature: $K = +1$ for the sphere, $K = -1$ for the pseudosphere.

Beltrami's pseudosphere does not cover the complete 2D hyperbolic space, but by embedding a part of it into 3D Euclidean space, he was able to apply the tools of Riemannian geometry. He showed that the pseudosphere has constant negative curvature and he was able to prove that if Lobachevsky's geometry leads to a contradiction, Euclidean geometry is contradictory as well. Thus, he proved that consistency of Euclidean geometry directly implies consistency of hyperbolic geometry and finally paved the way for the acceptance of Lobachevsky's findings almost 40 years later.

3.2.2 Models of Hyperbolic Space

Shortly after Beltrami has found the pseudosphere as a Euclidean model of hyperbolic space, a whole set of alternative models were proposed. In the following we briefly discuss the *Minkowski-*, *Klein-Beltrami-* and the *Poincaré*-model of the hyperbolic plane $\mathbb{H}^2$.

Hyperboloid or Minkowski Model

From a geometrical perspective it is the most convenient to start with an embedding of hyperbolic space in Minkowski 3-space. The corresponding model goes back to Weierstrass (Killing, 1880) and Poincaré (1881), who both - independently from each other - used a hyperboloid as a model of the hyperbolic plane $\mathbb{H}^2$. We might utilize a *Minkowski metric*

[1] also called *equitangential* curve, because the length of a tangent from its point of contact to the asymptote is constant

(Sommerfeld, 1909; Jansen, 1909) with its line element and metric tensor given by

$$ds^2 = dx^2 + dy^2 - dz^2, \quad \text{and} \quad G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}. \tag{3.15}$$

By using the mapping functions

$$\begin{aligned} f_x(r,\theta) &= \sinh(r)\ \cos(\theta) \\ f_y(r,\theta) &= \sinh(r)\ \sin(\theta) \\ f_z(r,\theta) &= \cosh(r) \end{aligned} \tag{3.16}$$

we can construct an *isometric* embedding of the hyperbolic plane $I\!H^2$, where r and θ are the polar coordinates on $I\!H^2$. Under this embedding the hyperbolic plane appears as the revolution of the curve $z^2 = x^2 + y^2 + 1$ about the $z-$axis as shown in Fig. 3.6.

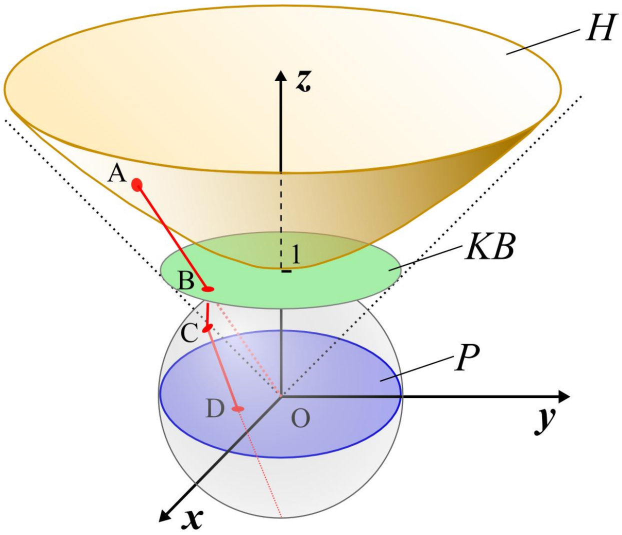


Figure 3.6: Three different models of the hyperbolic plane $I\!H^2$: The hyperboloid H embedded in Euclidean 3-space, the projection to the Klein-Beltrami model KB and the Poincaré Disk P.

Klein-Beltrami Model

The *Klein-Beltrami* model (Klein, 1871) was historically the first complete model of hyperbolic space and originally given by the mapping functions

$$\begin{aligned} f_x(r,\theta) &= \tanh(r)\ \cos(\theta) \\ f_y(r,\theta) &= \tanh(r)\ \sin(\theta), \end{aligned} \tag{3.17}$$

which map the polar coordinates r and θ of the infinite hyperbolic plane $I\!H^2$ to the finite open unit disk $\mathbb{D} = \{z \in \mathbb{C}, |z| < 1\}$ in the complex plane $\mathbb{C}$. It can also be obtained from the Minkowski model by means of projective geometry as indicated in Fig. 3.6 ($A \mapsto B$), i.e., by projecting points from the hyperboloid H onto the plane $z = 1$ along rays passing through the origin O.

Poincaré Disk Model

Similar to the Klein-Beltrami model, the Poincaré Disk (Poincaré, 1881) with its mapping functions

$$\begin{aligned} f_x(r,\theta) &= \tanh(\frac{r}{2})\,\cos(\theta) \\ f_y(r,\theta) &= \tanh(\frac{r}{2})\,\sin(\theta), \end{aligned} \tag{3.18}$$

projects the hyperbolic plane to the open unit disk $\mathbb{D}$ (Ramsay and Richtmyer, 1995). However, in contrast to the former, the latter is achieving a *conformal* mapping, i.e., angles are persevered. From Eq. 3.11 and the mapping functions 3.18 follows for the line segment in the Poincaré Disk in cartesian and polar coordinates

$$ds^2 = 4\frac{dx^2+dy^2}{(1-x^2-y^2)^2} \quad \text{and} \quad ds^2 = 4\frac{dr^2+r^2d\theta^2}{(1-r^2)^2} \tag{3.19}$$

Geometrically, the Poincaré Disk can be constructed as follows (cf. Fig. 3.6): First, a perpendicular projection of the Klein-Beltrami model onto the northern hemisphere of the unit sphere centered at the origin maps point B to C. Second, a stereographic projection of the northern hemisphere onto the unit circle in the ground plane $z=0$ maps point C to D in the Poincaré Disk. It turns out that the Poincaré model of $\mathbb{H}^2$ has a number of pleasant properties for visualization purposes:

- It maps the infinite large area of $\mathbb{H}^2$ entirely into the Euclidean unit disk.
- The projection is conformal, i.e. preserving angles. Consequently, shapes painted in $\mathbb{H}^2$ are not deformed when projected to the Poincaré Disk. However, distances are strongly distorted.
- The non-isometric projection exhibits a strong *fish-eye* effect: The origin of $\mathbb{H}^2$ - corresponding to the fish-eye fovea - is mapped almost faithfully, while distant regions become exponentially squeezed.

Embedding of the Hyperbolic Plane in 3-space

Naturally, there exists no isometric embedding of $\mathbb{H}^2$ into $\mathbb{R}^2$, since a projection of the negatively curved space into flat space introduces distortions in either length, area or angle. Hilbert (1901) also proved that $\mathbb{H}^2$ can not be isometrically embedded in its entirety in 3D Euclidean space. However, a locally isometric embedding into $\mathbb{R}^3$ is possible: we obtain a "wrinkled" structure, which resembles a saddle at every point of the surface. Sometimes, nature approximated the growth behavior of a hyperbolic surface, e.g. in some corals that need to maximize their contact area with the surrounding water that carries vital nutrients. In Fig. 3.7 it can be seen, that this is leading to structures resembling a 3-dimensional local embedding (of a patch) of the hyperbolic plane remarkably well. Note, that such a corrugated structure is also found in the human cerebral cortex which is comparatively thin (about 2-4 mm), but if laid out flat, covers about 2,500 cm^2.

3.2.3 Navigation in Hyperbolic Space

When visualizing the hyperbolic plane with the Poincaré model, only the center of $\mathbb{H}^2$ is faithfully displayed. Due to the strong squashing nature of the $\tanh$-function in Eq. 3.18,

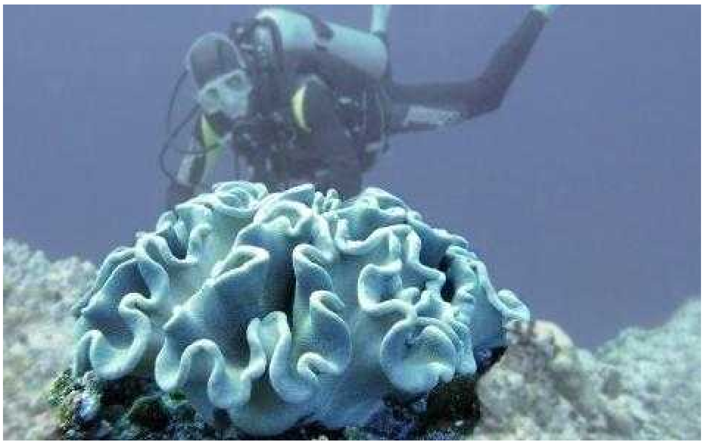


Figure 3.7: A local embedding of $I\!H^2$ in $I\!R^3$ would look very similar to such a leather-coral for which nature found a solution to maximize its contact area in order to absorb vital nutrients from the surrounding water. (Photograph by courtesy of Heinz Toperczer.)

distant regions farther away from the origin become exponentially "squeezed". In order to make full use of $I\!H^2$ in a SOM visualization framework we therefore seek for a method to "navigate" within $I\!H^2$ in such a way that we can move distant regions of the map into the focus of attention. In the following section we describe, how the so-called *Möbius transformations* are utilized to achieve that goal.

Möbius Transformation

The group of *Möbius transformations* Möb$(\mathbb{D})$ is an automorphism on the unit disk $\mathbb{D}$ given by

$$M(z) = \frac{az+b}{\bar{b}z+\bar{a}}, \quad a,b \in \mathbb{C}, \quad z \in \mathbb{D}. \tag{3.20}$$

Therefore, $M(z)$ takes an element of the unit disk $\mathbb{D}$ and maps it back onto $\mathbb{D}$. It is a bijective, homomorphic function, and thus describes the group of isometries of the Poincaré Disk (Anderson, 2001). We might rewrite Eq. 3.20 as

$$M_{c,\varphi}(z) = e^{i\varphi}\frac{z-c}{1-\bar{c}z}, \quad c \in \mathbb{C}, \quad |c| < 1 \tag{3.21}$$

(Strubecker, 1969). For $c = 0$, Eq. 3.21 describes a pure rotation of $\mathbb{D}$ around the angle φ. For $\varphi = 0$, Eq. 3.21 becomes a pure translation where the origin of $\mathbb{D}$ gets mapped to $-c$, and c becoming the new center of $\mathbb{D}$. Thus, the Möbius transformations allow a continuous translation of the focus to any other part of the infinite hyperbolic plane. An illustrative example how this mechanism is used to navigate within $I\!H^2$ is given in Fig. 3.13 of Section 3.3.1.

3.2.4 Geometric Properties of Hyperbolic Space

Distances in Hyperbolic Space

In order to compute the (hyperbolic) distance $\delta(z_1, z_2)$ of any two points $z_1, z_2 \in \mathbb{D}$ in the Poincaré Disk, we are taking advantage of the Möbius transformation as given by Eq. 3.21. Consider Fig. 3.8: The distance between the two points z_1 and z_2 is given by integration of Eq. 3.19 along the path of the arc between z_1 and z_2 (given by the circle meeting the border of $\mathbb{D}$ at right angles). To ease the computation, we might apply the isometric Möbius transformation consisting of a pure translation $M_{z_1,0}$ and a pure rotation $M_{0,\varphi}$ which maps the point z_1 to z_0 and z_2 to z_r, i.e. when identifying c of Eq. 3.21 with z_1:

$$M_{0,\varphi}(M_{z_1,0}(z_2)) = z_r = e^{i\varphi}\frac{z_2 - z_1}{1 - \bar{z}_1 z_2} \tag{3.22}$$

Since $z_r = r + 0i$ is real valued, it follows that

$$r = \left|\frac{z_2 - z_1}{1 - \bar{z}_1 z_2}\right|. \tag{3.23}$$

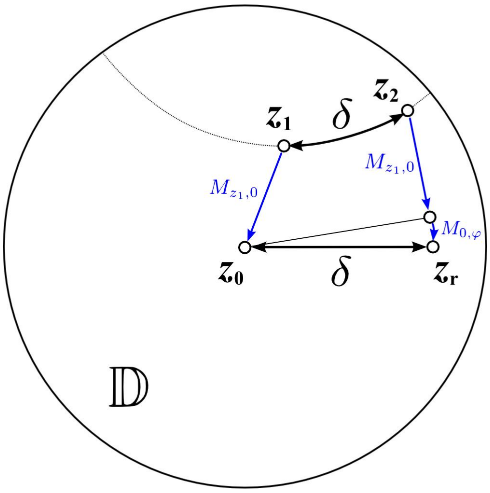


Figure 3.8: Hyperbolic distance.

We now only need to integrate the path along the x-axis and therefore have with Eq. 3.19 for the hyperbolic distance between z_1 and z_2:

$$\delta(z_1, z_2) = \int_0^r \frac{2}{1 - x^2}\, dx = 2\,\text{arctanh}(r) = 2\,\text{arctanh}\left(\left|\frac{z_2 - z_1}{1 - \bar{z}_1 z_2}\right|\right). \tag{3.24}$$

Area in Hyperbolic Space

Lets consider a hyperbolic circle with (hyperbolic) radius ρ as indicated by the yellow patch in Fig. 3.9(c). To compute its hyperbolic circumference $C(\rho)$ we integrate along its perimeter and with Eq. 3.19 and Eq. 3.24 we get

$$C(\rho) = \int_0^{2\pi} \frac{2r}{1 - r^2}\, d\theta = \frac{4\pi r}{1 - r^2} = 4\pi\frac{\tanh(\rho/2)}{1 - \tanh^2(\rho/2)} = 2\pi\sinh(\rho). \tag{3.25}$$

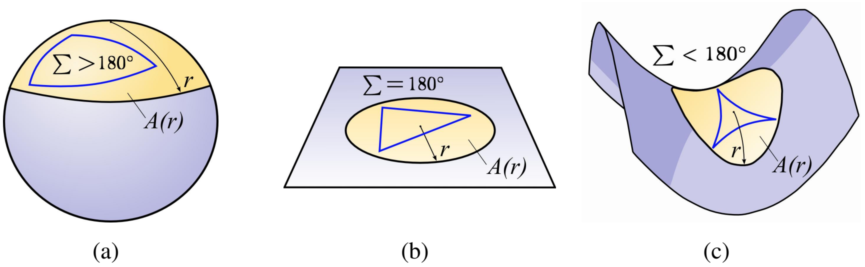


Figure 3.9: Growth of area in differently curved spaces. (a) shows a sphere with $K = +1$, (b) a flat plane with $K = 0$ and (c) a saddle with local curvature $K = -1$. They possess disparate geometric properties, such as different sums of angles in triangles, and different growth behaviors for the area of a circular patch.

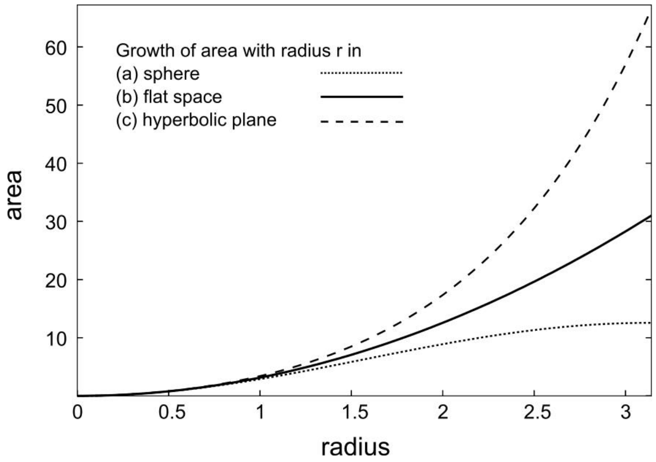


Figure 3.10: Growth for the area of a circle with radius r in differently curved spaces.

Similarly, we obtain for the hyperbolic area $A(\rho)$ of the circle

$$\begin{aligned} A(\rho) = \int_0^r \int_0^{2\pi} \frac{4r}{(1-r^2)^2}\, dr\, d\theta \quad &= \quad 4\pi \int_0^r \frac{2r}{(1-r^2)^2}\, dr \\ &= \quad 4\pi \frac{r^2}{1-r^2} = 4\pi \sinh^2(\frac{\rho}{2}). \end{aligned} \tag{3.26}$$

As we see from Eq. 3.25 and Eq. 3.26, the growth of the circumference and area for a circle of radius ρ bears two remarkable asymptotic properties, also shown in Fig. 3.10:

(i) For a small radius ($r < 1$), the space “looks flat”, since $C(r) \approx 2\pi r$ and $A(r) \approx \pi r^2$.

(ii) For larger r, both C and A grow asymptotically exponentially with the radius. Consequently, the area of a hyperbolic patch grows much faster than in the Euclidean case. In other words, the hyperbolic plane offers “much more space” for the embedding of objects than the Euclidean plane does.

3.3 Self-Organizing Maps in Hyperbolic Space

The asymptotically exponential scaling behavior discussed above makes hyperbolic spaces extremely useful for the accommodation of large hierarchical structures. This was also observed by Lamping and Rao (1994) who constructed the hyperbolic tree viewer allowing to browse through large tree-like graph data (cf. Section 2.3.2).

In the context of the self-organizing map, it offers us the opportunity to escape from the rather limited area the nodes of the SOM can occupy in the standard flat Euclidean case: Especially for larger networks the exponential scaling behavior of the hyperbolic space provides the nodes of the network with more freedom to map smaller portions of the original data space with less distortions as compared to Euclidean space.

3.3.1 Generation of a Regular Lattice Structure

As outlined in Sec. 3.1.1 we need some sort of regular lattice structure $\mathcal{L}$ to which we can attach the prototype vectors of the artificial neurons $\mathcal{A}$ to. Mathematically, a *regular tessellation* is the periodic tiling of the plane (or its generalization to higher dimensions) with congruent regular polygons (or polytopes in higher dimensions), i.e. equally shaped polygons where all sides of the polygon have the same length. For the Euclidean plane, there exist only three regular tessellations as depicted in Fig. 3.11 (Williams, 1979). For $I\!H^2$ there exists an infinite number of tessellations with congruent polygons (Magnus, 1974), of which a tiling with triangles is geometrically the most simple.

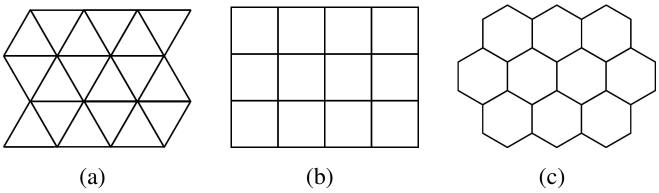


Figure 3.11: The three possible regular tessellations of the Euclidean plane with (a) equilateral triangles, (b) squares and (c) hexagons.

For the creation of a regular lattice structure based on equilateral triangles, we start with a center node placed at the origin of $I\!H^2$ - as indicated by the slightly larger blue node in Fig. 3.12(a); and we choose the tessellation order n describing the number of triangles meeting at each vertex. In order to do so, we have to observe two conditions for the angle α of the equilateral triangles:

$$\alpha < \frac{180^\circ}{3}, \tag{3.27}$$

because the negatively curved space induces a sum of angles of less than 180° for any triangle in $I\!H^2$ (cf. Fig. 3.9(c)). Second, the n triangles around a vertex must cover a full circle, i.e.

$$\alpha = \frac{360^\circ}{n}. \tag{3.28}$$

When combining the two conditions in Eq. 3.27 and Eq. 3.28 we see that we need at least $n = 7$ triangles for our tessellation scheme. Note, that there exists no upper bound for the number n of neighbors a node can have.

By choosing the angle of the equilateral triangles we also fix the triangle's side lengths. From the hyperbolic law of cosines (Anderson, 2001)

$$\cosh(a) = \cosh(b)\cosh(c) - \sinh(c)\sinh(b)\cos(\alpha), \tag{3.29}$$

where a, b, c are the hyperbolic lengths of a triangle's sides, and α is the interior angle opposite the side a, it follows for an equilateral triangle with side length a

$$\cos(\alpha) = \frac{\cosh(a)}{\cosh(a)+1}, \quad \text{or} \tag{3.30}$$

$$a = \operatorname{arccosh}\left(\frac{\cos(\alpha)}{1-\cos(\alpha)}\right) \tag{3.31}$$

When a vertex of the equilateral triangle is centered at the origin of $\mathbb{D}$ (cf. Fig. 3.12(a)) we obtain with Eq. 3.24 for its corresponding side length l in the Poincaré Disk

$$l = \tanh\left(\frac{1}{2}\operatorname{arccosh}\left(\frac{\cos(\alpha)}{1-\cos(\alpha)}\right)\right). \tag{3.32}$$

By placing a second node at distance l from the center of $\mathbb{D}$ as indicated by the upper green node in Fig. 3.12(a) we construct the first side of our "seed crystal" triangle. The iterative application of the Möbius transformation $M_{0,\varphi}$ with $\varphi = \cos(\alpha) + i\sin(\alpha)$ then generates the node coordinates of the first "ring" of the HSOM as indicated in Fig. 3.12(a).

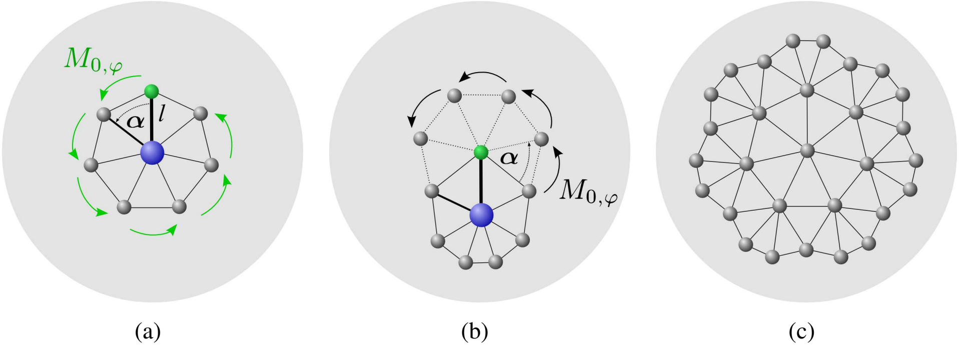


Figure 3.12: The tessellation process to build a self-organizing map in hyperbolic space. By iteratively applying a set of Möbius transformations we add ring by ring of equilateral triangles to the network.

In a next step, each node of the perimeter of the already existing network is moved to the center of $\mathbb{D}$ with an appropriate Möbius transformation - as exemplary shown for the green node in Fig. 3.12(b). With $n-3$ rotations applied to one of its "old" neighbors we can then compute the node coordinates of its "new" neighbors (cf. with the arrows in Fig. 3.12(b)). By repeating the process of moving a border node to the center and rotating its neighbor we can iteratively add ring by ring to the network of the hyperbolic self-organizing map. The resulting image for a network with $n = 7$ neighbors and $R = 2$ rings can be seen in Fig. 3.12(c). In the following we use the notation n^R to denote a hyperbolic grid of regular triangles with n neighbors and R rings. The number of nodes of a lattice constructed in this way grows very rapidly (asymptotically exponentially) with the chosen number of rings R. We therefore obtain HSOMs with large numbers of nodes quite easily. For instance, a 8^5-grid already contains 2281 nodes.

Focus and Context in the Hyperbolic SOM

When displaying the HSOM's lattice structure, the strong distortion of distances in the Poincaré Disk results in a heavy compressed grid at the periphery of $\mathbb{D}$. This effect can be observed in Fig. 3.13(a), where the outer triangles appear significantly smaller, although they all share the same (hyperbolic) size. By using the Möbius transformations as outlined in Section 3.2.3, we can translate the fovea in $\mathbb{D}$ such that distant regions are mapped to the origin with its higher resolution. An illustrative example how this applies to the HSOM setup is given in Fig. 3.13: It shows a navigation sequence where the fovea was moved from the center of $\mathbb{D}$ towards the highlighted region of interest at the 2 o'clock position. Note, that from the left to the right details in the target area get increasingly magnified, as the colored region occupies more and more display space. In contrast to standard zoom operations, the current surrounding context is not clipped, but remains visible gradually compressed at the periphery of the field of view. Since all operations are continuous, the fovea can be positioned in a smooth and natural way and we therefore obtain a human computer interface which inherits the merits of a *focus and context* aware system as described in Chapter 2, Section 2.3.

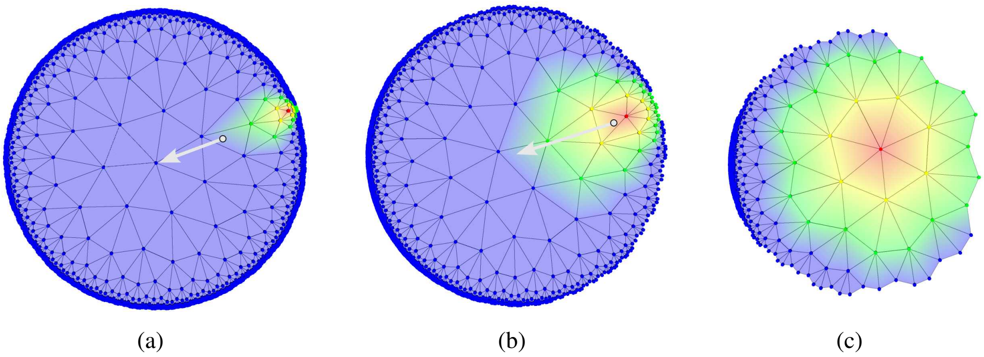


Figure 3.13: Navigation snapshots showing the isometric Möbius transformation acting on the regular tessellation of the HSOM. The three images were acquired while moving the focus from the center of the map to the highlighted region at the outer perimeter. The arrows indicate the translation of a selected point c to the origin of $\mathbb{D}$ as described by Eq. 3.21. Note the "fish-eye" effect: All triangles are congruent, but appear smaller as further they are away from the central focus.

3.3.2 Training Procedure

The HSOM is trained in the standard fashion as described in Section 3.1.1. The only, but striking difference is the adjustment of the neighborhood function Eq. 3.3 where the Euclidean distance $d_{\mathcal{M}}(a, a^*)$ in the map space is replaced with the hyperbolic distance $\delta_{\mathcal{M}}(z_a, z_{a^*})$ given by Eq. 3.24. Thus, the HSOM's neighborhood function is then given as

$$h(a, a^*) = \exp\left(-\frac{\operatorname{arctanh}\left(\left|\frac{z_a - z_{a^*}}{1 - \bar{z}_a z_{a^*}}\right|\right)}{\sigma^2(t)}\right), \tag{3.33}$$

where $z_a \in \mathbb{D}$ corresponds to the 2D position of node a in the Poincaré Disk.

As we see, the extension of the SOM to the HSOM is of remarkably simplicity: The complete SOM framework can be reused for the HSOM, but by plugging in a different metric into the neighborhood function h, we change the fundamental geometric properties of the SOM's map space $\mathcal{M}$. We shall see in the next section how this affects the performance and usability of the HSOM compared to the standard Euclidean SOM.

3.4 Comparing Euclidean and Hyperbolic SOMs

Both, the standard Euclidean SOM and the HSOM achieve a non-linear mapping from a high-dimensional continuous input space $\mathcal{X}$ to a two-dimensional discrete map space $\mathcal{M}$ in which data from the input space can be visualized. Setting aside the special features of hyperbolic space offering a "fish-eye" view on the data, we are interested in answering the question, how the quality of these mappings might be measured.

From an analytical point of view there is no straight-forward answer to this question. Ritter and Schulten (1988) have proposed a cost function for the SOM given by

$$E = \sum_k \sum_i h_{ia^*} \|\mathbf{x}_k - \mathbf{w}_i\|^2, \tag{3.34}$$

where a^* is the index of the best-match unit for the input $\mathbf{x}_k$. Eq. 3.34 only holds for a fixed neighborhood function h and a discrete data set. Under the assumption $\sum_j h_{ij} = 1$ for all i it can be decomposed into two terms (Kaski, 1997) as

$$E = \sum_k \|\mathbf{x}_k - \mathbf{v}^*(\mathbf{x}_k)\|^2 + \sum_i \sum_j h_{ij} N_i \|\mathbf{v}_i - \mathbf{w}_j\|^2, \tag{3.35}$$

where $\mathbf{v}^*(\mathbf{x}_k)$ is the centroid of the Voronoi region corresponding to the best-match unit for $\mathbf{x}_k$. $\mathbf{v}_i$ is the centroid of the Voronoi region of the unit with reference vector $\mathbf{w}_i$ to which a number of N_i data items are mapped to.

The first part of Eq. 3.35 measures the global quantization accuracy of the map, whereas the second term can be interpreted as a criterion for the topological ordering of the map's reference vectors. Unfortunately, it is highly depended on the map size and the neighborhood function h. Therefore, it is not suitable to compare maps of different sizes or with different neighborhood functions. As a remedy, the easy computable quantization error has traditionally been used as a separate goodness measure, and different heuristics to assess the neighborhood preservation of a map have been proposed as detailed below.

3.4.1 Quantization Error

The *average quantization error* over the input samples is usually defined as

$$E_{q\mathcal{X}} = \frac{1}{N_{\mathcal{X}}} \sum_i \|\mathbf{x}_i - \mathbf{w}^*(\mathbf{x}_i)\|, \tag{3.36}$$

where $N_{\mathcal{X}}$ is the total number of data items, and $\mathbf{w}^*(\mathbf{x}_i)$ is the prototype vector of the best-match unit for the input $\mathbf{x}_i$ (cf. Eq. 3.1). Hence, $E_{q\mathcal{X}}$ is a global measure for the average precision with which a data item is represented by the map.

The premier utilization of the SOM is not the classification of data, but rather its visualization. In such an exploratory data analysis scenario, the user is probably more interested in how faithful a certain node describes the data within its Voronoi cell. We therefore compute

$$E_{qa} = \frac{1}{N_a} \sum_{i \in V_a} \|\mathbf{x}_i - \mathbf{w}_a\|, \tag{3.37}$$

where V_a is the index set of all data items within the Voronoi cell of unit a, and N_a the cardinality of that set. Thus, E_{qa} is a local measure for the quantization error of each node a in the map. By averaging over all nodes, we obtain a global measure describing how well the nodes of a map reflect the data distribution:

$$E_{q\mathcal{M}} = \frac{1}{N_{\mathcal{M}}} \sum_a E_{qa}, \tag{3.38}$$

with $N_{\mathcal{M}}$ being the number of nodes in the SOM. Note, that both $E_{q\mathcal{X}}$ and $E_{q\mathcal{M}}$ describe an average quantization error, but from different perspectives and not necessarily yielding the same results.

3.4.2 Neighborhood Preservation

Besides the precision of a SOM in terms of its quantization error, the other important criterion for its quality is the degree of topological ordering it achieves, i.e. how well the mapping preserves neighborhood distances in the original data space. Since there is no straight-forward definition either of the concept of "neighborhood" or "preservation" there have been several suggestions to come up with a single numerical measure to assess the topology-preserving capability of a SOM (Bauer and Pawelzik, 1992; Villmann et al., 1994; Luttrell, 1994; Kaski and Lagus, 1996; Bezdek and Pal, 1995; Goodhill et al., 1996; Venna and Kaski, 2001).

C Measure

Goodhill and Sejnowski (1997) proposed a unified measure of topographic distortion called the *C measure*. Using our notation it is defined as

$$C = \sum_{i=1}^{N} \sum_{j<i} d_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)\, d_{\mathcal{M}}(\mathbf{w}^*(\mathbf{x}_i), \mathbf{w}^*(\mathbf{x}_j)), \tag{3.39}$$

where $d_{\mathcal{X}}$ and $d_{\mathcal{M}}$ are symmetric distance functions in the input and map space, respectively. They show, that by plugging in different functions for $d_{\mathcal{X}}$ and $d_{\mathcal{M}}$ different measures such as *metric multidimensional scaling* (Shepard, 1980), *minimal wiring* and *minimal path length* (Durbin and Mitchison, 1990), or *minimal distortion* (Luttrell, 1994) can be achieved. However, the choice of the distance functions introduces further parameters and Goodhill and Sejnowski (1997) show that the quality of the different measures is highly dependent on the choice of the initial mapping problem, i.e., whether it is "desired that generally nearby points should always map to generally nearby points as much as possible in both directions", or if one is more concerned in local continuity either in the input space $\mathcal{X}$ or the map space $\mathcal{M}$.

The findings of Goodhill and Sejnowski (1997) stress an inherent drawback of a single criterion for the topology-preserving capability of the SOM: It is always a trade-off between measuring global and local neighborhood preservation, which makes its advisability highly depend on the actual problem domain.

In our comparisons of Euclidean and Hyperbolic SOMs we therefore use two measures to compare a SOM's neighborhood preservation: *(i)* Spearman's rho to assess the *global ordering*, and *(ii)* a local measure called "trustworthiness" (Venna and Kaski, 2001) to evaluate the *local ordering* within a map.

Spearman's Rho

Bezdek and Pal (1995) introduced the notion of a *metric topology preserving* (MTP) transformation. Their idea is to examine the rank order of distance pairs in the input and map space, $\mathcal{X}$ and $\mathcal{M}$, respectively. According to their definition, MTP transformations "carry neighbors in $\mathcal{X}$ to neighbors in $\mathcal{M}$, and preserve (all) relative distance relationships". In their paper, they propose to compute Spearman's ρ as a measure for the MTP quality of a mapping. It is defined as the linear correlation coefficient of the ranks R_i and S_i (Press et al., 1992)

$$\rho_{\text{Sp}} = \frac{\sum_i (R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum_i (R_i - \overline{R})^2} \sqrt{\sum_i (S_i - \overline{S})^2}} \tag{3.40}$$

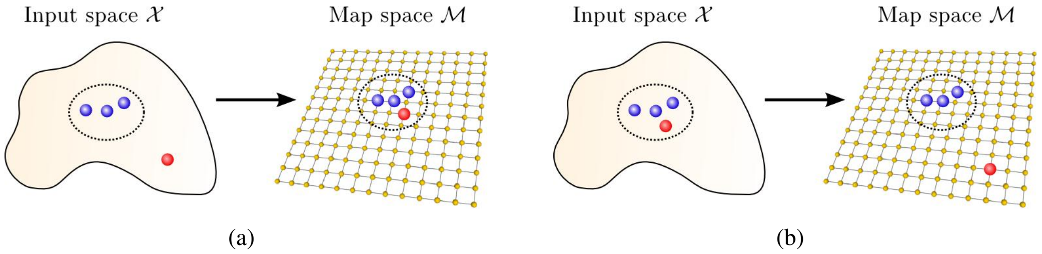
where R_i and S_i are the ranks of the ordered lists of the distances given by $d_{\mathcal{X}}$ and $d_{\mathcal{M}}$, respectively. It is therefore a measure for the global neighborhood preservation of a mapping function. A very useful property of ρ_{Sp} is its boundedness on the interval $[-1, 1]$, making it a viable candidate for comparing different maps. Bezdek and Pal (1995) prove that a mapping is *metric topology preserving* if and only if $\rho_{\mathrm{Sp}} = 1$. As ρ_{Sp} decreases from one, the mapping is becoming less MTP, and $\rho_{\mathrm{Sp}} = 0$ indicates a complete random mapping in terms of distance preservation.

Trustworthiness and Continuity

Spearman's rho measures the overall global mapping quality of the SOM. However, for an interactive visualization framework where the user explores the data on a map, a measure quantifying the goodness of a local patch on the map might be more meaningful. Venna and Kaski (2001) point out that any multi-dimensional scaling method introduces two kinds of errors when considering local neighborhoods in the input or the map space:

1. Data items within an ϵ-neighborhood in the map space $\mathcal{M}$ might actually come from distant regions in the input space $\mathcal{X}$ as indicated in Fig. 3.14(a).

2. Data items within an ϵ-neighborhood in the input space $\mathcal{X}$ might be mapped to distant regions in the map space $\mathcal{M}$ as shown in Fig. 3.14(b).



(a) (b)

Figure 3.14: Local errors affecting (a) trustworthiness and (b) continuity.

The first type of error might mislead a user to accept similarities in patterns which in fact are not present in the data, while the second type introduces discontinuities resulting in the loss of original data relationships within the mapping.

Venna and Kaski (2001, 2005) propose the two measures of *trustworthiness* and *continuity* to quantify the two errors described above. They are defined as

$$T(n) = 1 - S \sum_{i=1}^{N_{\mathcal{X}}} \sum_{j \in \tilde{X}_n(i)} (r_{\mathcal{X}}(i,j) - n) \tag{3.41}$$

and

$$C(n) = 1 - S \sum_{i=1}^{N_{\mathcal{X}}} \sum_{j \in \tilde{M}_n(i)} (r_{\mathcal{M}}(i,j) - n) \tag{3.42}$$

where $N_{\mathcal{X}}$ is the number of data items, $\tilde{X}_n(i)$ is the set of items within a neighborhood of n samples around data item i in the map space $\mathcal{M}$, but *not* in the input space $\mathcal{X}$ (e.g., the red item in Fig. 3.14(a)); and $r_{\mathcal{X}}(i, j)$ is the rank of item j in the ordered list of distances to item

i given by the distance $d_{\mathcal{X}}$ in the input space. The scaling factor S given by

$$S = \frac{2}{N_{\mathcal{X}} n(2N_{\mathcal{X}} - 3n - 1)} \tag{3.43}$$

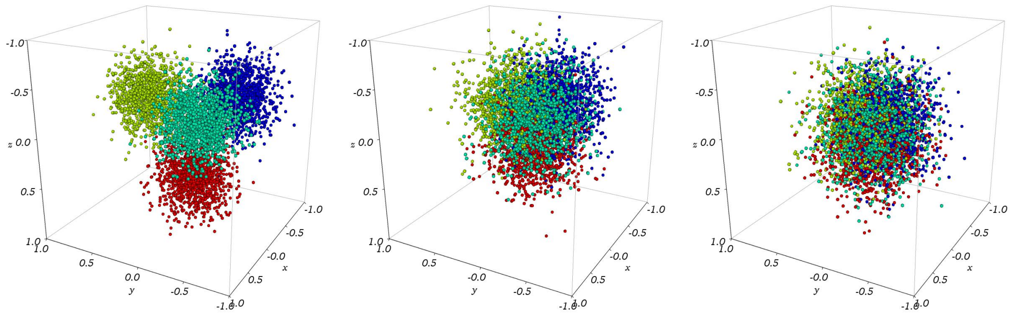
forces the values of $T(n)$ and $C(n)$ between zero and one. The definition of $\tilde{M}_n(i)$ and $r_{\mathcal{M}}(i,j)$ for the computation of $C(n)$ is analogue: $\tilde{M}_n(i)$ is the set of items within a neighborhood of n samples around data item i in the input space $\mathcal{X}$, but *not* in the map space $\mathcal{M}$ (e.g., the red item in Fig. 3.14(b)); and $r_{\mathcal{M}}(i,j)$ is the rank of item j in the ordered list of distances to item i given by the distance $d_{\mathcal{M}}$ in the map space.

3.4.3 Benchmarks with an N-Dimensional Tetreader

In order to compare SOM and HSOM with respect to the quality measures discussed above, we start with a rather simple dataset comprised of N+1 Gaussian spheres in N-space.

Qualitative Analysis

For N=3, we obtain an intuitively graspable dataset which can be easily visualized as shown in Fig. 3.15. It consists of four spheres where each of the spheres is made up of 1250 points given by a 3-dimensional Gaussian distribution with variance σ. By varying σ we are able to control the degree of mixture between the Gaussian spheres and thus the difficulty for the SOMs to show the four clusters in their mappings.



(a) $\sigma = 0.2$ (b) $\sigma = 0.5$ (c) $\sigma = 1.0$

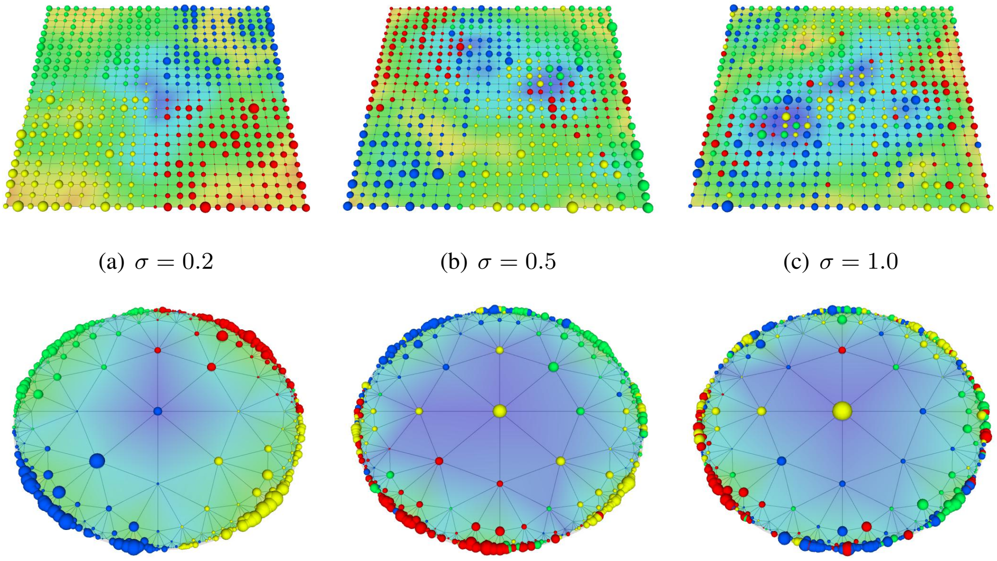
Figure 3.15: A three dimensional dataset consisting of four equidistant Gaussian spheres. From the left to the right the variance of the data distribution within the spheres decreases from $\sigma = 0.2$ in (a) to $\sigma = 1.0$ in (c).

For the comparison we have trained maps of three different sizes as given in Table 3.1. We applied the (H)SOMs to two 3- and two 10-dimensional tetraeder datasets with $\sigma = 0.2$ and $\sigma = 1.0$. Each (H)SOM was initialized and trained 10 times in order to obtain a reasonable statistic. We therefore trained 240 (H)SOMs for the comparison with respect to the tetraeder dataset.

SOM	13×13 (169)	25×25 (625)	48×48 (2304)
HSOM	8^3 (161)	8^4 (609)	8^5 (2281)

Table 3.1: Map sizes used for comparison of SOM and HSOM using the tetraeder dataset. The number in brackets denotes the number of nodes in the map.

Typical representatives of the mid-sized maps are shown in Fig. 3.16. Both map types, SOM and HSOM, show the same tendency with increasing σ: The ordering of the clusters within the map becomes less structured and the boundaries between them less dominant. A difference between the two map types becomes evident, when visualizing the length of the prototype vectors. To this end we can interpretate the node vertices and their connecting lines as a 2D mesh defining the surface of our maps. The surface color can then be interpolated continuously between node positions. The mappings show a HSV blue-to-red color-scale on their canvases, reflecting how close to the origin of $\mathbb{R}^N$ their prototype vectors are: Regions with prototypes from the periphery of $\mathbb{R}^N$ are painted red, whereas prototypes near to the origin are colored blue. For $\sigma = 0.2$, Fig. 3.15(a) shows that the density of data samples in the origin is comparatively sparse. In the cube center the spheres overlap only very slightly. This is also reflected by the maps which show small blue areas in their centers. As σ increases, more and more data items enter the central region of the cube. This is also reflected by the maps. However SOM and HSOM behave somewhat different: While for the SOM the deep blue regions move away from the center of the map and tend to occupy disconnected areas, the HSOM still displays a single deep blue region in its center.



(a) $\sigma = 0.2$ (b) $\sigma = 0.5$ (c) $\sigma = 1.0$

Figure 3.16: Visualization of the 3-dimensional tetraeder dataset with a 25×25-SOM and a 8^4-HSOM for varying values of σ. Node sizes reflect the number of data items mapped to a node. Node colors correspond to the originating sphere from the tetraeder dataset, e.g. a node is colored red when the majority of samples mapped to it comes from the red sphere. Additionally, the ground color reflects the Euclidean length of a node's prototype vector, i.e., a blue colored region signals that its prototype vectors are close to the origin of the dataset.

This effect becomes more pronounced when moving from the 3-dimensional dataset to a 10-dimensional hyper-tetraeder. Fig. 3.17 shows the resulting maps obtained with a 25×25-SOM (a) and a 8^4-HSOM (b). The dataset consists of 11 Gaussian spheres which are equally spaced in the 10-dimensional space. Due to the geometric properties of such a high-dimensional space most data samples are located in the periphery of the $\mathbb{R}^{10}$. Fig. 3.17(a) shows a 25×25 SOM which displays the 11 clusters as distinct regions on the map. However, the original symmetry of the dataset with equidistant spaced clusters in the periphery and an “empty” region in the center seems to be lost: The dark blue ground color reflecting the center of $\mathbb{R}^{10}$ is evenly spread across the whole map. In case of the HSOM, Fig. 3.17(b) shows a different picture: The center of the map consists of a void blue region and virtually all data items get mapped to the periphery of the hyperbolic map. This rendering might reflect the geometrical properties of the original dataset more faithfully, however at the cost of sacrificing visualization space: The user is initially presented a mostly “empty” data space.

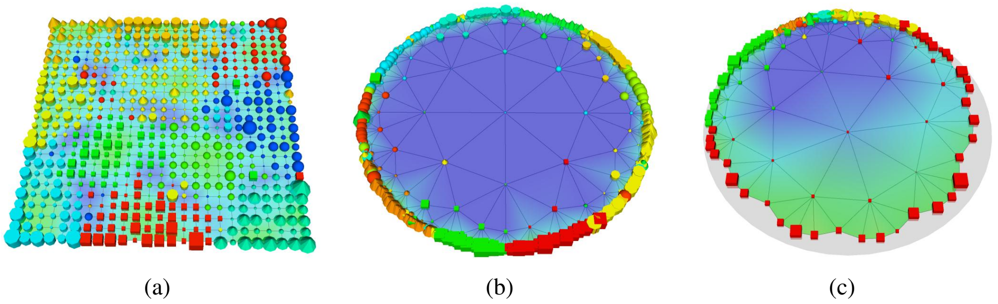


Figure 3.17: Visualization of the tetraeder dataset for $N = 10$ and $\sigma = 0.3$. The datasets consists of 11 Gaussian spheres in 10 dimensions. The same visual attributes as in Fig. 3.16 are used. In (c) the focus on the hyperbolic plane was moved into the 5 o'clock direction towards the periphery. The ground color of the plane indicates that data in this map region originates from the periphery of $I\!R^{10}$.

As indicated in Fig. 3.17(c), a navigation to the periphery is necessary to inspect the data more closely. In Chapter 4 we discuss a hierarchical variant of the HSOM which addresses this drawback.

Quantitative Analysis

After comparing HSOM and SOM on a qualitative level in the previous section, we now turn to a more thorough quantitative analysis. Throughout the following section, four datasets are used for the comparison: two 3- and two 10-dimensional tetraeders, each with $\sigma = 0.2$ and $\sigma = 1.0$.

Quantization Error. Fig. 3.18 shows the quantization error $E_{q\mathcal{M}}$ as given by Eq. 3.38. Note, that the variance over 10 runs resulted in error-bars significantly smaller than the symbols used for the plot. In terms of $E_{q\mathcal{M}}$ we can conclude, that for small maps the difference between SOM and HSOM is negligible, for larger maps however, the HSOM consistently achieves results approximately 25% better than the SOM. This is independent on the dimensionality or variance of the tetraeder dataset.

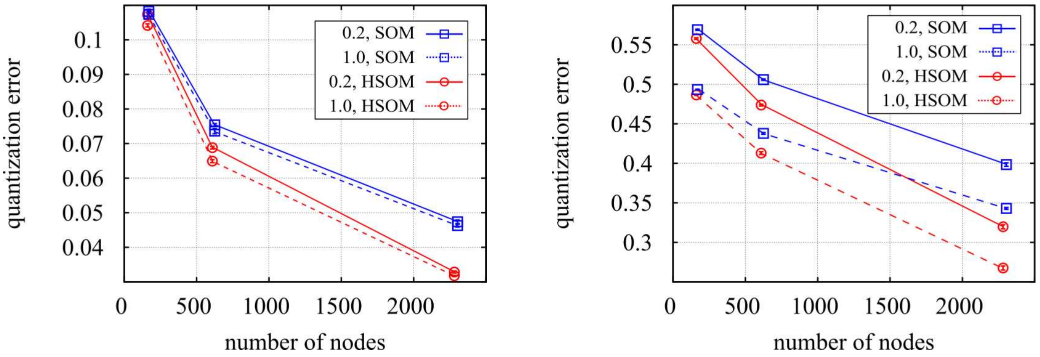


Figure 3.18: Quantization errors of differently sized SOMs and HSOMs for the 3-dimensional (left) and 10-dimensional (right) tetraeder dataset. All results were averaged over 10 runs.

Global Neighborhood Preservation. The global ordering capabilities of SOM and HSOM as measured by Spearman's rho are given by Fig. 3.19.

For the 3-dimensional tetraeder, the SOM achieves consistently better results than the HSOM. Additionally, the global ordering capability of the SOM does not depend on the map size, whereas the HSOM's global ordering gets worse with increasing map size. For

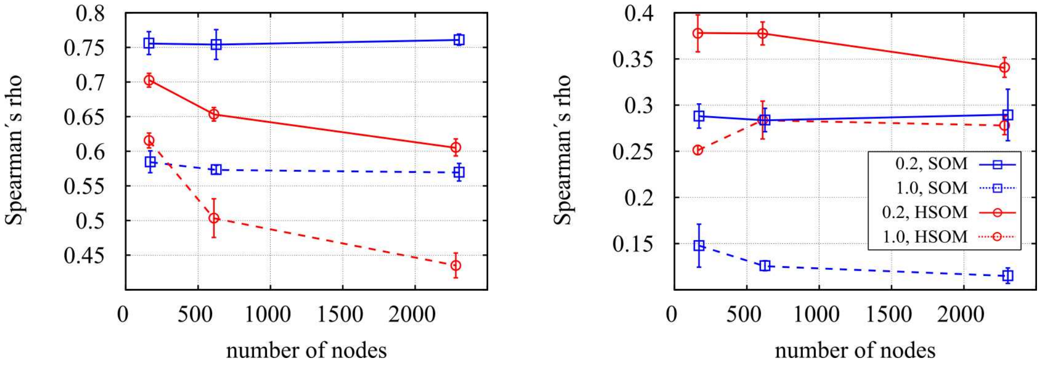


Figure 3.19: Spearman's rho for differently sized SOMs and HSOMs for the 3-dimensional (left) and 10-dimensional (right) tetraeder dataset. All results were averaged over 10 runs.

the 10-dimensional dataset, the HSOM is able to benefit from the additional growth the hyperbolic grid offers and now outperforms the SOM in terms of Spearman's rho.

The reason for the differing performance of SOM and HSOM becomes evident when we consider the scatter plots as shown in Fig. 3.20: For each data pair (i, j) the scatter plots show a point given by $(d_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j), d_{\mathcal{M}}(w^*(\mathbf{x}_i), w^*(\mathbf{x}_j)))$. For the low-dimensional case as shown on the left hand side, the SOM exhibits a distribution of pair-wise distances with a linear correlation close to one, resulting in a high value for the corresponding rank correlation measured by Spearman's rho.

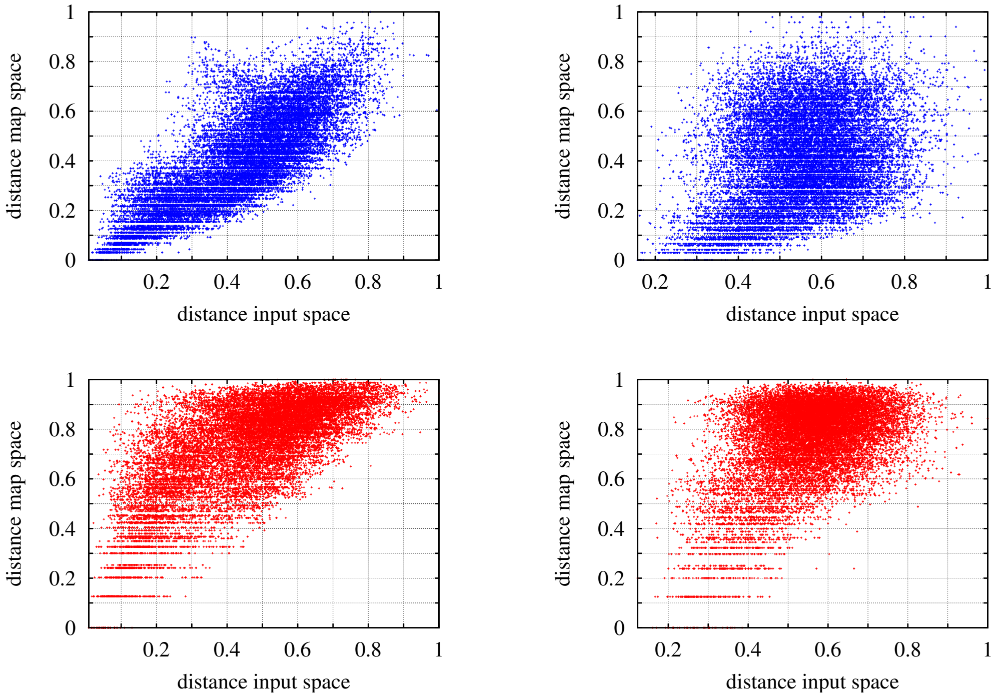


Figure 3.20: Scatter plots of the distances between two data items in the low dimensional map space (y-axis) over the distances in the high dimensional feature space (x-axis). For better comparability, all distance values were scaled to the interval $[0, 1]$.

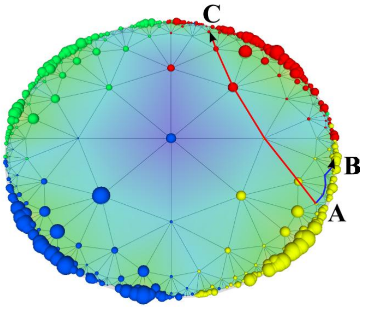
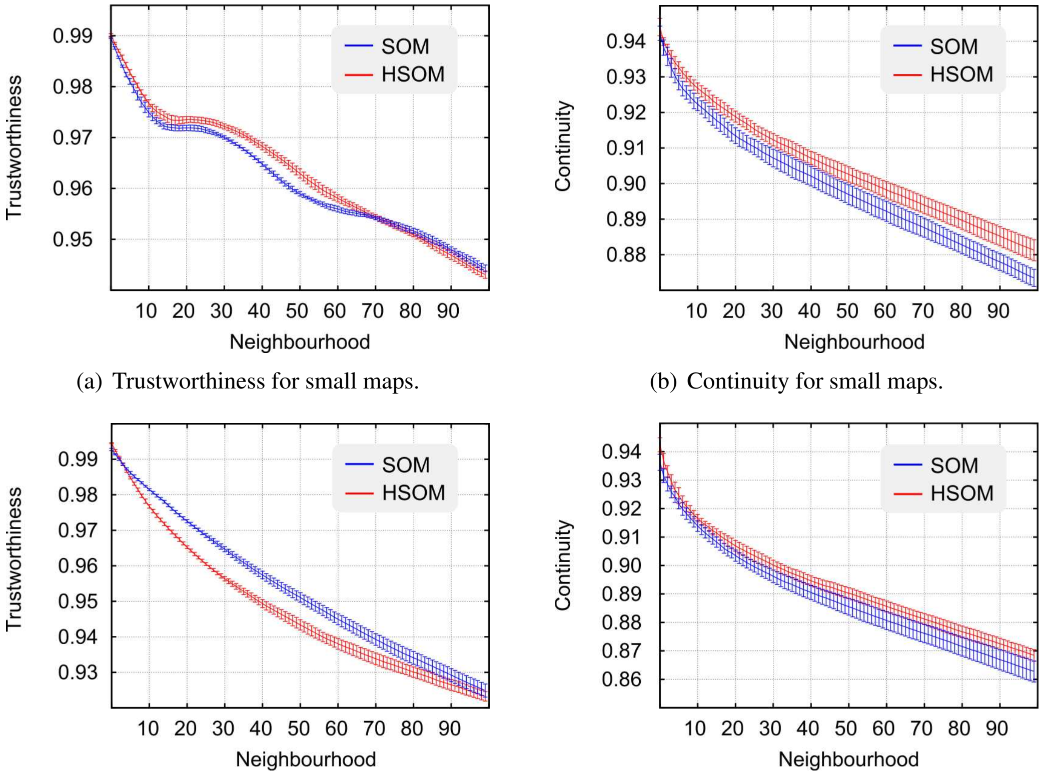


Figure 3.21: Since all triangles of the tessellation are equilateral, points *B* and *C* are equidistant from *A*.

In case of the HSOM, a non marginal amount of data pairs covers the upper left region of scatter plot (c), indicating that data items close together in the input space get mapped to distant regions in the map space. This is caused by the peculiar non-Euclidean metric: The distance between any two points in the periphery of the HSOM quickly approaches the maximal distance which can be achieved within the hyperbolic map. As an example consider Fig. 3.21: The distance between the two points A and B is given by the path along the blue curve. Although in the projection point C seems to be farther away from A than B, they are actually equidistant. In other words: The asymptotically exponential growth of hyperbolic space allows a larger fraction of nodes to be far away from each other than in the Euclidean case. For low-dimensional data this scaling behavior results in a mismatch of distances in input and map space. For larger dimensions however, the distribution of node distances in the HSOM are more suitable to describe the true data distribution as can be seen on the right hand side of Fig. 3.20.



(a) Trustworthiness for small maps.

(b) Continuity for small maps.

(c) Trustworthiness for large maps.

(d) Continuity for large maps.

Figure 3.22: Trustworthiness and continuity for small (13x13 SOMs and 8^3 HSOMs) and large (48x48 and 8^5 HSOM) maps on the 10-dimensional tetraeder dataset with $\sigma = 0.2$. All results were averaged over 10 runs.

Local Neighborhood Preservation. Figure 3.22 shows the local ordering capabilities of SOM and HSOM with respect to the N-dimensional tetraeders. Note, that the measures trustworthiness $T(n)$ and continuity $C(n)$ as defined by Eq. 3.41 and 3.42, are dependent on the size of the accounted neighborhood size n. Therefore, we need a plot over n for each map in order to compare their local ordering performances.

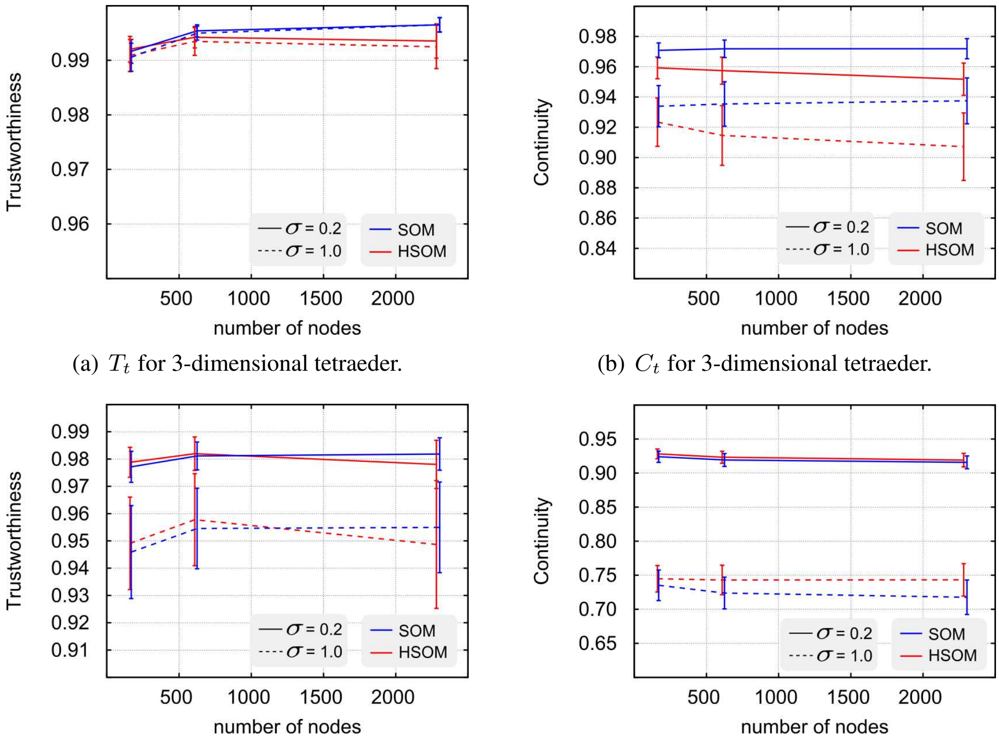
The figures show that the HSOM generally achieves a higher trustworthiness and continuity than the SOM on the 10-dimensional tetraeder for all neighborhood sizes. Only for $n < 4$ the SOM achieves a higher trustworthiness when mapping the dataset to a large map (Fig. 3.22(c)).

For a complete comparison of all datasets on all maps, we would need 48 plots. In order to assess both measures with a single quantity for a more convenient comparison, we compute the sum of $T(n)$ and $C(n)$ over a fixed neighborhood size. We call the resulting indicators *total trustworthiness* and *total continuity*, defined as the normalized sum over the first $n_{\max} = 20$ neighbors:

$$T_t = \frac{1}{20}\sum_{n=1}^{20} T(n) \quad \text{and} \quad C_t = \frac{1}{20}\sum_{n=1}^{20} C(n) \tag{3.44}$$

respectively. The value of $n_{\max} = 20$ is heuristically motivated and reflects the observation that humans tend to concentrate on limited neighborhoods in visual searches (Guan, 2007). A value of 20 was also used by Venna and Kaski (2005) for controlling the tradeoff between trustworthiness and continuity in their local multidimensional scaling approach. We believe it is a rather uncritical parameter, making it possible to provide a simple scalar comparison of local neighborhood preservation capabilities.

The results in Fig. 3.23 indicate that SOM and HSOM are generally on par with their local ordering capabilities in case of the rather simple tetraeder datasets. Only for the 3-dimensional data, the SOM appears to offer advantages with respect to the continuity of the mapping.



(a) T_t for 3-dimensional tetraeder.

(b) C_t for 3-dimensional tetraeder.

(c) T_t for 10-dimensional tetraeder.

(d) C_t for 10-dimensional tetraeder.

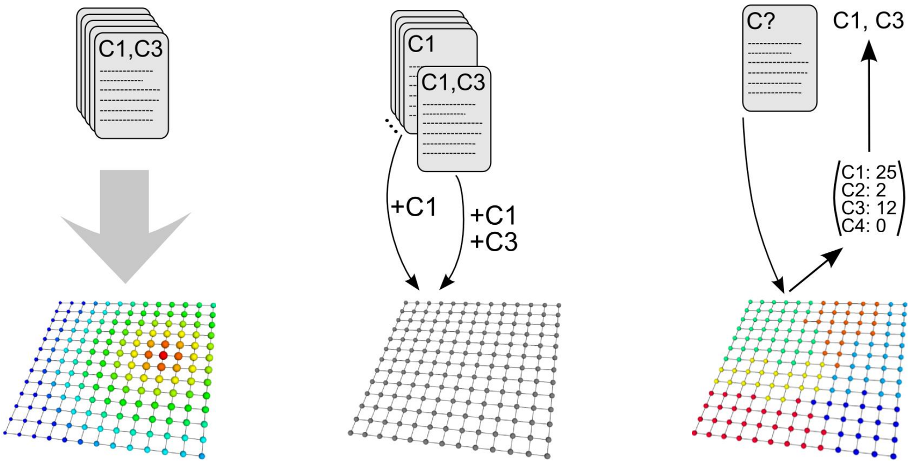
Figure 3.23: Total trustworthiness and total continuity for differently sized SOMs and HSOMs for the 3-dimensional (top) and 10-dimensional (bottom) tetraeder dataset. All results were averaged over 10 runs.

3.5 Text Categorization Tasks

After discussing the fundamental differences between SOM and HSOM for a rather simple, but intuitively graspable dataset, we now turn to more complex, real world problems from the domain of unstructured text data. Before going into more detail about the specific data sets we first turn to the question how the SOM and HSOM can be utilized for text categorization tasks (cf. Section 2.3.1).

3.5.1 SOMs in Context of Text Categorization and Information Retrieval

In order to categorize unknown documents with respect to a known text corpus, we apply the following three steps as also shown in Fig. 3.24.



(a) Training (b) Labelling (c) Retrieving

Figure 3.24: Text categorization steps: (a) First a training set is used to build an internal model of the collection represented by the SOM's reference vectors. (b) To each neuron we attach an additional topic vector. For training documents with known categories the elements of the topic vector belonging to the corresponding winner node are incremented. (c) For unknown documents mapped to the SOM, we can retrieve the corresponding topic vector from its winner node.

(a) In a first step, the SOM or HSOM is trained in standard fashion with a given text corpus. In order to use the corpus for categorization tasks a significant proportion of the dataset has to be labelled with elements from a set of topics or categories.

(b) In a second step we attach a *topic vector* $\mathbf{t}_a \in I\!R^{N_C}$ to each neuron $a \in \mathcal{A}$, where N_C is the number of distinct categories occurring in the document collection, and $\mathcal{A}$ the formal set of neurons of the SOM.

Each document of the training set is then mapped to the SOM. If the categories of the document are known, i.e. when it was previously labelled, we increment the corresponding elements of the topic vector attached to its winning neuron by one.

(c) A new unknown document $\tilde{D}$ can be categorized by mapping it to the previously build SOM. By retrieving the topic vector $\mathbf{t}_{a^*}$ of the best match node a^* for $\tilde{D}$, we can compute the measure $C_k(\tilde{D})$ for each topic k:

$$C_k(\tilde{D}) = \frac{t^k_{a^*(\tilde{D})}}{\sum_i^{N_C} t^i_{a^*(\tilde{D})}}, \tag{3.45}$$

where $t^i_{a^*}$ is the ith component of the topic vector $\mathbf{t}_{a^*}$. $C_k(\tilde{D})$ can be regarded as a confidence measure that the new document $\tilde{D}$ should be labelled with topic k.

Information Retrieval (IR) Tasks. In the context of IR the user is not interested in classifying unknown documents, but to retrieve all documents related to a certain topic from a large data set. IR systems are commonly evaluated in terms of precision and recall (cf. Section 2.2.2). In order to compute these measures we need a ranking function for the SOM which retrieves all documents of a given category ordered by relevance. By computing

$$r_k(D) = C_k(D), \quad t^k_{a^*(D)} d_{\mathcal{X}}(\mathbf{w}^*(D), \mathbf{f}(D)), \tag{3.46}$$

where $\mathbf{f}(D)$ is the feature vector describing document D, we are able to sort all documents according to their relevance to topic k. If $C_k(D_1) = C_k(D_2)$, the second part of Eq. 3.46 is used to rank the ties.

3.5.2 Artificial Bag Of Words with 20 Clusters

In order to be able to systematically study the behavior of SOM vs. HSOM for unstructured text data, we propose the employment of an artificial dataset closely resembling the characteristics of the so-called “bag of words” model. This model is the standard approach to represent unstructured text data in a high-dimensional vectorial feature space (Salton and Buckley, 1988), cf. Section 2.2.1.

Assume, we have a text corpus made up of N_T different terms covering a number of N_C different categories. We then generate an artificial bag of words with the probability density function $P(t)$ for the occurrence of term t:

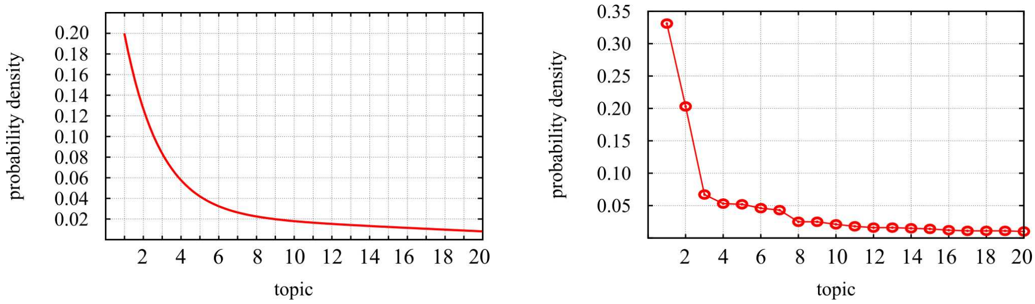
$$P(t) = \frac{1}{\sqrt{2\pi}\sigma}\left(P_b(t) + \sum_{c=1}^{N_C} \beta_c P_c(t)\right), \tag{3.47}$$

where $1/\sqrt{2\pi}\sigma$ is a normalization factor, β_c are weighting coefficients, and

$$P_b(t) = \exp\left(-\frac{(t-\mu_b)^2}{2\sigma_b^2}\right), \quad P_c(t) = \exp\left(-\frac{(t-\mu_c)^2}{2\sigma_c^2}\right) \tag{3.48}$$

are Gaussian distributions centered at μ_b, μ_c with variance σ_b^2, σ_c^2, respectively.

In the context of the bag of words feature space, the distribution of terms t given by $P_b(t)$ can be regarded as a *base vocabulary* which is present in all documents. Additionally, a document might contain terms t from distributions $P_c(t)$ which are connected to certain *topic categories c*. For the reported experiments we use the following parameters: $N_T = 1000$ different terms with a base vocabulary modeled by $\mu_b = 50$ and $\sigma_b = 25$. The distribution of terms in the $N_C = 20$ topic categories are all equally spaced with $\Delta\mu_c = 45$ and $\sigma_c = 12.5\ \forall c$.



(a) (b)

Figure 3.25: Probability density functions $P(c)$ for the occurrence of categories in document sets. (a) our chosen $P(c)$ for the artificial document set and (b) the actual distribution probability for topics in the Reuters-21578 collection (see below).

We now need to decide, how many and which topics an artificial document covers. Many real world text databases contain multi-class labels (Baeza-Yates and Ribeiro-Neto, 1999) since texts usually cover more than one single topic. In our model the random variable $X_T = 1 + g_{rnd}(2)$ determines the number of categories an artificial text covers, where $g_{rnd}(2)$ is a Gaussian random variable with variance 2. We then build a set of X_T randomly chosen categories C_{X_T}, where the probability for category c to be chosen is given by $P(c) = \nu\,(\exp(c)+\beta c+\alpha)$. For the reported results, we use $\beta = 100$, $\alpha = 1000$ and a normalization factor ν normalizing the sum $\sum_{c=0}^{N_C} P(c) = 1$, resulting in a probability density function as shown in Fig. 3.25.

After determining the categories which should be present for the artificial document, we set the coefficients β_c according to

$$\beta_c = \begin{cases} 1 + g_{rnd}(0.2) & \text{if } c \in \mathcal{C}_{X_T} \\ R(g_{rnd}(0.2)) & \text{otherwise,} \end{cases} \tag{3.49}$$

where $g_{rnd}(0.2)$ is a Gaussian random variable with variance 0.2, and $R(x)$ the ramp function, defined by

$$R(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.50}$$

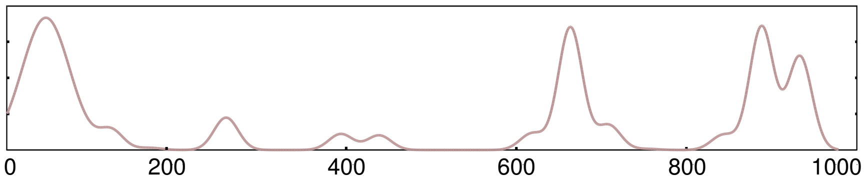


Figure 3.26: Example for the probability density function $P(t)$

An example for a typical probability density function $P(t)$ is shown in Fig. 3.26. In addition to terms of the base class (term 1 - 100) and noise terms there is also a high probability for the occurrence of words around the terms 635, 860 and 905 – representing the topics 13, 18 and 19, respectively.

By choosing such a model we are able to generate a high dimensional dataset with similar characteristics as the traditional bag of words model in a well defined manner. For the experiment we used a dataset with 5,000 items and trained four different-sized standard SOMs with grid sizes of 13x13, 25x25 and 48x48 with 169, 625 and 2304 nodes, respectively. For the HSOM, we used a topology with $n = 8$ neighbors and ring sizes of 3, 4 and 5 rings with 161, 609 and 2281 nodes, respectively. Each (H)SOM was initialized and trained 10 times in order to obtain a reasonable statistic.

Quantization Error and Neighborhood Preservation. With respect to quantization error $E_{q\mathcal{M}}$ and Spearman's rho, SOM and HSOM show similar characteristics as observed for the tetraeder dataset: For small maps, there is no distinctive difference between the two, and for larger maps, the HSOM is in slight advantage over the SOM. Note, that the variance of ρ for the SOM in Fig. 3.27 is noticeable larger than that for the HSOM. This result might indicate, that the SOM is more prone to local foldings which deteriorate the global ordering within the map.

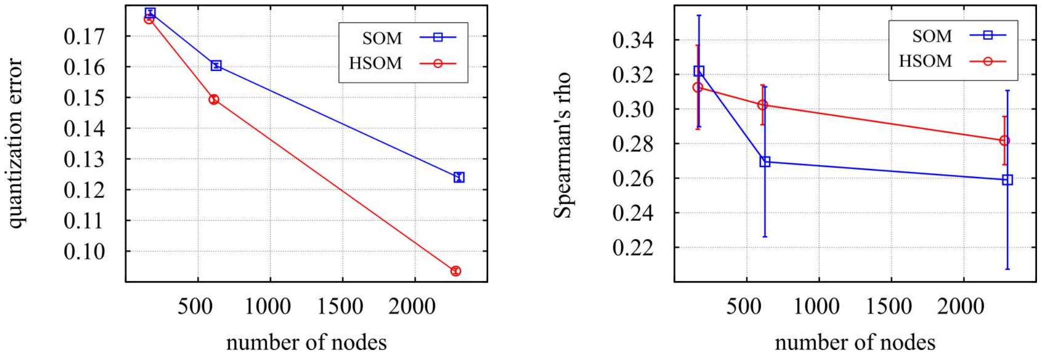


Figure 3.27: Quantization errors (left) and Spearman's rho (right) for the artificial bag of words dataset.

In case for the local neighborhood preservation, the measures of total trustworthiness and total continuity as defined by Eq. 3.44, indicate that the SOM outperforms the HSOM.

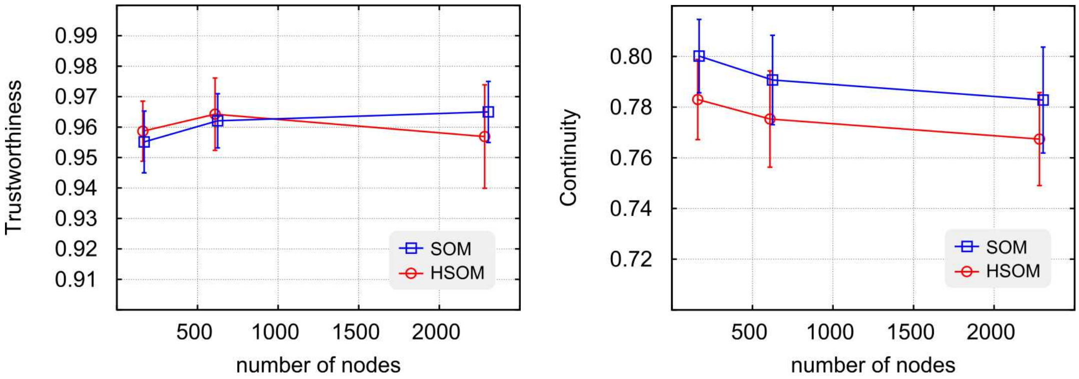


Figure 3.28: Total trustworthiness T_t (left) and continuity C_t (right) for the artificial bag of words dataset.

Classification Accuracy. In order to compare the capabilities of SOM and HSOM in the context of information retrieval systems and classification accuracy, we provide results on the measures of *precision* and *recall* as discussed in Section 2.2.2. The outcome as shown in Fig. 3.29 suggests that the difference between SOM and HSOM is only marginal. For less frequent topics the HSOM tends to achieve results with higher precision for any given recall-level. This leads to a "hump" in the micro-averaged results in Fig. 3.29 on the right, indicating the slight advantage of the HSOM over the SOM.

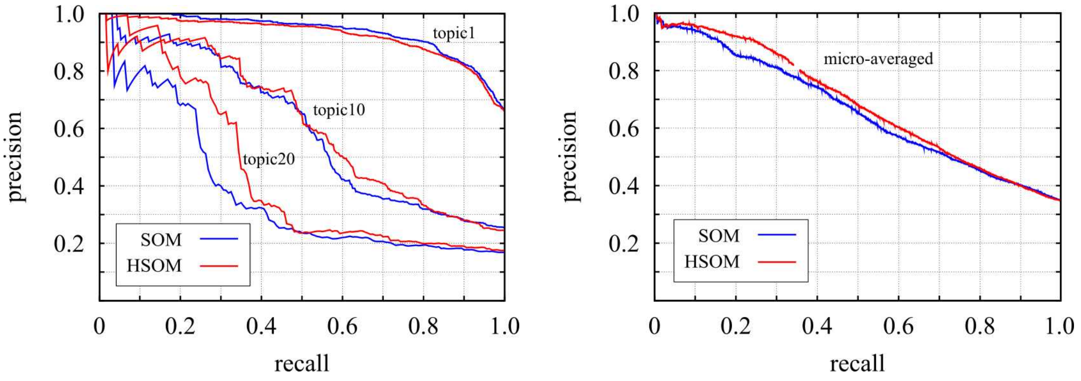


Figure 3.29: Precision over recall for the artificial document dataset. In (a) the curves for the categories 1, 10, and 20 are shown, in (b) the micro-averaged results over all 20 categories.

3.5.3 Reuters-21578 Newswire Articles

To complete the comparison of SOM vs. HSOM, we compare their performances on real world data, the Reuters-21578[1] newswire dataset. It consists of an assembly of articles which appeared on the Reuters newswire in 1987. They were collected and manually labelled by Reuters personell in order to provide the scientific community with a publicly available dataset for text categorization research. Since then it has become a standard benchmark in text mining applications (Joachims, 1998; Yang, 1999; Sebastiani et al., 2000; Hotho et al., 2003a).

The Reuters-21578 collection contains 120 distinct categories of which many occur only once in the whole dataset. The distribution of the top 20 topics of the collection is shown in Fig. 3.25 (b). Exemplary articles for the top three categories "earn" *(a)*, "acquisition" *(b)*, "money exchange rate" *(c)*, plus one article of the 20th most frequent topic "soybean" *(d)* are given in Fig. 3.30.

There has been extensive work on different document representations, feature selection or term weighting approaches for the Reuters data. For simplicity we here follow the classical *bag of words* model. After word stemming and stop word removal we arrive at a vocabulary of 5093 unique word stems $\{w_i\}$. Following standard practice (Salton and Buckley, 1988) we haven chosen a *term frequency* × *inverse document frequency* weighting scheme and applied the cosine metric for distance computations in the bag of words feature space.

All experiments were conducted with the "ModApte" split which is most commonly used in other studies. It defines a set of 9603 training and 3299 test documents in the Reuters-21578 collection. All reported results were obtained with a 48×48 SOM and a 8^5 HSOM each trained with 10 separate runs to collect the statistics.

The numerical results on quantization errors and neighborhood preservation (Table 3.2), as well as the performance with respect to precision and recall (Fig. 3.31) yield the same qualitative results as for the artificial document dataset. I.e., we observe a slight advantage of the HSOM over the SOM with respect to a higher average precision. A comparison to other state-of-the art classifiers is given at a later stage in Section 4.2.3.

[1] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

(a) STONE SPLITS STOCK, RAISES PAYOUT
Stone Container Corp said it is splitting its common stock 2-for-1 and increasing its dividend 33-1/3 pct. The dividend of 20 cts a share, an increase of five cts over the prior 15 cts a share on pre-split shares, is payable June 12 to holders of record May 22. The stock split also is payable June 12 to holders of record May 22. Reuter

(b) U.K. CLEARS CONS GOLD U.S. PURCHASE
The U.K. Trade Department said it would not refer Consolidated Goldfields Plc's purchase of American Aggregates Cors to the Monopolies Commission. Cons Gold said last month that its ARC America Corp unit had agreed to buy the Ohio-based company for 30.625 dlrs a share cash, or 242 mln dlrs, in a deal recommended by the Aggregates board. Reuter

(c) BUNDESBANK LEAVES CREDIT POLICIES UNCHANGED
The Bundesbank left credit policies unchanged after today's regular meeting of its council, a spokesman said in answer to enquiries. The West German discount rate remains at 3.0 pct, and the Lombard emergency financing rate at 5.0 pct. Reuter

(d) TAIWAN BUYS 54,000 TONNES OF U.S. SOYBEANS
The joint committee of Taiwan soybean importers awarded a contract to Cigra Inc of Chicago to supply a 54,000 tonne cargo of U.S. Soybeans, a committee spokesman told Reuters. The cargo, priced at 213.87 U.S. Dlrs per tonne c and f Taiwan, is for delivery between April 20 and May 5. Reuter

Figure 3.30: Examples of newsletter messages which have been manually tagged by Reuters personell with the labels *(a)* "earn", *(b)* "acquisition", *(c)* "money exchange rate" and *(d)* "soybean".

	$E_{q\mathcal{M}}$	ρ	T_t	C_t
SOM (2304 nodes)	0.357 ± 0.001	0.169 ± 0.018	0.661 ± 0.020	**0.686** ± 0.005
HSOM (2281 nodes)	**0.274** ± 0.004	**0.216** ± 0.015	**0.678** ± 0.029	0.661 ± 0.006

Table 3.2: Comparison of SOM and HSOM with respect to quantization error, Spearman's rho and total trustworthiness and continuity.

Visualization Issues

Fig. 3.32 shows a standard 48×48 SOM with 2304 nodes and a 8^5 HSOM with 2281 nodes. Both maps show the mapping of the complete Reuters dataset. All nodes are labelled according to their attached topic vectors (cf. Section 3.5.1). The visualization contains 20 differently shaped and colored glyphs which correspond to the most frequently occurring topic of the top 20 categories mapped to the corresponding node.

Although the HSOM outperforms the SOM in the numeric benchmarks described above, a distinctive drawback becomes apparent when considering Fig. 3.32: In the flat Euclidean space of the standard SOM the nodes are uniformly utilized, i.e. in the visualization the whole map space is evenly covered with data points. The HSOM visualization on the other hand shows a large "hole" in its center. This is caused by the peculiar geometry of hyperbolic space: the majority of nodes is located at the periphery of the map. Additionally, the outer nodes have more freedom to adapt to the high dimensional bag-of-words feature space. In this space the vast amount of data resides in the perimeter. The center is only very sparsly populated. The HSOM reflects this rather faithfully, however at the cost of visualization space which remains unused. We shall see in the next chapter how we can turn this property of hyperbolic space into an advantage also for visualization issues.

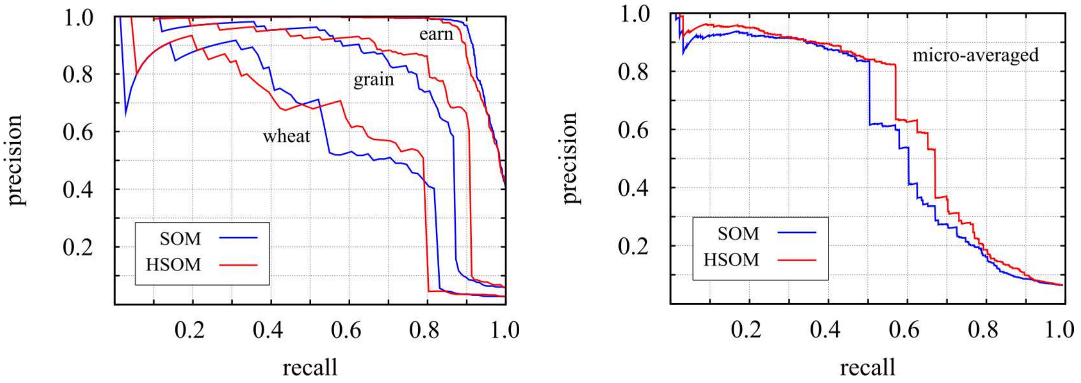


Figure 3.31: Precision over recall for the Reuters-21578 data. In (a) the curves for the categories “earn”, “grain” and “wheat” are shown (from top-right to bottom-left), in (b) the micro-averaged results over all categories.

3.6 Summary

In this chapter we have introduced the hyperbolic self-organizing map (HSOM). In principle, the HSOM operates in exactly the same way as the self-organizing-map as introduced by Kohonen more than 20 years ago. Its geometric properties, however, are fundamentally different: While the standard SOM uses the flat Euclidean space as the geometrical substrate for the construction of a regular lattice of formal neurons, the HSOM utilizes non-Euclidean hyperbolic space. The peculiar geometric properties of the hyperbolic plane $I\!H^2$ offer two distinctive advantages: First, the neighborhood around any point in that space grows asymptotically exponentially, allowing the formal neurons to “feel” more freedom for adaptation. Second, the projection of $I\!H^2$ onto the two-dimensional Poincaré Disk allows for a natural *focus & context* navigation scheme within very large map spaces.

Based on an intuitively graspable dataset we have demonstrated that for high-dimensional data, the HSOM outperforms the standard SOM with respect to quantization errors and global and local neighborhood preservation. In order to apply the (H)SOM for text categorization tasks we have developed a node labelling and ranking scheme allowing for a retrieval of documents ordered by their relevance to a desired given topic. By constructing an artificial bag-of-words dataset we have shown how a controlled dataset mirroring the statistical properties of large document collections can be used as a baseline for performance comparisons. Both the artificial dataset and the Reuters-21578 newswire benchmark established a slightly superior performance of the HSOM as compared to the standard SOM with respect to quantization error, neighborhood preservation and classification accuracy. The visualization capabilities of the HSOM, however, have turned out not to be beneficial. Because the HSOM reflects the statistical properties of the very high-dimensional text data rather well, the user is initially confronted with mainly empty space on the HSOM. Therefore, forcing the user to exhaustively navigate in $I\!H^2$ in order to be able to examine the whole dataset. This drawback is addressed in the following chapter.

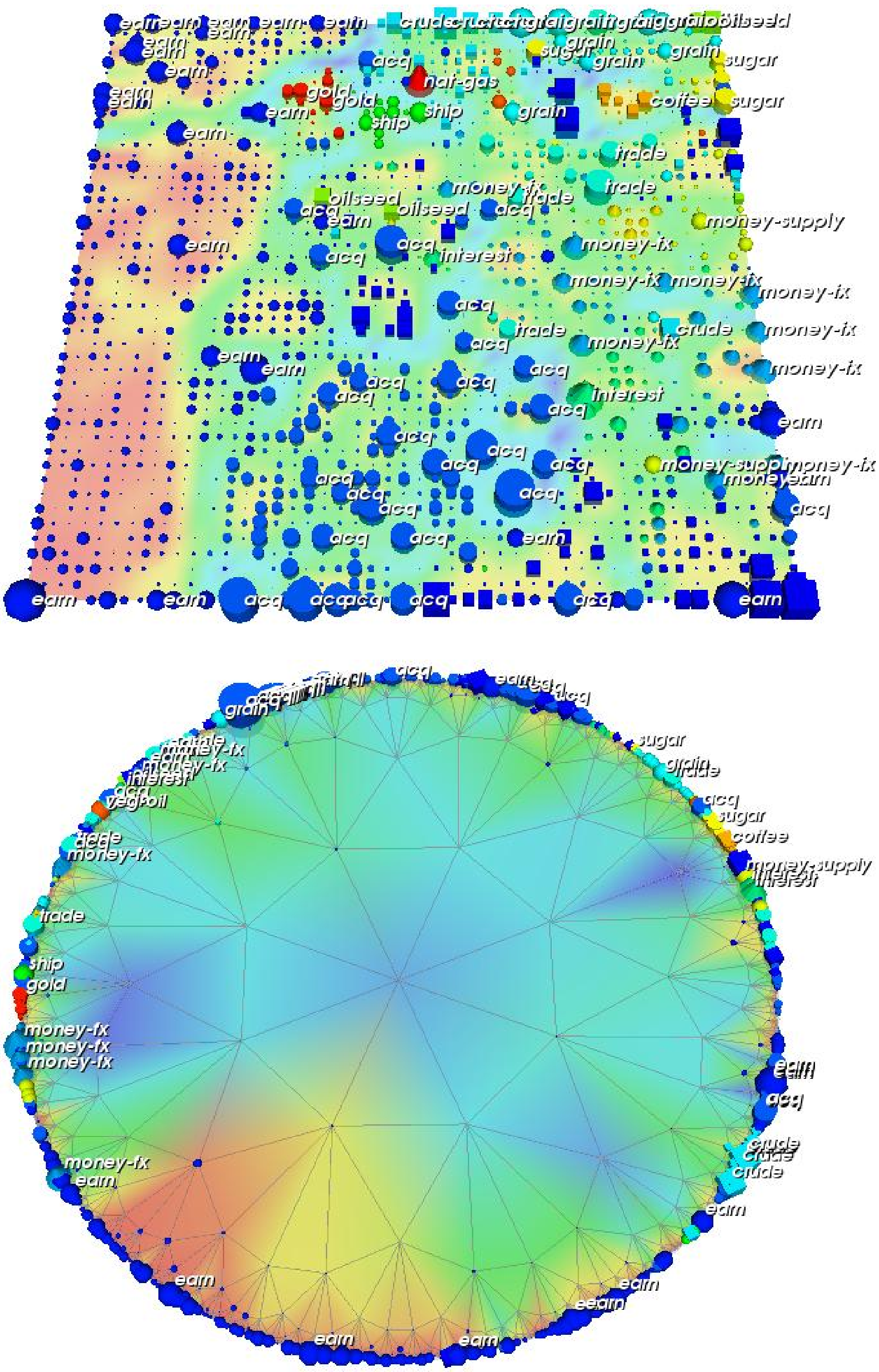


Figure 3.32: Visualization of the Reuters-21578 corpus with a 48×48 SOM and a 8^5 HSOM. The ground color indicates the average distance of the prototype vectors to their neighbors, and associated topics are coded by differently shaped and colored glyph objects.

Chapter 4

Learning Hierarchies in Hyperbolic Space

4.1 The Hierarchically Growing Hyperbolic SOM (H^{2}SOM)

The previous chapter has shown how to utilize the non-Euclidean hyperbolic plane $I\!H^2$ as an alternative map space for the SOM algorithm. Though the fundamental geometric properties of the mapping space were changed, the HSOM still performs a direct mapping, i.e. single data items from the high-dimensional feature space are mapped to a single node on the two-dimensional hyperbolic map.

Therefore, a central parameter affecting the resolution of a HSOM is the area of its map size. In order to represent very large datasets faithfully, the map area needs to be large enough to allow for small quantification errors (cf. Eq. 3.36 and 3.38). However, for large maps, we have to deal with two distinctive drawbacks:

1. In case of large HSOMs, the majority of nodes is located at the periphery of the network - caused by the asymptotically exponential scaling behavior of $I\!H^2$. The user is therefore forced to exhaustively navigate in $I\!H^2$ in order to examine the whole dataset. An example of such a situation is given in Fig. 3.17(b) and (c) and Fig. 3.32.

2. With linearly increasing map area, the number of nodes in a SOM increase quadratically. Since both the SOM and HSOM algorithm scale linearly with the number of nodes (Kohonen, 2001), training of large map areas can be computationally very expensive.

Several approaches have been suggested to address this computational problem arising for large map areas. Koikkalainen and Oja (1990) proposed the Tree-Structured Self-Organizing Map (TS-SOM), which consists of a fixed number of SOMs arranged in a pyramidal structure. The training of the pyramid is computed level-wise where the best match search is performed as a tree search reducing the complexity to $\mathcal{O}(log\ N)$. A Growing Hierarchical SOM (GHSOM) has been proposed by Rauber et al. (2002). Their approach combines individually growing SOMs with a hierarchical architecture and has successfully been applied to the organization of document collections and music repositories. Lately, Pakkanen et al. (2004) have described the Evolving Tree, which is constructed as a freely growing network utilizing the shortest path between two nodes in a tree as the neighborhood function for the self-organizing process. All of these approaches achieve a favorable computational complexity. However, the visualization of the learned hierarchies remains a demanding task.

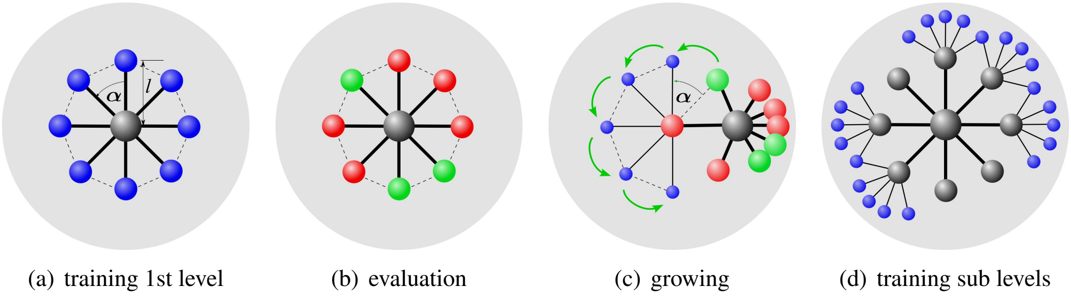
Either a map metaphor is not applicable anymore, or the transition between maps within or across the hierarchies introduces discontinuities making it hard to visualize and maintain the surrounding context. Thus, without guidance the user might be easily lost within the tree structure. Lamping and Rao (1994) discovered that hyperbolic space is ideally suited to embed large hierarchical structures. Their discovery motivated the introduction of the HSOM as introduced in the previous chapter. However, due to the exponential growth of its hyperbolic lattice, it also exacerbated the need for addressing the scaling problem of SOMs comprising very large numbers of nodes.

In the following sections we show that a solution can be achieved by a very natural extension of the HSOM to a *Hierarchically Growing Hyperbolic SOM* (H^{2}SOM).

4.1.1 Network Topology

The core idea of the hierarchically growing Hyperbolic Self-Organizing Map is to employ the same sort of lattice structure already used for the plain HSOM.

***(i)* Network Initialization:** We start with the root node of the hierarchy placed at the origin of $I\!\!H^2$ and choose the *branching factor* n_b. It determines how many child nodes a node may grow and how "fast" the network is reaching out into hyperbolic space. Since we employ the same lattice structure as already used for the HSOM, the comments of Sec. 3.3.1 also hold for the H^{2}SOM , i.e. the branching factor has a lower bound of $n_b > 6$. The nodes of the first hierarchical level are positioned along an initial lattice structure which is also used as the "first ring" for the HSOM (cf. Fig. 3.12(a)). For the case of $n_b = 8$ the resulting configuration is shown in Fig. 4.1(a).



(a) training 1st level (b) evaluation (c) growing (d) training sub levels

Figure 4.1: Architecture and learning procedure of the H^{2}SOM.

4.1.2 Training and Growing Procedure

After initializing the network, the training of the hierarchical structure largely follows the traditional SOM approach.

***(ii)* Training of first level:** The center node is initialized with the center of mass of the training data and does not take part in the training process. Its prototype vector stays fixed. The n_b nodes of the first hierarchical level are initialized with small random variations of the center prototype and are trained in the usual way: After finding the best match neuron a^*, i.e. the node which has its prototype vector $\mathbf{w}_a$ closest to the given input $\mathbf{x}$, $a^* = \text{argmin}_a \|\mathbf{w}_a - \mathbf{x}\|$, all reference vectors of the first level are updated by the standard adaption rule given by Eq. 3.2. For the neighborhood function $h(a, a^*)$ the hyperbolic distance of the nodes in the PD is used (Eq. 3.33). During the course of learning, the width $\sigma(t)$ of the neighborhood function and the learning step size $\epsilon(t)$ are continuously decreased. This allows for more and more specialization and fine tuning of the then increasingly weakly coupled neurons - just as in the standard SOM approach.

***(iii)* Evaluation and Growing:** After fixed training intervals for each node an expansion criterion is evaluated. In all experiments reported below, we use the node's quantization error as given by Eq. 3.37 as the growth criterion. If a given threshold Θ_{QE} for a node is exceeded it is marked for growing (as indicated by a red node in Fig.4.1(b)). Subsequently, all marked nodes are expanded by surrounding it with $n_b - 3$ children nodes (minus 3 because there are already two sibling and one parent node present at this stage). Algorithmically this can be done by applying a Möbius transformation such that the to be expanded node resides in the center of the *PD*. As an example, in Fig. 4.1(c) the leftmost marked node was translated to the center of the map. The coordinates of the child nodes are then obtained by iteratively applying the Möbius transformation $M_{c,\varphi}(z)$ with $c = 0$ and $\varphi = \cos(\alpha) + i\sin(\alpha)$ to one of the sibling nodes as indicated by the five arrows in Fig. 4.1(c).

Note, that a value of $\Theta_{QE} = 0$ might be chosen to achieve an expansion of every node within a hierarchical level. The growing process is then stopped if a prescribed depth within the tree is reached.

***(iv)* Reiteration:** After the growing step where all nodes meeting the growth criterion were expanded, all reference vectors from the previous hierarchies become fixed as indicated by the gray nodes in Fig. 4.1(d). The adaption then "moves" to the nodes of the new structural level and the learning process is reiterated.

4.1.3 Fast Tree Search for finding Best Match Units

During training, the most time consuming step in a standard SOM and the HSOM is the global search for the best match unit. The peculiar, intrinsically "uniformly hierarchical" structure of the hyperbolic grid offers an intriguing possibility to significantly accelerate this most time-consuming step: We can approximate the global search for the winner unit a^* by *a fast tree search*, taking as the search root the initial center node of the growth process and following the "natural" hierarchical structure in the hyperbolic grid: starting from the center node, we recursively determine the k best-matching nodes among its n_b neighbors until we reach the periphery. For $k = 1$, this will generate a path with $\mathcal{O}(\log_{n_b} N)$ node visits as shown in Fig. 4.2(a), instead of $\mathcal{O}(N)$ for a global search.

By choosing $1 < k \leq n_b$ we perform a *beam search*, a classical AI tree traversing algorithm commonly used as a heuristic method for solving combinatorial optimization problems (Russel and Norvig, 2003). The width of the search beam is determined by k. It covers a search space of asymptotically $\mathcal{O}(N^p)$ nodes, with exponent $p = \log_{n_b} k \leq 1$. Fig. 4.2(c) shows the search path for $k = 2$. Note, that by choosing $k = n_b$, p equals to one, resulting in a full search with $\mathcal{O}(N)$.

As Fig. 4.2(a) suggests, a value of $k = 1$ leads to a very narrow search space. As a compromise between rapid searching and wider exploration of the search space we also report results on a search scheme we call "SF-search"[1]. Here we choose a tree branching factor of $k > 1$ for the first structural level and truncate it to $k = 1$ for all search steps

[1] SF stands for "super-fast"

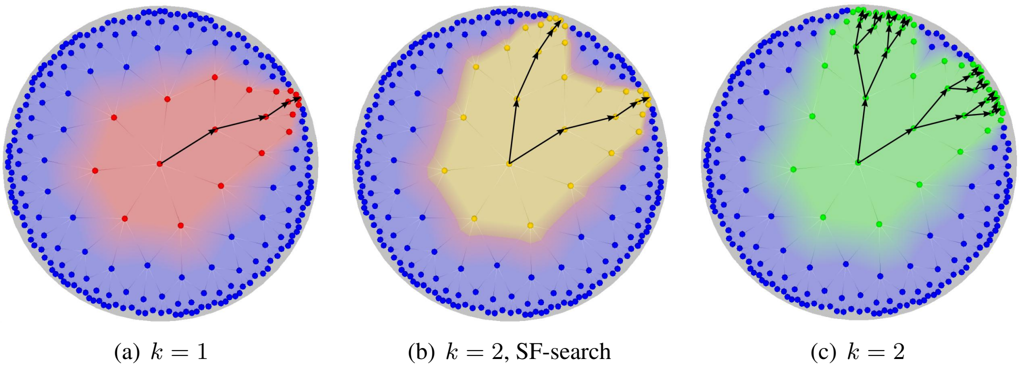


Figure 4.2: The tree search within the hierarchical structure of the H^{2}SOM significantly reduces the search space for finding the best match unit. The colored regions correspond to the nodes visited during the search for different values of k.

beyond. This results in an exploration of the search space into different directions, but still scales very favorably with $\mathcal{O}(k \cdot \log_{n_b} N)$ (cf. Fig. 4.2(b)).

The resulting computational complexities for different choices of k in comparison to a full global search are given in Fig. 4.3. For very large maps, the yielded scaling behavior permits speed-ups of several orders of magnitude, as compared with a global (standard SOM) search.

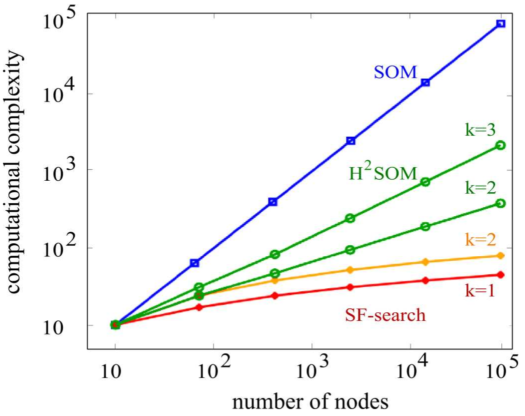


Figure 4.3: Computational complexity for the best match search in SOM vs. H^{2}SOM ($n_b = 10$).

Accuracy of Fast Tree Search

As seen above, the fast tree search is able to significantly reduce the computational costs of the self-organizing process. We now address the question how accurate the accelerated search performs. To this end we train maps with four different beam search parameters and compare how often the beam search finds the true best match unit obtained by global search.

The numbers presented in Fig. 4.4 are based on 5000 test samples from the artificial bag-of-words dataset as described in Sec. 3.5.2. The results show that the two fast variants already produce reasonable results: In 89.5% (92.6%) of the cases, the search with $k = 1$ ($k_{SF} = 2$) finds the true best-match node or its neighbor. The two wider branching searches with $k = 2$ and $k = 3$ achieve a direct or neighboring hit of the global winner in 96.9% and 97.7% of all cases.

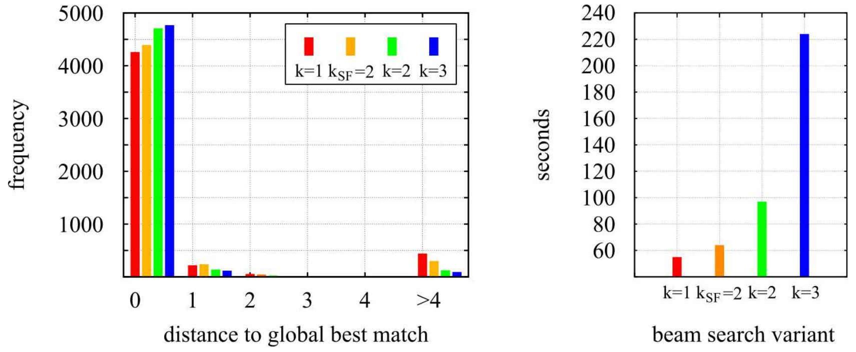


Figure 4.4: Accuracy and training times for the fast tree search: The histograms show for different beam search parameters k how many nodes are found for a given distance from the true, global best match unit (left). The figure reads as follows: For example, for $k = 1$ in approx. 4300 from 5000 cases the true best match node is found, but in 500 cases the node found is more than 4 units away from the global best match node. Additionally, the training times for the different variants are given on the right.

Naturally, the better results have to be paid by longer search times as shown in Fig. 4.4 above on the right. As the figures suggest, the beam search with a width of $k = 2$ achieves a good compromise between training time and accuracy. The following benchmarks of the H^{2}SOM with different datasets strengthen this proposition.

4.1.4 Benchmarking the H^{2}SOM

For the results reported here we have chosen a branching factor of $n_b = 8$ and an expansion criterion of $\Theta_{QE} = 0$. The growing process was stopped for hierarchical levels larger than five, resulting in map sizes of 2281 nodes in total. Since the map is build in a hierarchical fashion, the first 8 nodes constitute the first structural level of the data. The nodes of the outer rings then build up the hierarchy "ring by ring" (cf. Fig. 4.1). Consequently, the 2281 nodes are representing the 5 levels as follows:

	level 1	level 2	level 3	level 4	level 5
nodes	1-8	9-40	41-160	161-608	609-2280
number of nodes	8	32	120	448	1672

Table 4.1: Number of nodes in each hierarchical level.

For the H^{2}SOM benchmarks we have evaluated the performance indicators quantization error, Spearman's rho, trustworthiness and continuity for the hierarchical levels 3, 4 and 5, allowing a comparison with similar sized HSOMs as shown below.

Artificial Bag Of Words with 20 Clusters

Quantization error. We first look at the artificial bag-of-words dataset as described in Sec. 3.5.2. The results in Fig. 4.5 show that a H^{2}SOM trained with a beam width of $k = 3$ achieves the best quantization errors comparable to those of the HSOM. With decreasing beam width, the quantization error increases, but not dramatically.

Global and local ordering. The figures on Spearman's rho (Fig. 4.5) show an unexpected outcome: A H^{2}SOM trained with a broader beam search ($k = 3$) achieves significantly worse global ordering capabilities than those trained with smaller values of k. The same observation also holds for the local ordering measured by trustworthiness and continuity as shown in Fig. 4.6.

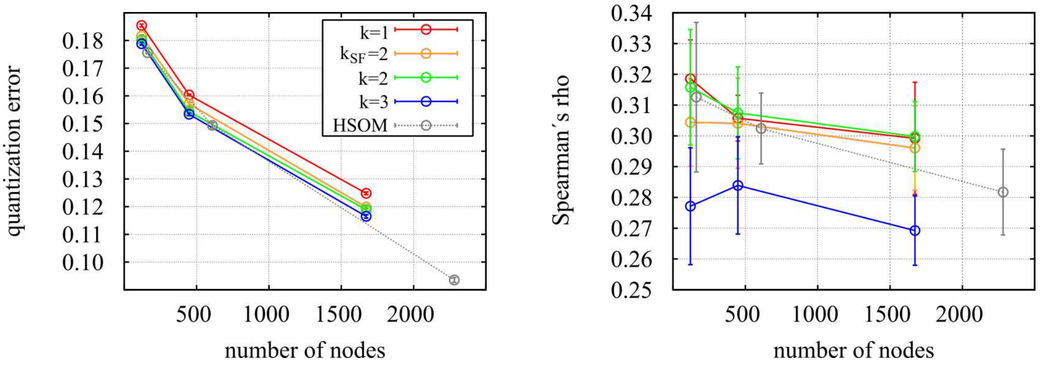


Figure 4.5: Quantization errors (left) and Spearman's rho (right) for the artificial bag-of-words dataset.

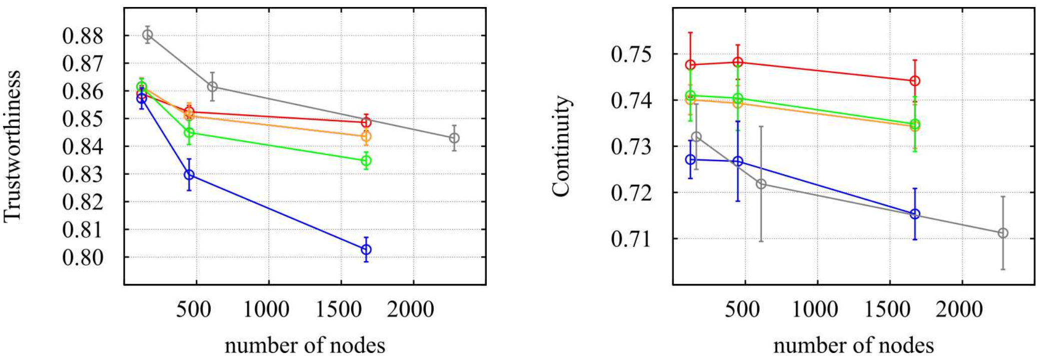


Figure 4.6: Trustworthiness (left) and Continuity (right) for the artificial bag-of-words dataset.

To gain an insight, why the ordering capabilities of the H^{2}SOM decrease with growing beam search width k, consider the limit for $k = n_b$: In this case we would perform a full search within each ring. The outcome of this would be equivalent if we would train each ring of the H^{2}SOM separately, i.e. the H^{2}SOM would consist of concentrically layered one-dimensional rings with no connection from ring to ring. This would lead to better performance with respect to the quantization error but at the cost of the topology preserving capabilities as the results presented in Fig. 4.5 and Fig. 4.6 strongly suggest. The hierarchical structure which is build during the training of the H^{2}SOM with smaller beam widths seems to be important to obtain a topological well ordered map.

Classification accuracy. Performance results with respect to precision and recall are shown in Fig. 4.7. For all beam widths the H^{2}SOM generally achieves a higher precision at any given recall level than the plain HSOM, whereupon the differences are more pronounced at higher recall levels: For example, at a recall level of 2/3 the micro-averaged precision is around 75% for the H^{2}SOM , whereas the plain HSOM achieves only around 55%.

Precision seems to increase with beam width, however the difference is not as pronounced as the difference between H^{2}SOM and HSOM.

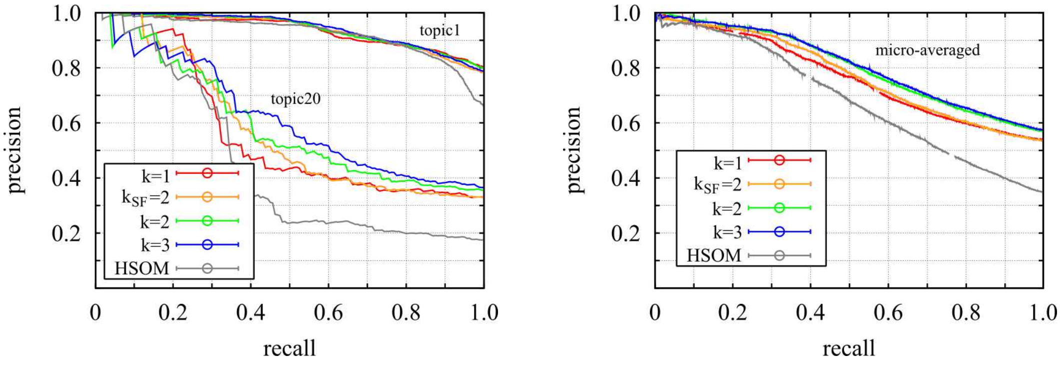


Figure 4.7: Precision over recall for the artificial document dataset. In (a) the curves for the categories 1, 10, and 20 are shown, in (b) the micro-averaged results over all categories.

Reuters-21578 Newswire Articles

Quantization Error and Neighborhood Preservation. The numerical results on the Reuters-21578 dataset are given in Table 4.2. The numbers indicate that the plain HSOM achieves lower quantization errors on the dataset. For $k = 2$ the difference in terms of $E_{q\mathcal{X}}$ (cf. Section 3.4.1) is about 1.5%, in terms of $E_{q\mathcal{M}}$ about 23%. This discrepancy can be explained by the different number of nodes available to each SOM: At the finest resolution the H^{2}SOM consists of 1672 nodes (see Table 4.1), whereas the HSOM operates on 2281 nodes – a difference of about 36%.

With respect to global and local topological preserving capabilities the H^{2}SOM outperforms the HSOM. As discussed above, trustworthiness and continuity do not benefit from larger search beam widths. For $k = 2$ the results indicate a good compromise between quantization errors and topology preservation.

	$E_{q\mathcal{M}}$	$E_{q\mathcal{X}}$	ρ	$\sum T(k)$	$\sum C(k)$	t_{train}
H^{2}SOM $k = 1$	0.349	0.425	0.282	**84.53**	**80.15**	12min 57s
H^{2}SOM $k_{SF} = 2$	0.344	0.417	0.272	83.18	79.73	14min 10s
H^{2}SOM $k = 2$	0.338	0.406	**0.287**	82.79	79.64	15min 14s
H^{2}SOM $k = 3$	0.335	0.402	0.266	78.78	77.22	18min 50s
HSOM	**0.274**	**0.400**	0.216	79.72	75.43	6h 39min

Table 4.2: Comparison of H^{2}SOM and HSOM for the performance measures quantization error and Spearman's rho.

The major difference between H^{2}SOM and HSOM becoming apparent from Table 4.2 is the decidedly decreased training time. A speedup of a factor of more than 25 decreases the training time from more than six hours to approx. 15 minutes, almost allowing for an online analysis of the Reuters-21578 dataset.

Classification accuracy. The results of Fig. 4.8 show a similar tendency as for the artificial bag-of-words dataset: The H^{2}SOM tends to keep up a higher precision than the plain HSOM.

Visualization capabilities

Fig. 4.9 shows the visualization of a hierarchically trained 8^5 H^{2}SOM. In contrast to the plain HSOM shown in Fig. 3.32, the H^{2}SOM also makes use of the center of the visualization display.

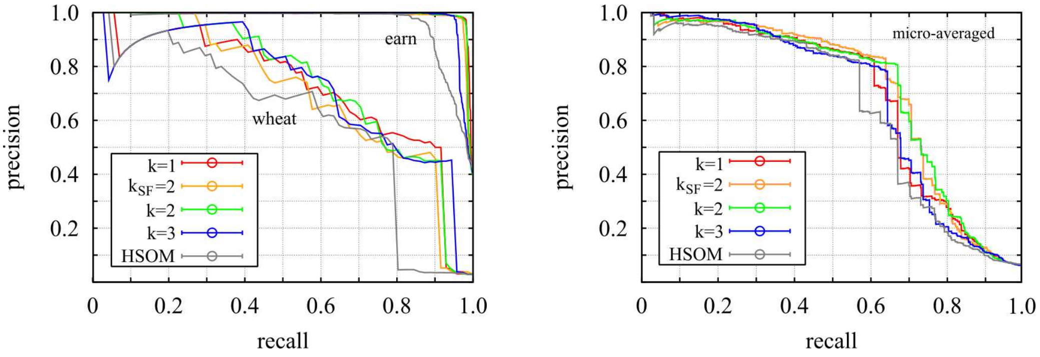


Figure 4.8: Precision over recall for the Reuters-21578 data. In (a) the curves for the categories “earn” and “wheat” are shown (from top-right to bottom-left), in (b) the micro-averaged results over all categories.

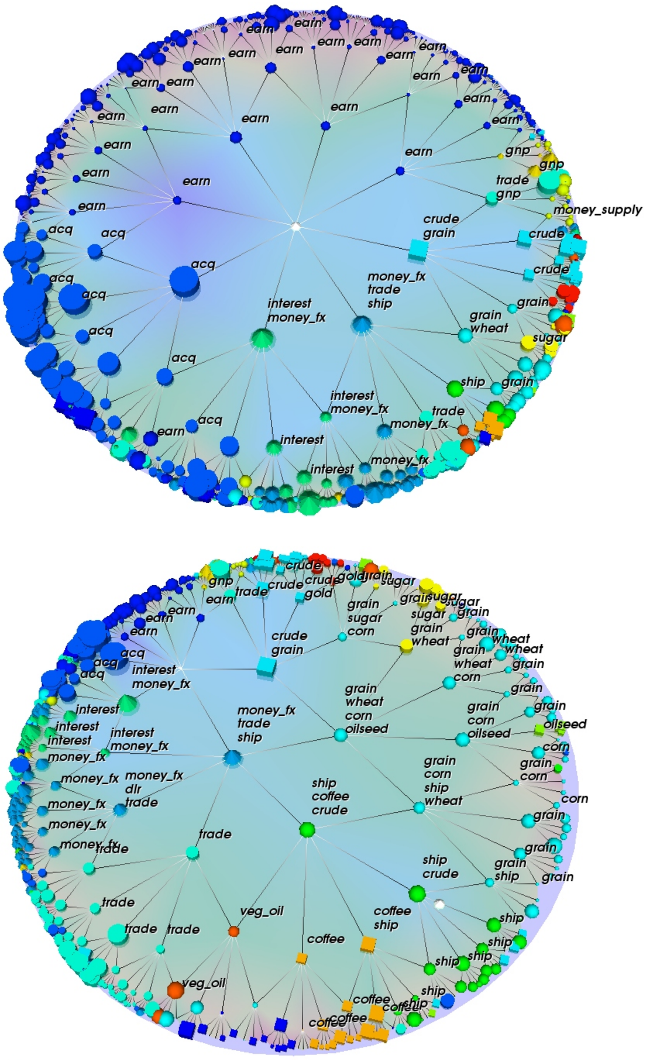


Figure 4.9: Visualization of the Reuters-21578 corpus with a 8^5 H^{2}SOM . At the top the focus (illustrated by a white cone) is centered, at the bottom the focus was moved towards the 4 o’clock position to magnify the less dominant topics within the dataset.

The first inner eight nodes represent the whole dataset. From the labels which were automatically obtained from the topic vectors (cf. Section 3.5.1) the user is able to quickly grasp the topic distribution in Reuters-21578 when looking at these eight nodes: Approximately half of the data is labelled with “earn” and “acquisition”, whereas the topics “interest”, “money_fx”, “trade”, “ship”, “crude” and “grain” are represented by only three nodes.

In the bottom image of Fig. 4.9 the user has moved the focus into the 4 o’clock direction. By doing so he enters the deeper hierarchical levels of the dataset which correspond to the less frequent topics. In this screenshot the prominent topics “earn” and “acquisition” only occupy a small region in the top left of the map, giving the user the possibility to explore the remaining content in more detail.

A more thorough explanation of the visualization engine and its application can be found in Chapter 5. Two user studies in Chapter 6 give additional insights on the benefits of the hierarchical hyperbolic approach.

MNIST

In order to assess the performance of the H^{2}SOM on a common non-textual dataset, we have applied the H^{2}SOM to the MNIST database of handwritten digits.

The MNIST database[1] consists of 60.000 training samples from approximately 250 writers and 10.000 test samples from a disjoint set of 250 other writers. We used the original 784-dimensional dataset which resembles 28x28 pixel grey level images of the handwritten digits. Since we used the dot product as our data metric, all samples were normalized to unit length. We have trained four standard SOMs of the sizes 7x7, 13x13, 25x25 and 48x48 with 49, 169, 625 and 2304 nodes, respectively. In comparison we have trained five H^{2}SOMs with a branching factor of $n_b = 8$. As a termination criterion we used a combination of maximal depth and quantization error: The growth process was stopped when either a predetermined hierarchy level was reached (in our case 2, 3, 4, 5 or 6 rings with maximal 41, 161, 609, 2281 or 8521 nodes, respectively), or a node's quantization error was less than a third of its parent's quantization error. In all cases 600.000 training steps were performed and the given results were averaged over 10 runs (except for the large SOM which was just trained twice due to the long computing times).

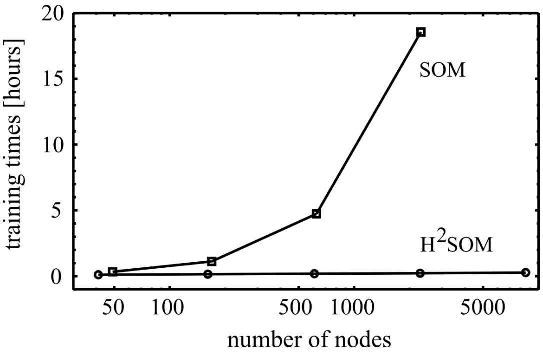


Figure 4.10: Training times for different sized SOMs and H^{2}SOMs for the MNIST database. Note, that the abscissa is drawn with a logarithmic scale.

Fig. 4.10 shows the training times for computing the maps. From the graph the favorable scaling behavior of the fast best match search in the H^{2}SOM becomes evident: even very large maps are trained within a few minutes, while standard SOMs quickly take several hours to complete.

We additionally applied the SOMs as a classification tool for classifying the handwritten digits of the MNIST test dataset. To this end, the labeled training data is mapped to the SOM and all nodes are labeled with the most frequent label of the training items mapped to it. If there is no training item mapped to a node, i.e. the node is an interpolating node, it is labeled according the majority of votes from the neighborhood on the lattice grid. To each test item then the class label of its corresponding best match node is assigned.

In Table 4.3 the classification accuracies for different SOMs are given. Again, the most prominent difference is the time needed for the training of the networks. Due to the high data dimensionality ($d = 784$) the large SOM took more than 18 hours to compute, while the large H^{2}SOM using the "super-fast" SF-search was finalized in only 16 minutes. Despite using a full search for the SOM during training, the H^{2}SOM achieves a better mean quantization error. When using the SOMs as a classification tool, we used (a) the

[1] `http://yann.lecun.com/exdb/mnist/`

	SOM		H^{2}SOM, $n_b = 8$				
	13x13	48x48	3 rings		5 rings		6 rings
nodes	**169**	**2304**	**161**		**2281**		**8521**
QE	0.2094	0.1510	0.1993		0.1441		0.1175
t_{train}	1:07h	18:34h	0:09h		0:13h		0:16h
t_{test}	7.8s	181s	(a) 1.8s	(b) 8.4s	(a) 3.0s	(b) 101s	(b) 514s
Class	classification performance [%]						
0	93.9	**98.3**	96.0	**98.1**	**98.3**	**99.2**	99.5
1	**98.3**	**98.6**	**98.3**	98.1	98.5	98.5	99.1
2	86.6	**94.6**	**89.1**	**92.4**	92.4	93.1	94.7
3	**80.2**	91.3	76.1	79.5	90.0	**92.7**	94.6
4	69.0	88.3	**73.2**	**76.3**	**90.4**	**93.6**	94.4
5	66.9	90.0	**83.5**	**89.1**	87.4	**92.5**	93.1
6	**93.9**	**97.1**	89.7	92.7	96.0	96.3	97.7
7	81.2	91.0	**81.7**	**85.9**	**91.4**	**92.8**	93.9
8	**76.4**	88.8	59.1	67.6	88.1	**90.9**	91.8
9	**59.8**	88.0	55.9	57.8	**88.3**	87.7	90.7
total	81.0	92.7	80.5	**85.3**	92.2	**94.4**	95.8

Table 4.3: Comparison of the H^{2}SOM to similar sized standard SOMs. The table shows the training times in hours and minutes for the map formation of the 60000 training samples and the seconds for the best match lookups for the 10000 test samples of the MNIST database. For the H^{2}SOM the test runs were performed with (a) the rapid SF-search with $k_{\mathrm{SF}} = 2$ and (b) a slower global search. (All results were obtained on a standard laptop with 1.5 GHz Pentium-M processor).

SF-search with $k_{\mathrm{SF}} = 2$, and (b) a slower global search to find the best match nodes for the 10.000 test samples. In the first case, the overall performance of the SOM is 0.5% better, though for half of the classes the H^{2}SOM achieves the same or better results. When using the slower global search only for retrieval *after the fast training* of the H^{2}SOMs, the classification performance for the latter becomes considerably better and the H^{2}SOM now clearly outperforms the SOM. The last column shows the results for a large H^{2}SOM with 8521 nodes (it does not have a SOM counterpart, since it would have taken too long to compute). In terms of quantization error and classification accuracy, the results for this very large H^{2}SOM are superior without investing significantly more time in training the network.

Visualizing the MNIST database. Turning to the visualization capabilities of the H^{2}SOM we show in Fig. 4.11 a H^{2}SOM with a branching factor of $n_b = 12$. In (a) the Poincaré Disk is shown in a centered view, such that the top-level structure of the dataset is visible as the innermost ring of nodes. The prototype vectors are overlaid as textures on the node's glyphs. The colors are just a visual hint to indicate the class to which the majority of training samples belong to in the corresponding region of the map.

The H^{2}SOM can be seen to have learned the following top level structure from the data: The upper three inner nodes resemble mixtures between "4"s, "9"s and "7"s. Clockwise follows a node with a prototype looking like a blurred slanted "9", then two differently orientated "1"s follow. At the bottom, three prototypes similar to an "8", a "3" and a "5" are shown, and then an articulated "0", a "2" and a "6" appear. In Fig. 4.11(b) the user has moved the focus towards the one o'clock node which is now centered. Here it can be seen, that at this next structural level the data splits up into equally slanted "7"s at the top, "9"s to the right and "4"s at the bottom right of the map.

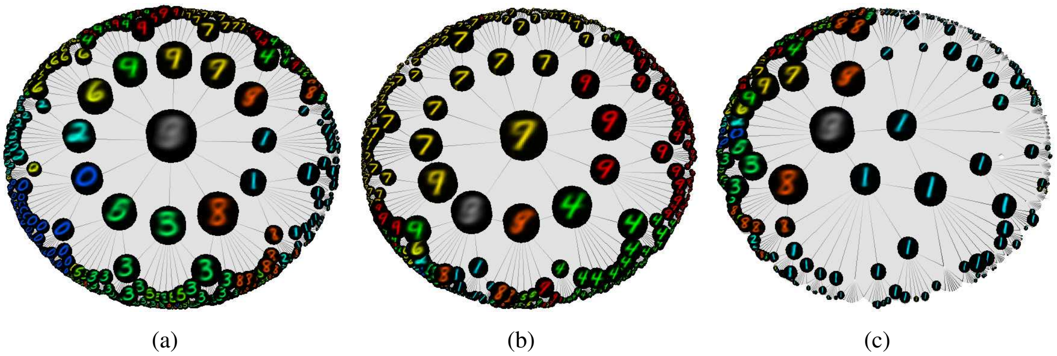


Figure 4.11: Screenshots from different focus positions in the MNIST database. (a) shows the overall coarse structure of the dataset is shown, in (b) the user has moved the focus to the "7" node from the 1 o'clock position in (a). In (c) the focus of attention was moved to the area covering the "1". Here several nodes were not expanded, because the low variation of the data resulted in low quantization errors of the nodes.

4.2 Hierarchical Feature Organization

As we have outlined in Section 4.1.4, the H^{2}SOM is very well suited to represent and visualize large data sets in a hierarchical manner. For the more complex task of text classification we have shown that the hierarchical approach leads to a performance increase with respect to information retrieval specific measures such as precision and recall when compared to a flat HSOM.

For an even further exploitation of the natural hierarchical structure of the H^{2}SOM we propose the employment of a hierarchical feature organization. All experiments so far have been conducted with flat feature spaces, i.e., each hierarchical level of the H^{2}SOM was trained with the same set of features from the input space $\mathcal{X}$.

The hierarchical levels of the H^{2}SOM order the data at different structural levels: The innermost ring with only a few neurons represents the data at a rather coarse resolution. As we go deeper into the hierarchies of the H^{2}HSOM, the more fine grained the resolution of the mapping becomes. A hierarchical representation of text data might therefore lead to further improvements with respect to text classification or information retrieval tasks in conjunction with the H^{2}SOM: If it would be possible to represent text data at different "semantic resolutions", we could use these different resolutions for the training of the different hierarchical levels of the H^{2}SOM. Ideally, the benefit would be twofold: First, a representation with a coarse semantic resolution would need significantly fewer dimensions than a text representation covering every subtle meaning of the thousands of words in a language, i.e., fewer dimensions mean less training time and therefore reducing the computational costs. Second, a coarser semantic resolution during the initial construction of the top hierarchical levels of the H^{2}SOM might lead to a better generalization capacity of the resulting model.

In the next section we therefore suggest to extend the flat *bag-of-words* to a hierarchically organized *pyramid-of-words*. The name is borrowed from the field of computer vision, where hierarchical approaches to represent image data have been successfully applied for more than 20 years (Burt and Adelson, 1983; Gonzalez and Woods, 2008). The term *image pyramid* is commonly used to refer to the pyramidal structure for representing images at different resolutions.

4.2.1 The Pyramid of Words

For images, there exist a wide variety of different approaches to represent the data in a hierarchical pyramidal way (Gonzalez and Woods, 2008). A straight forward manner is to represent images at different resolutions as depicted by Fig. 4.12.

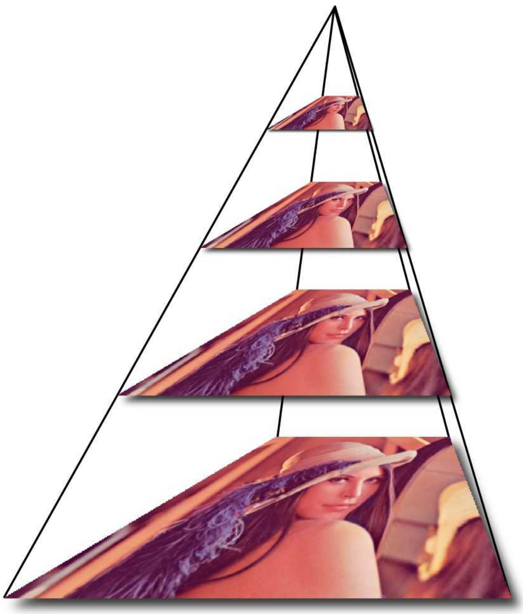


Figure 4.12: A straight forward example for the hierarchical representation of an image pyramid.

The standard approach to represent unstructured text data is the flat *bag-of-words* (refer to Section 2.2.1) which we have also been following so far. A naive approach to reduce the dimension of the bag of words in a hierarchical manner would be the utilization of differently sized dictionaries depending on the semantic resolution we would like to look at. At the finest resolution the representation of a text document would use a very high dimensional vector where the components represent every distinct word occurring within the whole document database. At coarser resolutions a smaller dictionary only covering the most “important” words appearing in the collection would describe the documents with fewer dimensions. For the feature selection process deciding which components, i.e. words, go into the dictionary at each level, standard techniques from classical information retrieval would apply. However, our experiments indicate that most document collections are not suited for such a hierarchical dimensionality reduction technique: The bag of words is an inherent sparse representation. By reducing the size of the underlying dictionary the chance that single documents are represented by the zero vector is drastically increased, i.e. at the top level of the pyramid a large proportion of the documents would be represented by the zero vector.

A different approach to reduce the dimensionality of the bag-of-words is *latent semantic indexing* (LSI) which first has been introduced by Deerwester et al. (1990). Since then quite a few papers have been published on LSI (Hofmann, 2001; Shima et al., 2004; Efron, 2007) – just to name a few. Many works have shown that LSI *“succeeds in keeping or improving slightly the classification performance in a low dimension”* (Liu et al., 2004). However, the computational costs to perform the singular value decomposition (SVD) necessary for LSI is non-neglectable.

Incorporating WordNet’s Semantic Lexicon

Though LSI is a promising candidate, we concentrate on a new approach which has not been as extensively studied as LSI, but opens a new direction also very favorable from a computational point of view. The lexical database WordNet (Fellbaum, 2001) contains a vast amount of semantic information which has been compiled by lexicographers and linguists.

In WordNet each database entry is represented as a *synset*[1]. For each synset a *gloss* describes the semantic concept the synset represents. For example, the synset *(shop, store)* is described by the gloss "a mercantile establishment for the retail sale of goods or services". WordNet's main contribution is the network between synsets which describes the semantic relationship between connected synsets. In the following we make extensive use of the *hypernym* relation, which can be regarded as a "is a kind of" relation. I.e., a word A is a hypernym of word B, if A's meaning embraces the meaning of B. In our example WordNet defines the following hypernym tree for the synset *(shop, store)*:

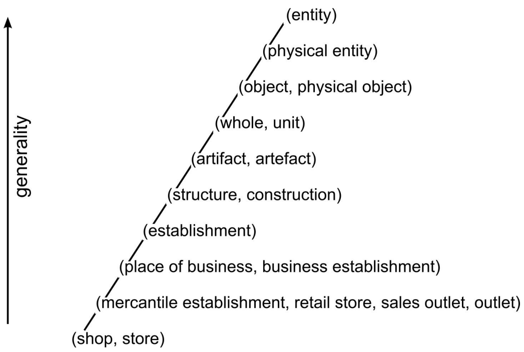

generality
(entity)
(physical entity)
(object, physical object)
(whole, unit)
(artifact, artefact)
(structure, construction)
(establishment)
(place of business, business establishment)
(mercantile establishment, retail store, sales outlet, outlet)
(shop, store)

Figure 4.13: The WordNet hypernym tree for the synset (shop, store).

The basic idea to construct our hierarchical document representation is as follows: Each component - i.e., word - of the bag-of-words is looked up within WordNet. Each feature vector can then be rewritten at different hierarchical levels corresponding to the levels of WordNet's hypernym tree. Before we give a more formal description, consider the example shown in Fig. 4.14:

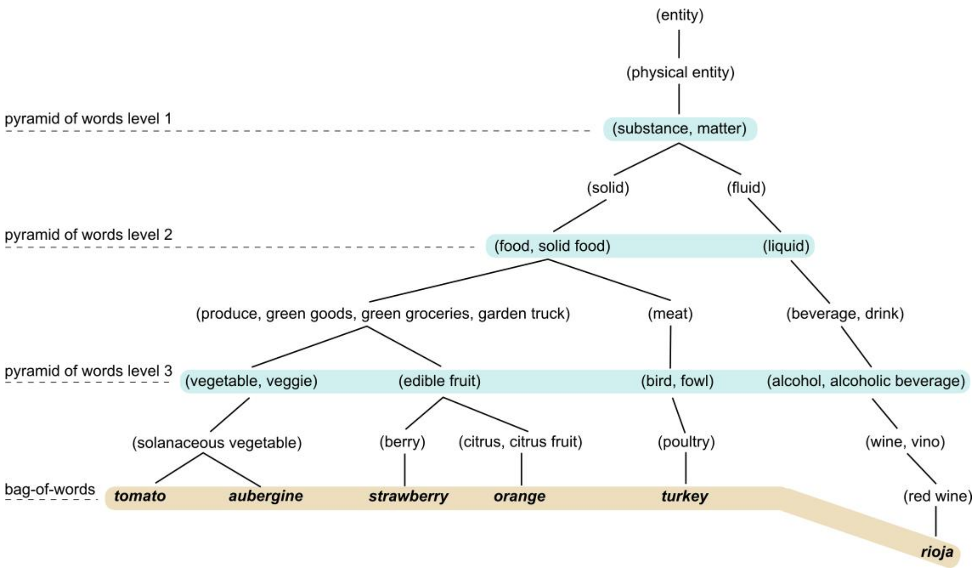


Figure 4.14: An example for the pyramid-of-words: The lower row of words *tomato, aubergine, strawberry, orange, turkey* and *rioja* correspond to the entries in the standard bag-words-model. At the *semantic level 2* of the pyramid-of-words these entries are represented by the WordNet synsets *(food, solid food)* and *(liquid)*.

For simplicity, we look at a very short document (which might be a recipe) consisting of the words *tomato, aubergine, strawberry, orange, turkey* and *rioja*. In a standard

[1] *synset* stands for *synonym set*

bag-of-words representation we need six dimensions to represent the semantics at single word level. By incorporating WordNet's hypernym relations we are able to reduce the six dimensions to four at the third semantic, and to two at the second semantic level as indicated in Fig. 4.14. As discussed before, the benefits are twofold: First, we need fewer dimensions to represent the content, second we are able to generalize at a higher level. If we consider a second document such as *potato, carrot, raspberry, ham* and *juice*, the two example documents have zero overlap in the standard bag-of-words representation. However, at the higher semantic levels in the pyramid-of-words, the documents share the same synsets and thus have a distance of zero in that feature space.

Technical Details

The above example motivates the approach to use WordNet's hypernyms to construct a hierarchical pyramid-of-words. We now address the technical question of how the semantic levels of the pyramid can be constructed. WordNet's database is constructed in a *bottom-up* fashion, and to our knowledge all previous work on integrating WordNet into text feature sets has followed this bottom-up approach (Agirre and Rigau, 1996; Kehagias et al., 2001; Hotho et al., 2003b; Bloehdorn and Hotho, 2004; Hung and Wermter, 2004). I.e., entries from the bag-of-words are enhanced by features which are taken from the hypernym tree n-levels *above* the considered word. Our experiments strongly suggest that the bottom-up approach is not suited for a hierarchical representation as motivated above. Consider the table below:

word	hypernym levels					
jam	conserve	confiture	sweet	dainty	nutriment	**food**
sauce	condiment	flavorer	ingredients	foodstuff	**food**	
pepper	flavorer	ingredients	foodstuff	**food**		
dinner	meal	nutriment	**food**			
diet	fare	**food**				

Table 4.4: Hypernyms of some words occurring in the Reuters-21578 corpus.

The left column contains words which are present in the Reuters-21578 corpus. The right column shows their corresponding hypernym sequence. All words are related to the same semantic concept of *food*, but all words have a different depth d_{hyp} in WordNet's hypernym tree. E.g., $d_{\text{hyp}}(\text{jam}) = 7$, $d_{\text{hyp}}(\text{dinner}) = 4$. By constructing the pyramid-of-words bottom-up, e.g. by using the second hypernym entry for each word, we still would need five different dimensions to represent their meanings (*confiture, flavorer, ..., food*). By following a *top-down* approach we achieve a true semantic hierarchy where all words can trigger the concept *food* simultaneously. Consequently, we build the pyramid-of-words as follows:

$$\{p_i\}^l = \left\{\text{hyp}(w_i,\ d_{\text{hyp}}(w_i) - (\gamma(l-1) + \delta)\right\}, \tag{4.1}$$

where the p_i are the elements in pyramid level l, the w_i make up the standard bag-of-words model, $d_{\text{hyp}}(w_i)$ denotes the depth of w_i in WordNet's hypernym tree, hyp(w_i, x) denotes the xth hypernym of w_i and γ and δ are two additional parameters: γ controls the semantic granularity of the different levels in our pyramid-of-words. For $\gamma = 1$ we take each hypernym level of WordNet into account; δ controls the offset for our first semantic level: The root level of WordNet's hypernym tree for nouns contains just the one element *entity*. Since it would not be helpful to substitute each w_i with a constant, we introduce δ to set the first semantic level of the pyramid to a chosen semantic level in WordNet's hypernym tree.

Nouns vs. Verbs in WordNet

So far, we have only shown examples for nouns. In addition to nouns[1], WordNet also defines hypernyms for verbs[2]. Eq. 4.1 does not differentiate between the two, but as Table 4.5 and Fig. 4.15 show, WordNet treats nouns and verbs differently:

	$\max(d_{\text{hyp}})$	$\mu(d_{\text{hyp}})$	$\sigma(d_{\text{hyp}})$
nouns	18	8.98	1.97
verbs	13	3.42	1.58

Table 4.5: Statistics on WordNet hypernym depths of all nouns and verbs occurring in Reuters-21578.

In WordNet nouns are typically represented on a finer hypernym scale than verbs: On average, a noun has approximately nine hypernym levels, whereas verbs on average exhibit only 3.4 levels. As Fig. 4.15 indicates, the frequency curves of the hypernym depths could be modelled as Gaussians. Since the entries of the pyramid-of-words in each level should reflect the same semantic hierarchy independent of the word's morphology, we introduce an additional scaling factor for the hypernyms of verbs. So, Eq. 4.1 becomes

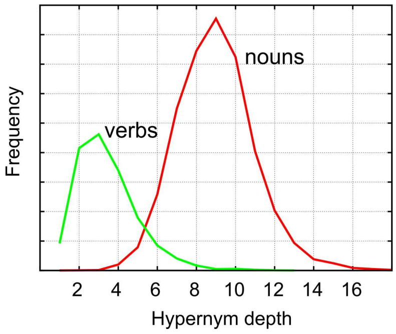


Figure 4.15: Frequencies of hypernym depths for nouns and verbs in WordNet.

$$\{p_i\}^l = \left\{\text{hyp}(w_i,\, d^m_{\text{hyp}}(w_i) - (\gamma(l-1)+\delta)\right\}, \quad (4.2)$$

with the scaled hypernym depth $d^m_{\text{hyp}}(w_i)$ depending on the morphology of word w_i:

$$d^m_{\text{hyp}}(w_i) = \begin{cases} d_{\text{hyp}}(w_i) & \text{if } w_i : \text{noun} \\ \beta\, d_{\text{hyp}}(w_i) + \alpha & \text{if } w_i : \text{verb}, \end{cases} \quad (4.3)$$

where α and β are chosen such that the resulting frequency distributions share the same mean and the same standard deviation.

Technical details about the implementation for the pyramid-of-words can be found in Chapter 5, where the overall technical system is described.

4.2.2 A Neural Model to perform Word Sense Disambiguation (WSD)

For reasons of simplification we have omitted an important detail so far. For the construction of the pyramid-of-words we have to tackle the problem of polysemy: A single word usually has more than one meaning, i.e., it is not a priori clear which of the multiple WordNet synsets and its hypernyms we have to incorporate for a given bag-of-words element. Consider the following example. The noun *store* has four different senses in WordNet:

1. (shop, **store**) → (mercantile establishment, retail store, sales outlet, outlet) → (establishment)

[1] thing *A* might be a kind of thing *B*

[2] doing *A* might be a way of doing *B*

2. (**store**, stock, fund) → (accumulation) → (net income, net, net profit, lucre, profit, profits, earnings)
3. (memory, computer memory, storage, computer storage, **store**, memory board) → (memory device, storage device) → (device)
4. (storehouse, depot, entrepot, storage, **store**) → (depository, deposit, depositary, repository) → (facility, installation)

When constructing the pyramid-of-words for a given document, we have to chose which of the possible senses to take into account. This task is known as *word sense disambiguation* (WSD). In context of WordNet, several approaches have been suggested, i.e. by Agirre and Rigau (1996); Hotho et al. (2003b). Generally, there exist two classes of approaches: The first just considers WordNet's statistics describing how often a certain sense occurs in average English language. The method then picks the most common sense. In case of the word *store*, always the first synset *(shop, store)* would be chosen. The second, more advanced class, uses the context a word appears in to disambiguate its sense. In all works known to author, the WSD algorithm then picks out the single most probable of all senses.

We here propose a simple neural model which for a given document computes the *probability* that a word within that document carries a certain sense, i.e. we do not pick out a single sense for each word, but assign a probability to each of the possible solutions. The model makes use of WordNet's *gloss* entries. These can be regarded as small documents describing each sense with a few words. For our "store" example the corresponding glosses for the senses 1 - 4 are as follows:

1. *A mercantile establishment for the retail sale of goods or services; "he bought it at a shop on Cape Cod"*
2. *A supply of something available for future use; "he brought back a large store of Cuban cigars"*
3. *An electronic memory device; "a memory and the CPU form the central part of a computer to which peripherals are attached"*
4. *A depository for goods; "storehouses were built close to the docks"*

For all words in the document we now attach to each possible sense a formal neuron describing the *activation potential* for the corresponding sense in that document. Fig 4.16 visualizes the concept. Formally, we can write the activation potential of sense number n for a word w_i in document d_j as

$$s_n(w_i \mid d_j) = \frac{\mathbf{f}_{\text{bow}}(d_j)\,\mathbf{f}_{\text{bow}}(g_n(w_i))}{\sum_k \mathbf{f}_{\text{bow}}(d_j)\,\mathbf{f}_{\text{bow}}(g_k(w_i))}, \tag{4.4}$$

where $\mathbf{f}_{\text{bow}}(x)$ denotes the feature vector given by the standard bag-of-words models for document x, and $g_n(w_i)$ corresponds to WordNet's *gloss* of sense s_n for word w_i.

Since there is no disambiguated Reuters-21578 corpus, a direct evaluation of our simple neural model would involve a manual disambiguation of words in a significant proportion of the corpus - a very time-consuming and laborious task. We therefore give an exemplary result on two documents containing our "store" example. The real benefits of the WSD-model are shown in Section 4.2.3, where we give the benchmark results on a H^{2}SOM trained with the pyramid-of-words - with and without our simple WSD-model.

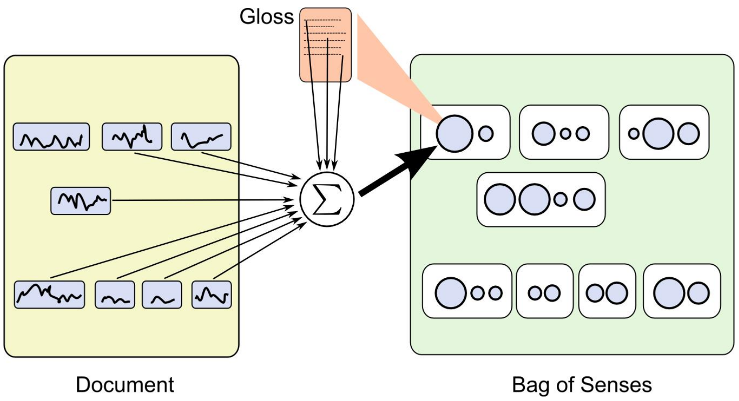


Figure 4.16: Neural model for word sense disambiguation. The left side shows a document with eight words. On the right hand side, for each word a set of senses is shown in the white boxes. To each sense a formal neuron is attached. Its activation resembles the probability that sense has for the given word in the given document.

WSD Example

The following example shows two documents containing several instances of the word "store":

(a) TECHNOLOGY/NEW ERA FOR INFORMATION HANDLING
Ground-breaking new systems for storing and retrieving information are ushering in a new era for computer companies and computer users. Within the past few weeks, International Business Machines Corp, Eastman Kodak Co and others have launched products that radically increase the amount of data that can be catalogued and shelved in computerized libraries. [...] Analysts say commercial versions of these chips are several years away, though some suspect that IBM may start volume production of its four-megabyte chip sometime this year. Such chips will enable computer makers to build computers with immense memory capacities. Reuter

(b) STOP AND SHOP'S BRADLEES FEBRUARY SALES UP
Stop and Shop Cos Inc said sales for the four weeks ended February 28 for its Bradlees Discount Department Stores Division were up six pct to 104 mln dlrs from 98 mln dlrs a year before, with same-store sales up three pct. The company said the modest comparable store sales increase was due to a combination of difficult weather conditions in the Northeast, a later Easter this year and a possible slowing in consumer demand. Reuter

Figure 4.17: Examples of two documents containing several instances of the word *store*. In document *(a)* the meaning of *store* is closely related to WordNet's third sense, in **(b)** it is more closely to WordNet's first sense.

Table 4.6 lists the activation potentials for the different meanings of *store* we obtain from our WSD-model with Eq. 4.4 for the contexts of documents *(a)* and *(b)*.

4.2.3 Results on Reuters-21578

For the results reported here we assemble all components of the previous section and work with a pyramid-of-words approach given by

$$\{p_i\}^l_{d_j} = \left\{s_n(w_i \mid d_j)\ \text{hyp}_n(w_i,\ d^m_{\text{hyp}}(w_i) - (\gamma(l-1)+\delta)\right\}, \tag{4.5}$$

where the p_i are the elements for document d_j in pyramid level l, $s_n(w_i|d_j)$ and $d^m_{\text{hyp}}(w_i)$ are given by Eq. 4.4 and Eq. 4.3, respectively, and $\text{hyp}_n(w_i, x)$ denotes the xth hypernym of the nth sense of word w_i. The construction might seem bloated, but exhibits only the two

	WordNet synset	*(a)*	*(b)*
nouns	*(shop, store)*	0.072	1.000
	(store, stock, fund)	0.052	0.000
	(computer memory, storage)	0.753	0.000
	(storehouse, depot, entrepot)	0.082	0.000
verbs	*(store, give away, put in)*	0.000	0.000
	(store)	0.041	0.000

Table 4.6: The sense activation potentials from our WSD-model for the *store*-example.

free parameters γ and δ. If not stated otherwise, we always use $\delta = 3$ and $\gamma = 2$, resulting in a representation as shown by the blue bars in Fig. 4.14.

For the preprocessing, we use WordNet's *morphy* engine (for details see Chapter 5, Section 5.1.2). The resulting dimensionalities for the standard bag-of-words and the hierarchical pyramid-of-words are given in Table 4.7. The table shows, that the first levels of the pyramidal representation achieves a significant dimensionality reduction. The larger number of entries in the last level - as compared to the standard bag-of-words - is explained by the polysemy of words, i.e. many words in the bag-of-words have more than one possible meaning.

bow	pow level 1	pow level 2	pow level 3
17480	495	6574	19989

Table 4.7: Dimensionality for different representations of Reuters-21578: for the standard bag-of-words, and for three levels in the pyramid-of-words.

Additionally all elements of the pyramid-of-words are weighted in standard fashion with a *term-frequency* × *inverse document frequency* approach as described in Section 2.2.1. The training of the H^{2}SOM with the pyramid-of-words (pH^{2}SOM) is carried out in standard fashion as described in Section 4.1.2. Just one detail is changed: The innermost ring is trained with feature vectors corresponding to the first level of the pyramid-of-words, the next ring with the second pyramidal level, and so on. Usually, the H^{2}SOM is comprised of more rings than the pyramid-of-words has levels. We therefore fall back to the standard bag-of-words for the training of the outermost rings in the H^{2}SOM. In analogy to image processing this corresponds to the pixel-level.

Fig. 4.18 illustrates the performance of the pH^{2}SOM. The microaveraged results for the twenty most frequent Reuters categories do not show a dramatic improvement over the H^{2}SOM trained with the standard bag-of-words model, but nevertheless for any given recall the precision of the pH^{2}SOM generally outperforms that of the plain H^{2}SOM. Note, that all four SOM-types show a similar shaped curve: The precision starts at very high values for low recalls and then gently declines with increasing recall levels. Interestingly, for all SOMs the microaveraged curve shows a sharp cave-in at a precision level of approximately 80%. The main difference between the different SOM types is "how far they have come" in terms of recall at that point.

The data in Table 4.8 shows a small decrease in precision for HSOM, H^{2}SOM and pH^{2}SOM with respect to the SOM baseline, but a strong increase in recall. At its cut-off point the pH^{2}SOM is able to retrieve approximately 39% more documents than the plain SOM - at the cost of 2.8% less precision.

Though the microaveraged results for the pH^{2}SOM and the H^{2}SOM show similar

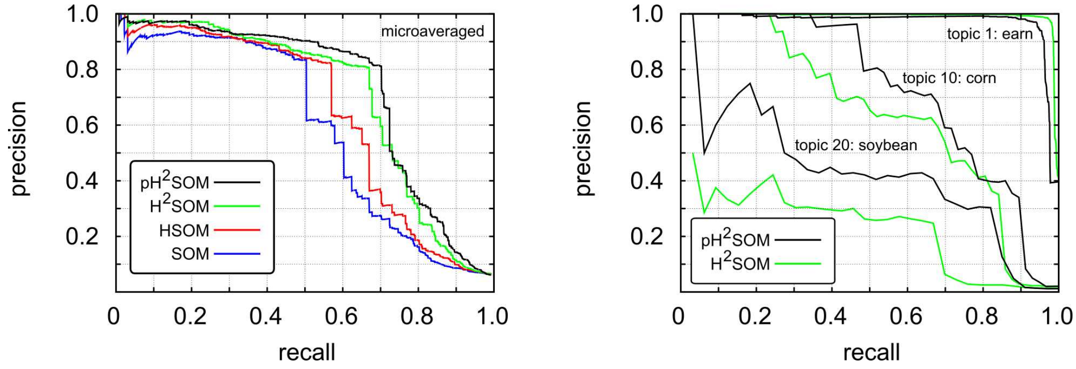


Figure 4.18: Performance in terms of precision/recall for the Reuters-21578 data and different SOM types. The left figure shows the micro-averaged results over the twenty most frequent categories, the right shows results on the three topics *earn*, *soybean* and *corn*, which are the most, the least and median frequent topics, respectively.

	recall / precision at cut-off point	relative gain
SOM	0.504 / 0.835	
HSOM	0.570 / 0.815	+13.1% / -2.4%
H^{2}SOM	0.670 / 0.806	+32.9% / -3.5%
pH^{2}SOM	0.702 / 0.812	+39.3% / -2.8%

Table 4.8: Recall / precision values for the different SOM types at their microaveraged cut-off points.

characteristics, the performance on single topics expose a more significant difference: At recall levels above 0.90, Fig. 4.18 shows that the precision of the pH^{2}SOM for the most frequent topic *earn* does not achieve equally high values as the H^{2}SOM. However, with respect to information retrieval tasks, this is not a significant drawback. At a high recall level of 90% the pH^{2}SOM still achieves a precision of over 98%, well above any critical value for practical information retrieval problems. The pyramidal approach shows it strength for the less frequent topics in the Reuters corpus: The *soybean* topic occurs in only 1% of documents in the training set. For this comparatively hard task, the precision achieved by the pyramid-of-words is generally between 0.1 and 0.4 above the precision achieved with the standard bag-of-words. In terms of relative performance gain this is a strong increase between 30% and 130%.

By taking these results on the Reuters-21578 corpus into account, we conclude that the H^{2}SOM strongly benefits from the pyramid-of-words approach. Generally, the observed precision for any given recall has been the highest of all SOM-models so far, and specifically for less frequent topics, the pyramid-of-words is able to push the classification results by large margins.

Influence of the WSD model

In order to evaluate the contribution of the WSD model, we have constructed a pyramid-of-words without any word sense disambiguation: For words with more than one sense we have set a constant weighting factor of one, i.e. $s_n(w_i \mid d_j) = 1$ in Eq. 4.4. Fig. 4.19 shows the results: Without WSD the general performance is significantly reduced. The strong decrease in precision is caused by the introduction of a large amount of noise in the data: By including the hypernym tree for all possible senses, the chances are increased that two items share the same entries. The result is a "smeared" dataset which seems to make it harder for the SOMs

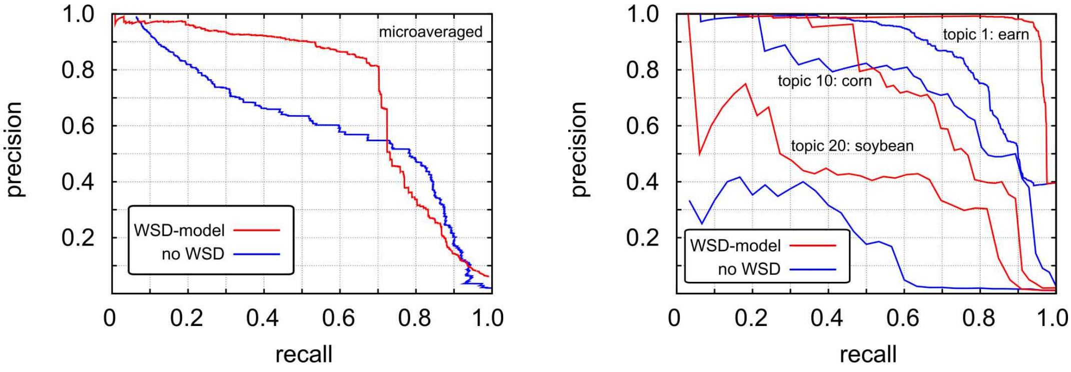


Figure 4.19: Performances for the pH^{2}SOM - with and without the inclusion of the WSD model for the construction of the pyramid-of-words.

to separate between different topics. We therefore believe that word sense disambiguation is an important requirement for the construction of text feature vectors with the use of WordNet.

Comparison to other Classifiers

Joachims (1998) has benchmarked several machine learning approaches with the Reuters-21578 dataset. This gives us the opportunity to relate the pH^{2}SOM to other methods. Since Joachims (1998) has used to ten most frequent categories in his experiments, we show in Fig. 4.20 the corresponding microaveraged result for the pH^{2}SOM. Additionally plotted are the precision-recall breakeven points (where precision equals recall) Joachims (1998) has reported. The pH^{2}SOM performs significantly better than a *naive Bayes* classifier and its breakeven point is directly between that of the *C4.5* decision tree and the *k-nearest neighbor* approach. The *support vector machine* using a *RBF-kernel* clearly outperforms all other approaches with respect to classification accuracy.

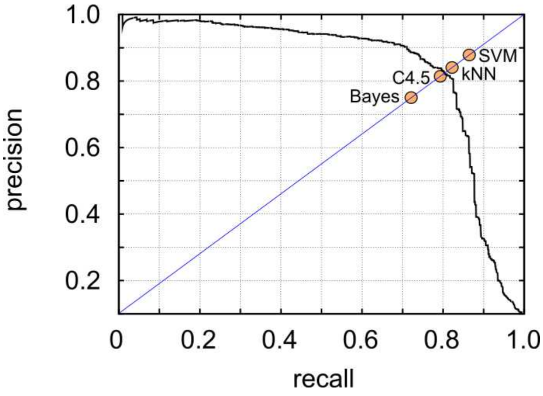


Figure 4.20: Microaveraged precision-recall curve for the pH^{2}SOM over the ten most frequent categories in Reuters-21578. Also plotted are the precision-recall breakeven points reported by Joachims (1998) for a *Naives Bayes* classifier, a *C4.5* decision tree, a *kNN*-approach and a *SVM*-classifier with *RBF*-kernel.

The above comparison shows that the pH^{2}SOM does not offer the very high classification performance of the support vector machine. However, since the pH^{2}SOM does not act like a black-box classification system as the SVM, but additionally provides insightful browsable maps supporting the understanding of large bodies of document data, we believe that the observed categorization performance compares sufficiently well with the more specialized (non-visualization) techniques.

4.3 Dealing with Time in Document Streams

In many cases document collections contain time-stamps. E-mails for example, are associated with the time they were sent, news items are published at a certain time, or a message in an Internet forum is posted at a specific moment. In addition to the semantic structuring of the document collection, the user might therefore be interested in how the different topics evolve over time. In order to deal with time components in text databases we propose an extension to the H^{2}SOM which we describe at this stage, since the formal neurons of the network are affected. An application of the extension is given in the following chapter.

4.3.1 Extending the H^{2}SOM by Leaky Integrators

To each neuron of the H^{2}SOM we attach a time dependent activation potential enabling the nodes of the network to act like *leaky integrators*:

$$\mathcal{A}_i(t) = \beta\, \mathcal{A}_i(t-1) + \mathcal{S}_i(t) \quad \text{with} \quad \mathcal{S}_i(t) = \begin{cases} \mathcal{I}_l & \text{if node } i \text{ is best-match node} \\ & \text{during time interval } t \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where β is a decay factor controlling the amount of leakage, $\mathcal{I}_l$ the amount by which the activation is increased in H^{2}SOM level l and t a time interval. As news or other text items “flow” in, the neuron activities of the corresponding best match nodes increase. At times with no news coverage, node activations decrease again. Note, that the H^{2}SOM has not one, but several best match nodes for each data item - one in each hierarchical level. Since the upper hierarchical levels contain much fewer nodes, the probability for nodes in the inner rings to be a best match node is much higher than in the perimeter of the H^{2}SOM. We therefore scale the node inputs $\mathcal{I}_l$ depending on their hierarchical level l, such that activations in the periphery are not dominated by the inner nodes.

From a practical point of view it has turned out that a discretization of the time span defined by the oldest t_{min} and newest t_{max} item in the database is favorable. By dividing the time span $[t_{\text{min}}, t_{\text{max}}]$ into N equitemporal intervals, t in Eq. 4.6 runs between 1 and N and denotes the following interval:

$$\left[t_{\text{min}} + (t-1)\,\frac{t_{\text{max}} - t_{\text{min}}}{N},\ t_{\text{min}} + t\,\frac{t_{\text{max}} - t_{\text{min}}}{N} \right] \tag{4.7}$$

Results on real world data sets as reported in Chapter 5 show that the sequence of the $\mathcal{A}_i$ can be quite jagged. This makes it hard to detect coarser trends within a visualization. For visualization issues we therefore compute a moving average defined as

$$\mathcal{A}_i^{\text{MA}}(t) = \frac{1}{M} \sum_{j=0}^{M-1} \mathcal{A}_i(t-j) \tag{4.8}$$

The width M of the smoothing window controls the length of the time interval over which the node activations are averaged.

4.4 Summary

In this chapter we have introduced the hierarchically growing hyperbolic self-organizing-map (H^{2}SOM). The H^{2}SOM generally utilizes the same sort of hyperbolic lattice structure already employed for the HSOM. While the HSOM treats all nodes of the network equally, the H^{2}SOM opens up a new direction to exploit the exponentially growing hyperbolic space: The growth law in $I\!H^2$ allows for a natural embedding of arbitrary large hierarchical data structures. We can therefore train the neurons of the H^{2}SOM in a hierarchical manner: The initial top-level hierarchy is trained in standard fashion and nodes exceeding a predetermined quantization error are expanded. The resulting sub-levels are then trained with those data items which get mapped to their parental nodes. This scheme allows for a drastically reduction of the most time consuming step of the self-organizing approach. By approximating the global search for the winner neuron by a *fast tree search*, taking as the search root the initial center node of the growth process and then following the natural hierarchical structure in the hyperbolic grid, we achieve a search path with a complexity of $\mathcal{O}(logN)$ instead of $\mathcal{O}(N)$ for a global search.

In the context of text categorization tasks our experiments reveal that for maps consisting of the same number of neurons the H^{2}SOM generally outperforms the HSOM in terms of neighborhood preservation and classification accuracy. This is achieved together with a reduction in training time from several hours to several minutes. Only in terms of quantization error the H^{2}SOM performs worse. This is explained by the different utilization of the lattice structure: While the HSOM can use all nodes to approximate the dataset, the H^{2}SOM uses the first “inner” levels of the grid for the hierarchical structuring of the data, such that less nodes are available to represent the data on the same level as the single nodes of the HSOM can do.

In addition to textual data we have given an example of how the H^{2}SOM performs on the MNIST dataset of handwritten digits. Also for this data, the H^{2}SOM easily outperforms the SOM in terms of training time and classification accuracy.

Motivated by approaches from computer vision to represent image data in a hierarchical manner, we have introduced the notion of the *pyramid-of-words*. By the integration of the WordNet lexical database, we achieve the benefits of a semantically representation of documents: On a coarse semantic level documents can be represented with fewer dimensions. This allows for a matching of documents which do not share the same words but which talk about the same cognitive concepts. In order to deal with linguistic polysemy we have introduced a simple neural model which performs a word sense disambiguation using the context of the documents in which the to be disambiguated words appear. The results on the Reuters-21578 corpus indicate that overall classification performance is slightly increased by the pyramid-of-words. For low frequently appearing topics the precision is significantly increased between 30% and 130% as compared to the standard bag-of-words representation.

The comparison to other classifiers shows that the H^{2}SOM trained with the pyramid-of-words achieves very competitive results. In terms of overall precision-recall accuracy, the H^{2}SOM is significantly better than a *naives Bayes* classifier and almost on par with a *k-nearest neighbor* approach. A *support vector machine* (SVM) performs even better, however at the cost of larger computational complexity and not offering a visualization framework as the H^{2}SOM does.

Since many document collections contain time-stamped messages, we have presented an extension to the H^{2}SOM by adding leaky integrators to each neuron. As we show in the next chapter this allows for a visualization of topic developments in time.

Chapter 5

Design for an Interactive Text Visualization System

In the previous Chapters 3 and 4 we have described the theoretical foundations for the hyperbolic SOM (HSOM) and the hierarchically growing hyperbolic SOM (H^2SOM). Based on standard benchmark datasets like the Reuters-21578 corpus we have demonstrated the advantages of the H^2SOM with respect to text categorization and information retrieval tasks. This chapter introduces the technical components which are used for the visualization framework of the H^2SOM. We give examples on the operation of the overall visualization system for three real world case studies.

5.1 Components

The overall system architecture is comprised of four building blocks which are schematically shown in Fig. 5.1. The following sections describe these components in more detail.

5.1.1 Database

For large scale datamining we need a scalable data storage system which provides a standardized way to store and retrieve large amounts of data. A SQL database engine naturally meets these demands. For our system we have chosen MySQL[1], an open source solution with a large installation base. All components of the system exchange data via standardized SQL calls. On the implementation level these are realized by MySQL's native C-API, but could be easily exchanged by any other database vendor.

In the following we describe the four central data modules of the system, represented by the four green cylinders I - IV in Fig. 5.1.

Data Module I: Raw Data

In our system each document is represented by at least five fields which are stored in the raw text table of the database:

1. a unique identifier assigned during import
2. a date specifying when the document was published

[1] http://www.mysql.com

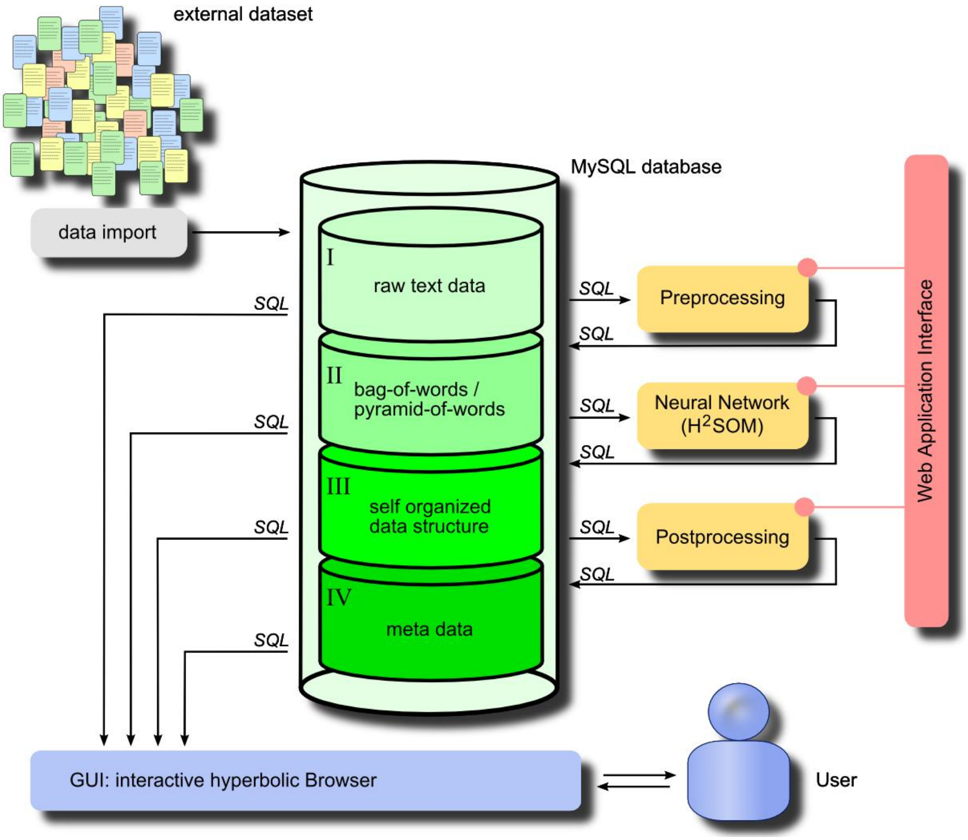


Figure 5.1: The overall system architecture consists of four building blocks: *(i)* the database system (green), *(ii)* the textmining engine (orange) with the H^{2}SOM as its central element, *(iii)* an interactive visualization framework (blue) and *(iv)* a web application interface (red).

3. a title
4. the full text of the document
5. a list of topics which might be assigned to the document

The framework also allows to store additional customized fields for each document collection, but this additional information is not processed by the text mining engine. Different document corpora are generally delivered with different formats. Therefore, the data import usually requires some sort of customized preprocessing or parsing. In all reported cases below we have applied Perl[1] scripts to handle the data import and to achieve a standardized document representation within our framework.

Data Module II: Feature Data

The second data module stores the results of the preprocessing engine (which is described in more detail in Section 5.1.2 below). Within the context of text mining these numerical feature vectors result from the bag-of-words and the pyramid-of-words, respectively.

Data Module III: SOM Data

The third data module reflects the internal states of the system's Neural Network. The corresponding SQL tables represent the hierarchical data structure that was learned by the H^{2}SOM

[1] http://www.perl.org

from the feature data of module II. Technically, this corresponds to the node tree of the H^{2}SOM's neurons in hyperbolic space plus their associated prototype vectors.

Data Module IV: Meta Data

The fourth data module stores results obtained from the postprocessing engine (described in more detail in the next section below). It contains statistical data such as average prototype vector distances, node labels or node activation potentials (cf. 4.3) for display by the interactive hyperbolic browser.

5.1.2 Text Mining Engine

The text mining engine - shown as the three orange boxes in Fig. 5.1 - represents the heart, or better said the brain of our system. It consists of the three parts described in the following.

Preprocessing

The preprocessing engine is responsible for extracting numerical feature vectors from the unstructured raw text data. The bag-of-words and the pyramid-of-words features are generated in six steps as follows:

1. **Tokenization:** A simple parser breaks down each document into a set of single words. Separators are white spaces and punctuation marks. Numbers are treated as symbols and are divided into the three groups *small number*, *large number* and *year number* by a simple rule set[1]. For each token its frequency within the document is counted. After this step, e.g., document (b) in Fig. 4.17 on page 69 becomes: *stop:3 and:4 [...] said:2 shops:2 shop:1 reuter:1*. Note, that this representation could already be used as a very simple bag-of-words.

2. **Advanced Stemming:** In a second step, we utilize WordNet's *morphy* engine to reduce the dimensionality of the simple bag-of-words obtained in the previous step. WordNet (Fellbaum, 2001) offers a morphological parser which is able to handle a range of morphological transformations such as plurals and different tenses. By looking up each token from step 1 by *morphy* we obtain a much smaller list of words. In case of Reuters-21578 the dimensionality reduces from 40831 unique words to 17480 base forms.

3. **Stop Word Elimination:** Some words to not carry valuable semantic information. In information retrieval they are called *stop words* (Baeza-Yates and Ribeiro-Neto, 1999). Common examples for stopwords are *the*, *to*, *of*, and *and*. We apply a stop word list taken from Salton (1991) to delete these words from the bags.

4. **Hypernyms and Dictionary:** This is the first step to create the pyramid-of-words described in Section 4.2.1. For each base form obtained from step 3, we utilize WordNet to look up the hypernym tree for each possible sense of that base form. The results are written to data module II in order to accumulate a hypernym dictionary for the analyzed document corpus. Table 5.1 shows two exemplary entries from the resulting hypernym dictionary.

   From the data in the hypernym dictionary we are able to compute a global document statistic for the occurring word base forms. E.g. from Table 5.1, we see that in 30934

[1] Numbers below and above 100 are *small* and *large*, and numbers between 1910 and 2010 are treated as years

word	count	number of documents	depth
food#1_03_00020429	30934	8833	3.08
city#1_15_08406385	6981	3222	6.92

Table 5.1: Examples for entries in the hypernym dictionary. The *word* corresponds to WordNet's unique identifier for the corresponding word/sense combination, *count* is the number of occurrences within the whole database whereas *number of documents* denotes the number of distinct documents the term occurs in. The *depth* value corresponds to $d^m_{\text{hyp}}(w_i)$ in Eq. 4.3.

cases a word triggers the concept "food" within 8833 distinct documents. Inspired by the *term frequency* × *inverse document frequency* weighting in information retrieval (Baeza-Yates and Ribeiro-Neto, 1999), we compute a *tfidf-ranking* value for each word defined as:

$$\text{tfidf}_D(w) = \log\left(N^D_w\right)\log\left(\frac{N_D}{N^w_D}\right), \tag{5.1}$$

where N^D_w is the number of occurrences of word w in the whole document collection D, N^w_D is the number of documents word w occurs in, and N_D is the total number of documents in the database. Therefore, $\text{tfidf}_D(w)$ can be interpreted as a measure of importance of word w within document collection D. We use this measure to limit the dimensions of the feature vectors created in step 6 below.

As a side note, the combination of $\text{tfidf}_D(w)$ with the hypernym dictionary opens a simple but powerful statistical instrument: We can easily build categorial summaries, i.e. how many different types of food are mentioned in a document collection, and how "important" they are with respect to the corpus. As an example, the three most important German cities in the Reuters collection are "Bonn" [1], "Hanover", and "Frankfurt".

5. **Word sense disambiguation:** In a next step, for each document we compute the word sense probability each possible sense of a base form has within the context of that document. Details of the WSD-model are described in Section 4.2.2 above.

6. **Vector creation:** As a final step, we create the numerical feature vectors for the bag- and the pyramid-of-words. To this end, we sort the dictionary obtained in the previous step with respect to $\text{tfidf}_D(w)$. We do not take the full bag-of-words with all 17480 base forms, but just the first 8000 with respect to $\text{tfidf}_D(w)$ into account. The reason for the limitation is from a computational view only. We did not observe a significant decrease in classification accuracy if we drop the less important terms (with respect to $\text{tfidf}_D(w)$) from the dictionary. However, numerical computations with fewer dimensions are much faster.

   After determination of the bag's components according to the dictionary statistics, we are able to construct the feature vectors for each document by simply counting the relevant words occurrences in that document. We also follow standard practice (Baeza-Yates and Ribeiro-Neto, 1999) and weight each component with a *term frequency* × *inverse document frequency* approach:

$$\text{tfidf}_d(w) = N^d_w \log\left(\frac{N_D}{N^w_D}\right), \tag{5.2}$$

[1] In the time of Reuters-21578, Bonn was still the capital of Germany.

where N_w^d is the number of times word w occurs in document d.

Neural Network - H^{2}SOM

The neural networks engine reads the numerical representations for the bag- and the pyramid-of-words created by the preprocessing steps explained above. The H^{2}SOM is constructed and trained as laid out in Chapter 4. The training results, i.e. the resulting prototype vectors are stored within data module III.

Postprocessing

The postprocessing engine computes additional statistical data from the self-organized data structure obtained by the H^{2}SOM. This meta data is intended to be visualized by the interactive hyperbolic browser:

1. **Number of hits:** For each node we compute how often it was the best match node for a data item. Since the smaller number of nodes in the upper levels of the H^{2}SOM hierarchy results in larger hit numbers, we normalize the values according to

$$h(a) = \frac{H(a)}{\max_{i \in L(a)}(H(a_i))}, \tag{5.3}$$

   where $h(a)$ is the number of normalized hits for node a, $H(a)$ is the number of absolute hits for node a, and $L(a)$ is the set of nodes which constitute the hierarchical level node a is located in. The normalization results in values between zero and one for the number of hits in each hierarchical level of the H^{2}SOM. Zero, if the node was never a best match node, one, if the node was hit with the largest frequency in its hierarchy level.

2. **Assigned topic:** If the documents in the collection are labeled with topic or class information, each node carries the topic distribution of those documents it was a best match node for - for details refer to Section 3.5.1.

3. **Purity:** The purity of a node is defined as

$$\text{pur}(a) = \frac{1}{N_{t_a^k > 0}}, \tag{5.4}$$

   where $N_{t_a^k > 0}$ is the number of non-zero entries in the topic-vector for node a.

4. **Average Distance:** For each node, we compute its average distance in the feature vector space to its two sibling nodes in the hierarchy. This measure is usefull to display a U-matrix variant in the hierarchy of the nodes.

5. **Keywords:** In order to provide the user with hints about the contents of the documents mapped to each node of the H^{2}SOM, we label all nodes with keywords. To this end we sort the node's prototype vectors by their component values such that the component with the highest entry comes first. We then use the corresponding dictionary entries from data module II to assign the first ten entries from the sorted values as keywords to the corresponding node. This is also schematically shown in Fig. 5.2.

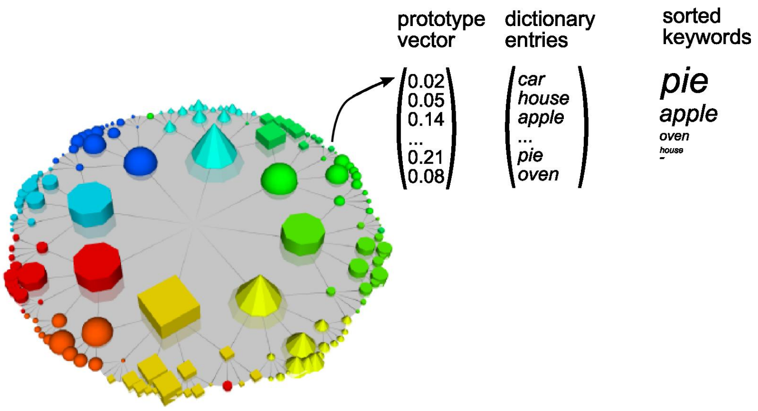


Figure 5.2: Automatic keyword generation. The node's prototype vectors are sorted with respect to their component values and the top corresponding dictionary entries are assigned as keywords. The indicated font scaling is executed by the visualization module.

6. **Time dependent node activations:** If the document collection contains time stamps, i.e. an information when a certain document was published, the system computes a time dependent activation potential for each node. For details refer to Section 4.3.

5.1.3 Visualization Engine

The visualization engine is the key component of the interactive browser which enables the user to browse through the hyperbolic hierarchy learned by the H^{2}SOM. The GUI is based upon the open source framework "The Visualization Toolkit" (VTK)[1] by Schroeder et al. (1997). The toolkit consists of a C++ class library providing a wide variety of objects and methods for data visualization. In context of the hyperbolic browser our visualization methodologies are introduced in Section 5.2 below.

5.1.4 Web Application Interface

The web application interface provides methods which can be accessed by external clients. The services can be executed by means of XML-RPC calls. Therefore, providing a platform independent way to integrate the results of our text mining system into third party web applications. To give a brief overview on the provided methods, we enlist them here:

- *makeDictionary*: Compute the complete dictionary statistics and generate the bag- and pyramid-of-words representation for a given document collection.
- *makeMap*: Generate a hierarchical hyperbolic self-organizing map.
- *mapDocument*: Map an unknown document to an existing H^{2}SOM map.
- *getCategories*: Obtain the topic vectors for a given map region.
- *getSimilar:* Map an unknown document to an existing map and retrieve similar documents.
- *getFeedback:* Obtain user feedback about the quality of suggested similar documents.

[1] http://www.vtk.org

- *searchInDocs:* Perform a standard indexed search within the document collection and return search results as map positions. This method is particularly useful to perform a *Semantic Search* which is detailed below in Section 5.2.5.

The web application interface was deployed in the *aid* project described in Section 5.3.3.

5.2 Visualization Methodologies

As described in Section 3.3.1 the nodes of the HSOM and the H^{2}SOM are placed on the vertices of a regular tessellation of the two-dimensional hyperbolic plane $\mathbb{H}^2$. In the following we describe how the resulting maps are visualized in Euclidean 3D space. The graphical components are introduced and demonstrated by examples. The overall application of the complete framework to real world data sets is given in Section 5.3.

5.2.1 The H^{2}SOM in 3D Euclidean Space

The Poincaré disk itself is modelled as a VTK-surface in 3D Euclidean space. It is shown as a light blue disk in Fig. 5.3 where VTK's 3D coordinate system is illustrated as a bounding box around the Poincaré disk. The user interface distinguishes between two navigation modes: *(i)* a navigation within the whole 3D scene, allowing for arbitrary translation and rotation of the 3D viewpoint as shown in Fig. 5.3, and *(ii)* a navigation within the Poincaré disk allowing to adjust the focus within the hyperbolic space as discussed in Section 5.2.2 below.

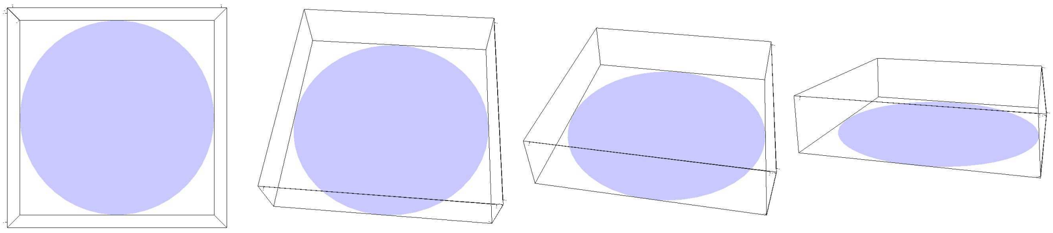


Figure 5.3: The circular Poincaré disk shown in 3D: The boxes indicate the 3D coordinate system used by the VTK visualization engine. The user interface permits to "grab" any point in the 3D scene and drag it to a new location, allowing for an arbitrary rotation of the Poincaré disk in 3-space.

The H^{2}SOM is graphically represented by three VTK objects:

1. A set of 3D glyphs illustrates the nodes of the network. The framework allows to modify shape, color, texture and size of the objects. Node properties can either be fixed, or their values might be driven by data obtained from the postprocessing module as described on page 79. In Fig. 5.4 the nodes are represented by eleven different glyph types, corresponding to the eleven clusters of the 10-dimensional tetraeder dataset (refer to Section 3.4.3 on page 38).

2. Directional links between the nodes visually encode the H^{2}SOM's hierarchical structure. Child nodes are connected by fading lines to their parents. This is illustrated in Fig. 5.4 where the "up-direction" within the hierarchy is indicated by bright ends of the connecting lines. Sibling nodes within the same hierarchy level are not connected to each other.

3. The nodes constitute the mesh points of a 2D surface defining the H^{2}SOM's map. Scalars might be mapped through an arbitrary color scale to define its appearance. In

Fig. 5.4, the average node distances in their feature space is plotted, i.e. neighboring nodes sharing similar prototypes are rendered on a red background opposed to nodes on a blue background whose prototype vectors are far apart.

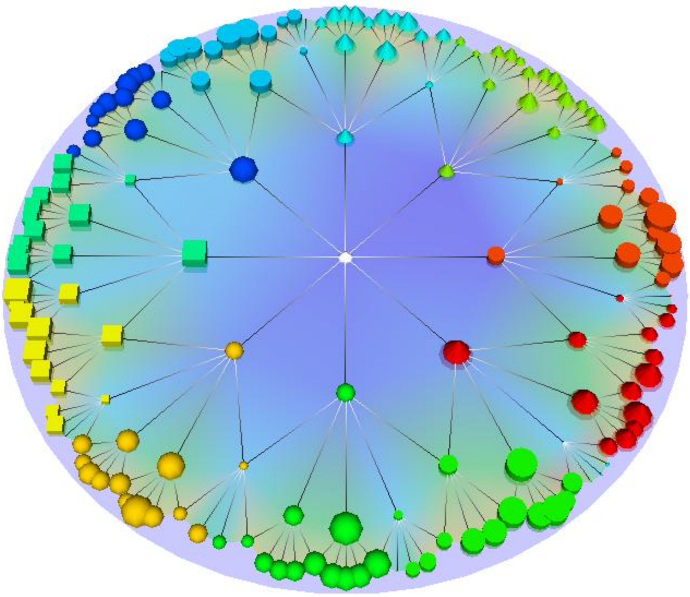


Figure 5.4: A visualization of a 8^5 H^{2}SOM with three hierarchical levels shown. Nodes are represented by 3D glyphs and the hierarchical tree by directional links between them. The surface spawned by the nodes is rendered with a color scale to represent average node distances in the feature space.

5.2.2 Navigating in Hyperbolic Space

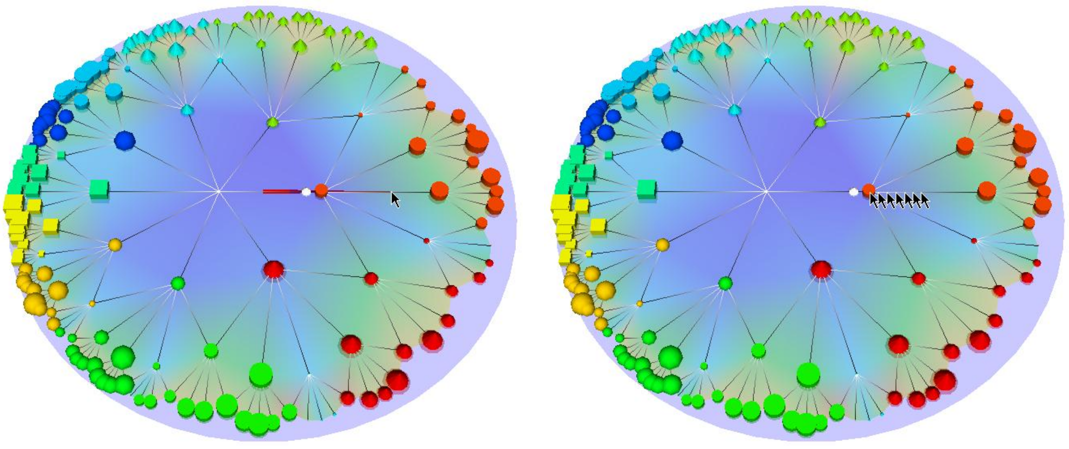
As described in Section 3.2.3 on page 28 the group of *Möbius transformations* $\text{Möb}(\mathbb{D})$ allows for a continuous translation of the focus to any point of the infinitive hyperbolic plane. In Fig. 5.4 above the focus is centered at the origin of $I\!H^2$ and depicted by a bright white cone. For an interactive exploration of the hyperbolic self-organizing H^{2}SOM tree our framework offers two different mouse interaction schemes.

Driving the Focus

For the first navigation variant, we use VTK's 3D engine to obtain the complex coordinate $z_x \in \mathbb{D}$ where the mouse pointer intersects with the surface of the unit disk $\mathbb{D}$ (representing the Poincaré disk). While the mouse button is pressed we compute

$$c_{t+1} = c_t + v \, \frac{z_x}{\text{arctanh}(c_t)}, \tag{5.5}$$

where c_t is the position of the focus point within the hyperbolic map at time step t and v a scalar controlling the velocity of the movement. The scaling factor $\text{arctanh}(c_t)$ compensates for the squeezing effect of the Poincaré projection. It ensures that the perceived changes of node positions appear with constant speed regardless of the focus position in hyperbolic space. All node positions then undergo a Möbius transform $M_{c_{t+1},0}$ as defined by Eq. 3.21. Fig. 5.5 (a) shows the resulting situation after the user has pressed the mouse at the indicated position for a few time steps. Depending on the chosen v, Fig. 5.5 (a) is reached at approximately one second of movement - if started from a centered map.



(a) (b)

Figure 5.5: Illustration of the two hyperbolic navigation schemes: In the "driving" variant **(a)**, mouse position and origin of the Poincaré disk define a motion vector by which the focus point is translated while the mouse button is pressed down (here the focus point is moved towards the right of the map). In the "dragging" variant **(b)** any point in the Poincaré disk can be grabbed and moved to a new location (here the red cylinder is dragged to the left towards the center of the map).

Dragging the Focus

For the second navigation variant, we evaluate two coordinates: the first corresponds to the point $z_1 \in \mathbb{D}$ where the user presses and holds down the mouse button, the second to the point z_2 where he drags the mouse to while holding down the button. These two coordinates define a unique Möbius transformation $M_{d,0}$ such that

$$M_{d,0}(z_1) = \frac{z_1 - d}{1 - \bar{d} z_1} = z_2 \tag{5.6}$$

maps point z_1 to z_2. All node positions then undergo that Möbius transformation $M_{d,0}$. Fig. 5.5 (b) shows the resulting situation after the drag operation. The seven mouse bitmaps indicate the mouse movement from the right to the left, i.e. the user has grabbed the innermost red cylinder sphere in Fig. 5.4 and moved it to the new position as indicated in Fig. 5.5 (b).

Both hyperbolic navigation variants achieve a continuous translation of the focal position and allow for a smooth focus & context navigation framework. In Section 6.2.2 both variants are compared to each other in one of our user studies.

Dynamic Node Rendering

In Figures 5.4 and 5.5 three of the five hierarchical levels of the H^{2}SOM are shown. From a visualization viewpoint it is not desireable to show more than three or four hierarchical levels at once for two reasons:

1. Due to the strong fish-eye effect resulting from the Poincaré projection, objects at the perimeter do not have sufficient rendering space to offer a meaningful visualization.

2. The number of nodes grows asymptotically exponentially with increasing hierarchy level. The visualization of several thousands of 3D objects representing the nodes in the deeper hierarchies would computationally not be feasible.

In order to allow for a visualization of very large H^{2}SOMs we therefore propose the following solution: The VTK framework uses a fixed number of node objects corresponding to the first inner three or four node rings. Starting at the node displayed at the center of the Poincaré disk, the node tree is expanded until all to be shown levels have been visited. The resulting effect is shown in Fig. 5.6: Both screenshots contain the same number of 3D objects representing the network's nodes. In (a) it can be seen that the nodes on the left become increasingly squeezed as the focus is moved towards the right portion of the map. In (b) the focus - again shown as a bright white cone - has crossed the first hierarchy level and a number of 3D objects is "copied" from the left to the right - providing the resources to visualize more details of the deeper hierarchies in the right part of the map. In the implementation this is efficiently realized using a hashed data structure which copies the corresponding node properties from the internal node representation to the VTK object properties.

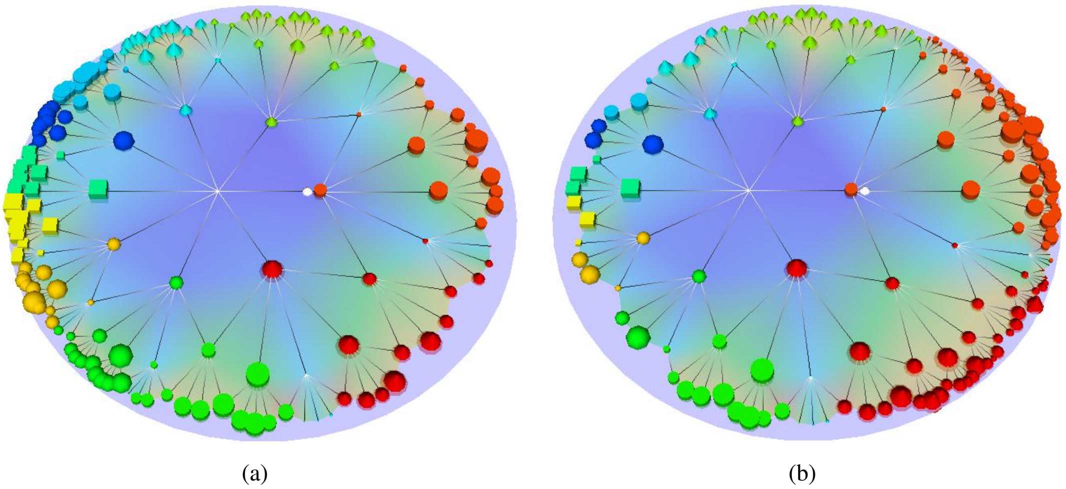


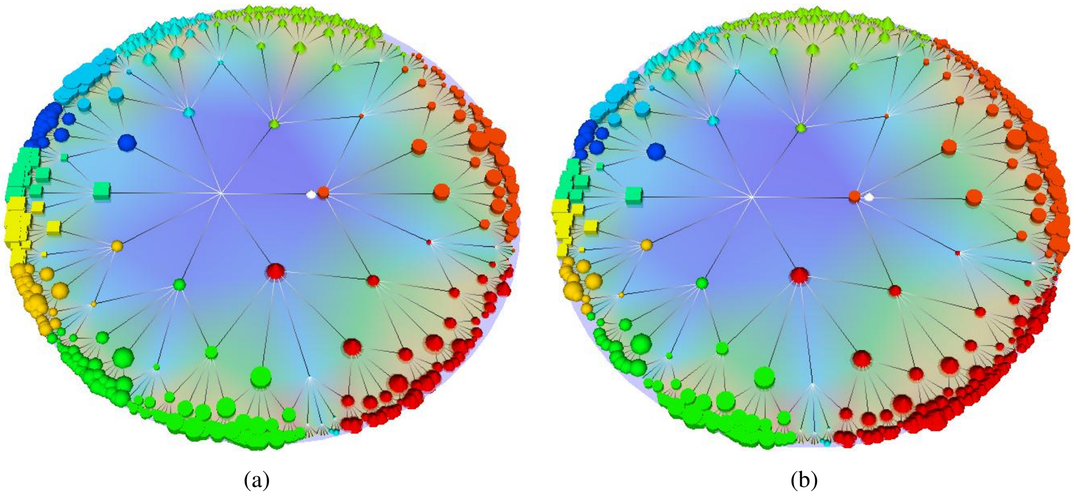
Figure 5.6: The dynamic node visualization renders the nodes on demand. Depending on the focus position only the first n rings with respect to the center node are shown although the map might have many more hierarchy levels (here $n = 3$). From (a) to (b) the focus has crossed the first hierarchy level such that (b) shows more details from the right part of the map than (a) does.

In Fig. 5.6 only three hierarchical levels are displayed to demonstrate the dynamic node rendering more clearly. If we increase the number of shown levels to $n = 4$, the dynamic node effect becomes more subtle as shown in Fig. 5.7.

The dynamic allocation of graphical resources depending on the focal position allows for a very realistic navigation in the infinite hyperbolic space: First, the user always sees the full surrounding context as far as it is sensible with respect to the available rendering space. And second, the user observes no limitation of "where he can go": the framework dynamically loads and shows the requested node data from the corresponding hyperbolic area as soon as the user enters it.

5.2.3 Self-Organizing Tag Clouds

As outlined in the introducing Chapter 2 so called *tag clouds* have become popular in Web 2.0 applications. They are used as a visual guideline to give hints about the semantic content of e.g., web pages, articles or products (Hassan-Montero and Herrero-Solana, 2006). Very similar to previous approaches by Kohonen et al. (2000); Lagus et al. (2004); Rauber and Merkl (2001) we derive automatic labels from the self-organized prototype vectors. In addition to the simple keyword sorting described above in Fig. 5.2 we introduce two more



(a)

(b)

Figure 5.7: The second example for the dynamic node rendering shows the visualization with four hierarchical levels in the display. In this case the dynamic node effect is more subtle than in Fig. 5.6.

components:

1. Each node carries ten keywords corresponding to the highest entries of its prototype vector. Since the rendering space is limited by the Poincaré disk, the number of actually displayed keywords for node a is given by:

   $$N_{\mathrm{k}}(a) = m \;\exp\left(-\frac{4\ \mathrm{arctanh}^2\left|\frac{z_a - z_0}{1-\bar{z}_a z_0}\right|}{R}\right), \tag{5.7}$$

   where $z_a \in \mathbb{D}$ is the position of node a and z_0 the position of the node closest to the origin of the Poincaré disk, R is the radius controlling the width of the labelling area, and m the maximal number of keywords which should be displayed per node.

2. The font sizes of the tag cloud elements are chosen according to the maximal component value of the corresponding node prototype vector. The heuristic is motivated by our employment of a scalar metric: The prototype vectors are always scaled to unit length, i.e. their component values range between 0 and 1. If the value of the maximal component is close to zero, there are a large number of non-zero entries in the prototype bag. Any attempt to label the map region will be difficult, because we would miss a non neglectable amount of important words. Consequently, we use a small font for the corresponding tag cloud entries. If, on the other hand, the maximal component is large, the chance that the corresponding words describe the map area rather faithfully is also larger. Therefore, a larger font is used.

The components provide hints where in the H^{2}SOM the topics are concentrated and where they can be described with a few keywords. The visual appearance of the resulting tag cloud resembles the appearance of tag clouds from Web 2.0 applications. The main - but very distinctive - difference is that in case of the H^{2}SOM the tag cloud data is not generated by human Web 2.0 input, but in a completely self-organizing manner driven by the hyperbolic neural network.

Note, that similar to the dynamic node visualization described above, also the tag cloud is generated on demand. While the user navigates within the data set in hyperbolic space, keywords appear and disappear as the user enters different regions of the map, resulting in a morphing tag cloud depending on the chosen focus.

Examples for the self-organizing tag clouds are given in Section 5.3 where we show the application of our framework to several real world datasets.

5.2.4 Visualizing the Evolution of Trends and Topics

By adding the third dimension to our visualization scheme, we obtain the opportunity to embrace a further view on the data: We can map the time dependend node activation potentials (refer to Section 4.3.1) to an elevation of the nodes above the Poincaré disk. By a continuous mapping of the time variant data the user sees a movie showing the evolution of document topics in time. Snapshots from a developing news peak in the Reuters-21578 corpus are shown in Fig. 5.8. A more thorough discussion on the interpretation of such a visualization is given in the applications section below.

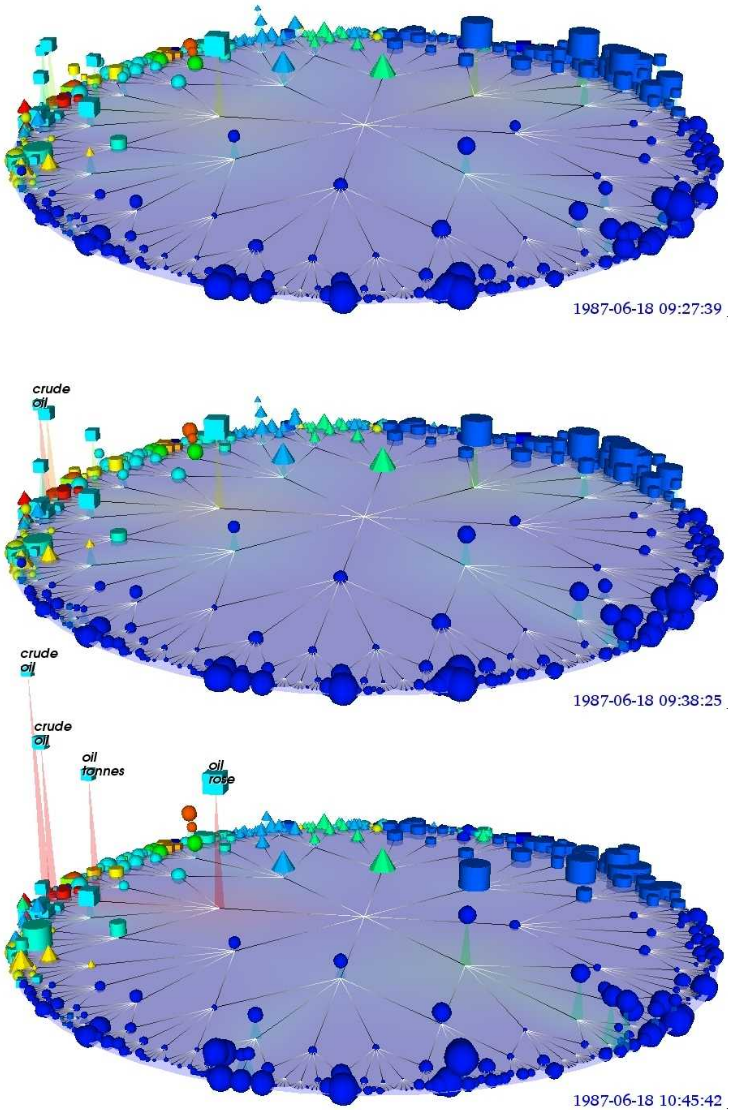


Figure 5.8: Animation of news activities through time. The three still images grabbed from a movie stream show a developing news peak in the left part of the map.

5.2.5 Semantic Searches

In standard information retrieval the search for information is most commonly realized by indexed search, i.e. an index is pre-build which consists of pointers to documents containing index words (Baeza-Yates and Ribeiro-Neto, 1999). When the user submits a search query, the pre-build index is searched for the query terms and documents containing these terms are returned sorted by their relevance. The approach scales very well but only retrieves documents which contain at least one of the query terms. Documents which cover the sought after topics but do not contain any of the query terms will not be found, i.e., classical IR is not able to recognize the semantics of a user query.

The H^{2}SOM offers a computational very efficient way to add semantics to user searches. Following classical information retrieval we utilize MySQL's fulltext indexing capability to index the complete document database. A search query is then executed in standard fashion: To each document d returned by MySQL as relevant to query q a relevance score S is attached with $0 < S(d,q) \leq 1$. For each query we then compute a node relevance score

$$R(a) = \sum_{d \in \mathcal{H}(a)} S(d,q), \qquad (5.8)$$

where $\mathcal{H}(a)$ is the set of documents for which node a is a best-match node. By visualizing the node relevance scores directly on the map, the user sees where the documents found by the classical IR approach get mapped to. The topology preserving capabilities of the H^{2}SOM enrich the search result with a strong semantic component: Documents which do not contain any of the search terms might still be found, since they reside within or in close vicinity to the marked areas.

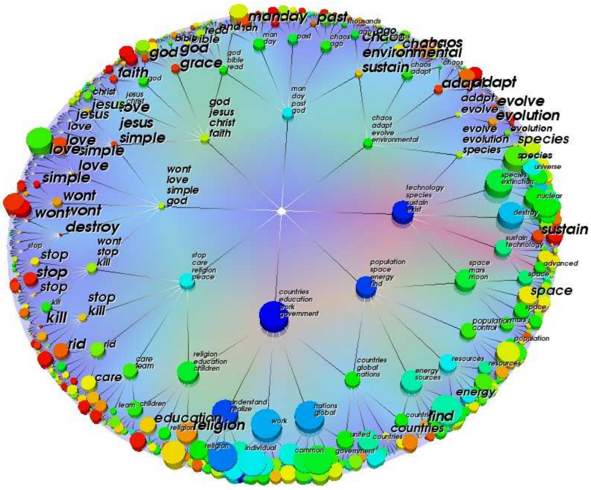


Figure 5.9: The combination of classical IR and the topologically ordered H^{2}SOM achieves semantic search capabilities.

5.3 Results on Real World Data

5.3.1 Stephen Hawking on Yahoo!-Answers

In July 2006 the physicist Stephen Hawking asked the following question on Yahoo!-Answers[1]: *"How can the human race survive the next hundred years? - In a world that is in chaos politically, socially and environmentally, how can the human race sustain another 100 years?"* A few days later Internet users had given more than 20.000 answers to his question.

The example shows that the Internet is able to attract and motivate a large number of persons to contribute with personal knowledge, views or experiences to a broad number of topics. However, due to the massive scale and the lack of any review process the information contained within the more than 20.000 answers is hard to digest for the average Internet user: On the Yahoo! website, the answers are displayed in a list form. In order to acquire an

[1] http://answers.yahoo.com

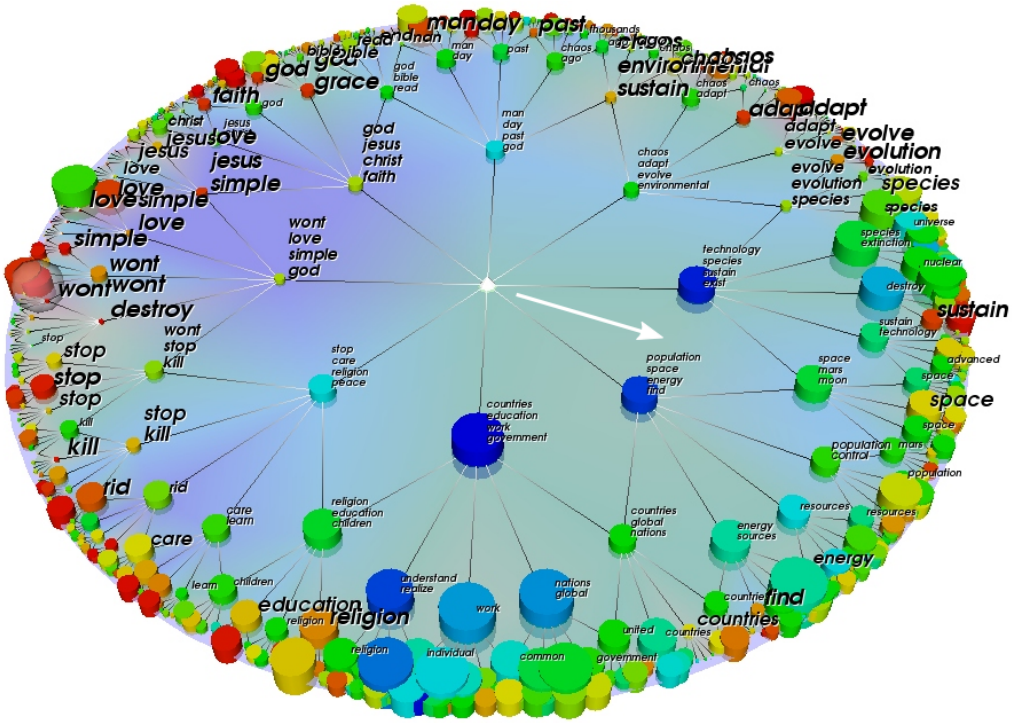


Figure 5.10: Visualization of 21.807 answers to Stephen Hawking's question "How can the human race survive the next hundred years?". The self-organizing tag cloud consists of different font sizes to indicate thematic clusters. At the left perimeter a node labelled "wont" is marked as selected. The superimposed arrow indicates the movement of the focal position leading to the next figure.

overview the user has to sample and read a substantial amount of the posted messages.

For the analysis by our H^{2}SOM framework we used a customized web crawler to retrieve and store some 21.807 answers which had been given up to that date. After processing the data with the text mining engine[1] (cf. Section 5.1.2), the visualization components show the map given in Fig. 5.10. We used the following mapping for the display:

- **Node sizes** reflect the normalized document numbers as given by Eq. 5.3.
- **Node colors** indicate the maximum component value of the corresponding prototype vector - on a HSV colorscale from blue to red. Note, that this is the same variable determining the font sizes within the tag cloud.
- **Ground colors** show the average node distances in the feature space: Nodes within blue areas are far away from each other, whereas nodes in red areas are similar to each other.

The centered map in Fig. 5.10 presents all 21.807 messages with a single image. The tag cloud provides a broad overview of the predominant topics. Visually noticeable is a cluster of large tags in the left part of the map containing *god, love, jesus, won't* and *stop*. In order to inspect the map content on a single document basis, the user can select nodes. The selection is indicated as a transparent sphere drawn around the selection such as shown in Fig. 5.10 on the left. A node selection initiates a SQL query and retrieves all messages for which the

[1] On a standard PC with a single 2 GHz processor, the processing time is about 20 minutes.

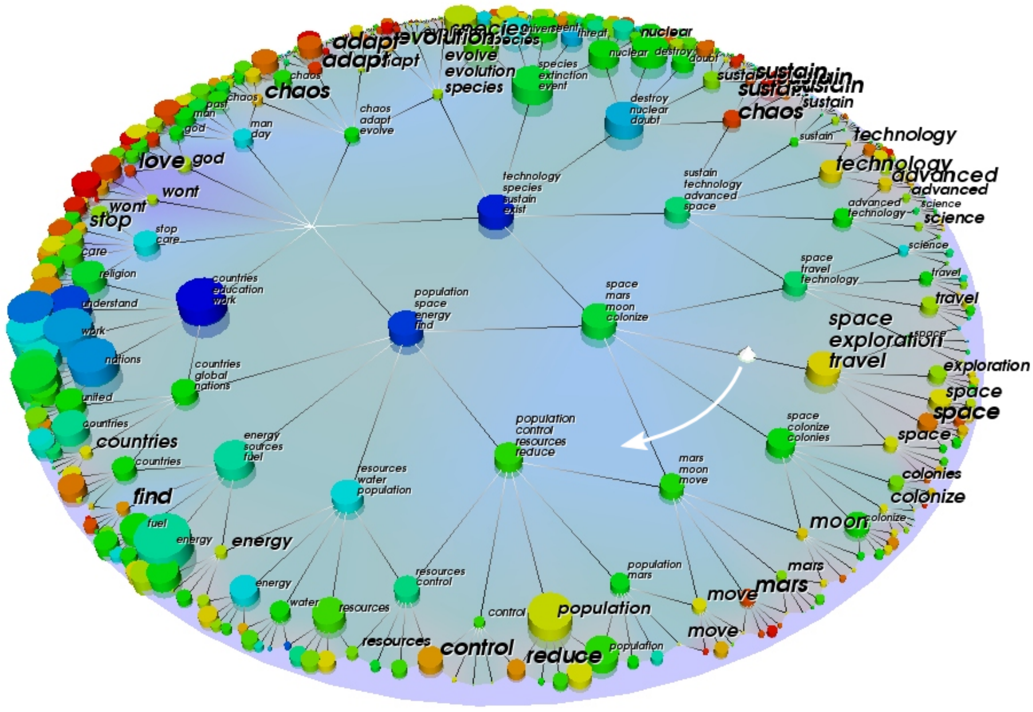
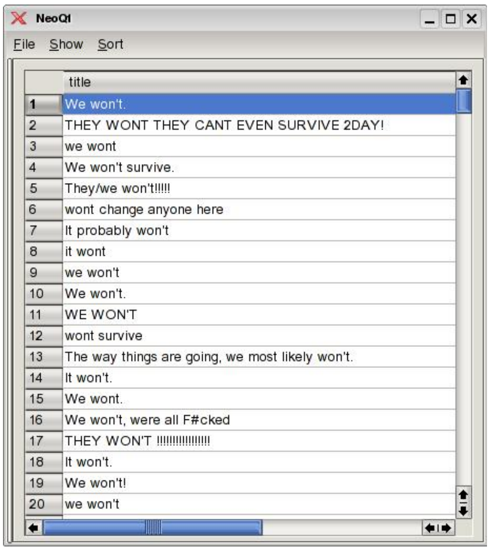


Figure 5.11: The user has selected a focus position towards the "space" cluster at the 4 o'clock position in the map. A keystroke initiates a rotation around the origin of $I\!H^2$ as indicated by the arrow, allowing to inspect the whole document collection at the chosen semantic level.

selected node is a best-match node and displays them in list form as shown in Fig. 5.12. In our example the *won't* cluster consists of more than 300 very simple posts containing only a few words. The first 20 of them are shown in the screenshot of Fig. 5.12. As the listing shows, the messages are not very substantial and could be easily deleted by another mouse click. This shows how the interface alleviates the cleaning of large collections of documents to get rid of those posts which are not interesting for the reader.



	title
1	We won't.
2	THEY WONT THEY CANT EVEN SURVIVE 2DAY!
3	we wont
4	We won't survive.
5	They/we won't!!!!!
6	wont change anyone here
7	It probably won't
8	it wont
9	we won't
10	We won't.
11	WE WON'T
12	wont survive
13	The way things are going, we most likely won't.
14	It won't.
15	We wont.
16	We won't, were all F#cked
17	THEY WON'T !!!!!!!!!!!!!!!
18	It won't.
19	We won't!
20	we won't

Figure 5.12: List of messages for a selected node.

In order to get an overview of the more substantial postings the user can move the focal position of the map towards the perimeter to display the content on a finer semantic scale. Fig. 5.11 shows an exemplary snapshot where the user has moved his attention towards the *space exploration* cluster at the 4 o'clock position on the map. An additional mouse binding facilitates the exploration of the map at any given semantic level: Mouse wheel rotation is mapped to a rotation of the focal position around the origin of the hyperbolic plane $I\!H^2$. A complete animation which revolves the focus around 360 degrees can be initiated by keystroke. A 360° animation sequence at the semantic scale shown in Fig. 5.11 takes approximately 100 seconds[1]. Thus, the user is able to "drive through" all

[1] several H^2SOM movies can be viewed at http://www.techfak.uni-bielefeld.de/ags/ni/projects/textvis/

21.807 messages in less than two minutes obtaining an overview of the topics and their distribution on the chosen semantic level.

In order to allow for a quantitative analysis, the tag cloud can be extended by percental values, i.e. “wont” becomes “5.2% wont”, describing that 5.2% of the messages in the corresponding semantic level carry the tag “wont”. The following compilation reflects the hierarchic mapping which has been learned by the self-organizing process for the answers to Hawking’s question:

A 8% of the posters suggest that mankind will be able to adapt to changing environments.

B 69% of the answers can be broken apart into the following topics:

  a) 34% broach the issues of “our species”, “technology”, “space” and “energy and other resources”:
     i. 13% talk about a global catastrophy leading to the extinction of our species.
     ii. 12% are concerned in the control of resources. The topic is evenly distributed between “birth control”, “water and food”, and “energy”.
     iii. 4% trust that technological developments will ensure our survival.
     iv. 5% expect a colonization of space.
  b) 21% of the answers are quite substantial and discuss more complex themes. The vocabulary of this group is the largest of all (measured by the number of non-zero components in the bag-of-words representation).
     i. 10% discuss the topic of the United Nations, a global government and the complex relation between states. A small proportion of this cluster covers the field of “terrorism”.
     ii. 11% focus on the individual. Issues are greed, individual responsibilities and the pursuit of knowledge.
  c) 14% can be subsumed by the main topics “education” and “understanding”: the teaching of children, understanding and caring for each other, achieving peace and stopping war and environmental destruction.

C 5% do not expect that mankind will survive the next 100 years. The postings within this cluster are characterized by simple word distributions. Many of them are modifications of *“we won’t”* and contain only a few words.

D 8% trust in “god” and “belief”. With respect to the word distribution in the bag-of-words representation these messages again do contain only relatively few words.

E 10% of the messages are mapped to a region which can not be described by a few words. Several messages are French or German and only build coherent clusters at a deeper semantic level. E.g., there are about 30 messages saying: *“Like they always do one day at a time”.*

Semantic Search

The semantic search capabilities were described in Section 5.2.5 where Fig. 5.9 shows the results of a classical IR query for the term “science” superimposed on the map surface. The node relevance score $R(a)$ given by Eq. 5.8 is mapped via a HSV blue to red colorscale onto the map surface with the color red indicating a high proportion of relevant documents to the query. In Fig. 5.13 the user has zoomed into the deeper hierarchies of the H^{2}SOM while following the red trail leading to the classical IR search results. The tag cloud in Fig. 5.13

shows that the region containing most of the relevant documents is indeed labelled with a *science* tag. The neighboring regions are tagged with *advanced technology*, *travel*, *space*

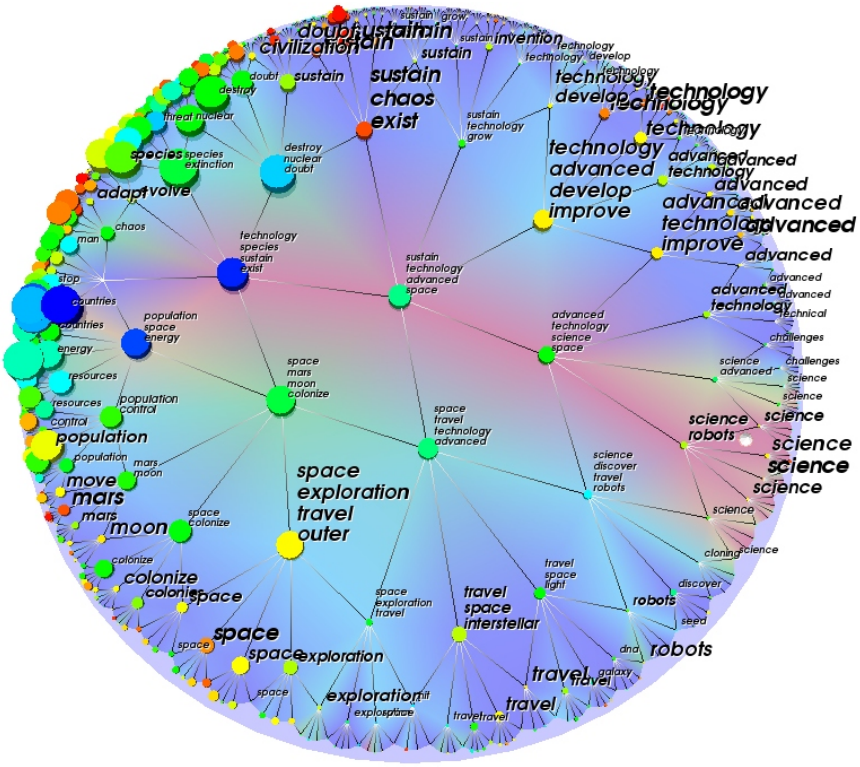


Figure 5.13: Visualizing semantic searches: The results from a classical information retrieval search of the term "science" are superimposed as a colorscale on the map ground. The user has followed the semantic path and has zoomed into the "science" region.

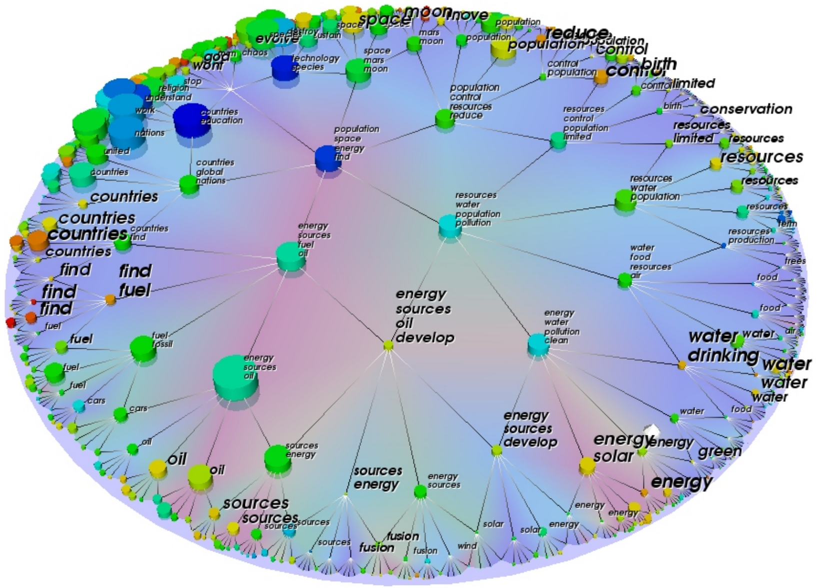


Figure 5.14: Another example for a semantic search: In this case the user submitted the query "energy" and has subsequently moved the focus towards the highlighted region. The self-organizing tag cloud annotates the corresponding map area in which users contributed to the semantics of the term "energy".

exploration, *colonize*, *moon*, and *mars*. The example illustrates the benefits of the semantic organization by the hierarchical self-organizing map: If searching for views on the impact of science on the survival of the human race, the combination of classical IR and semantic guidance through the self-organizing tag cloud does not only reveal documents containing the word “science”, but also closely related posts talking about advanced technologies and the colonization of space - topics in tight associated with science. Fig. 5.14 shows a second example where the user asked for user contributions related to the term “energy”.

5.3.2 Reuters-21578

The quantitative evaluation of the H^2SOM on the Reuters data in Section 4.1.4 and 4.2.3 has shown that with respect to classification accuracy the H^2SOM offers very competitive results. In contrast to the Hawking data where all documents have been published within a very short timeframe, the Reuters data offers a collection of documents from a broader time span. In the following we give an example how the time dependend node activation potentials might be utilized to detect up-and-coming events within news streams.

The Reuters data is separated into a training set containing news wire messages from the 26th of February 1987 to the 7th of April 1987, and a test set from the 8th of April 1987 to the 20th of October 1987. For the here reported results we have trained the H^2SOM with messages originating from the training set and demonstrate the topic detection capabilities with data from the test set being “new” to the system.

As described in Section 5.2.4 the user interface provides the capability to visualize an incoming document stream as an animation directly reflecting the dynamics of messages with respect to their arrival time.

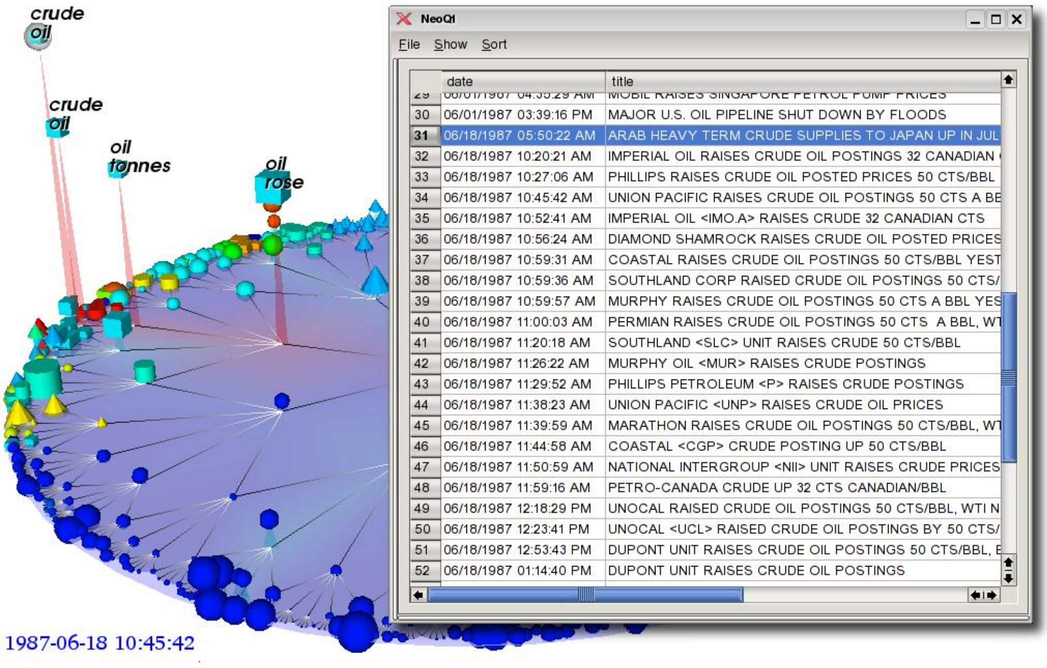
The human eye is very sensitive to moving objects which share a “common fate” (Ware, 2004). In case of developing news topics which share a common semantic background they also share a set of common nodes in the H^2SOM hierarchy. In an animation these nodes are rising in parallel and thus attract the observers attention. An example for such a developing situation is shown in the three still images in Fig. 5.8. By selecting a peaked node the user is able to retrieve the messages responsible for the risen activity. Fig. 5.15 shows the selected node and the corresponding list of messages sorted by their arrival time. The system automatically selects the timestamp corresponding to the current timestamp in the animation. From the list in Fig. 5.15 it becomes apparent, that the news peak was caused by a boost of newswire stories stating the raise of crude oil prices by several companies almost at the same time.

5.3.3 aid infodienst

In this section we describe a case study which was carried out in close cooperation with the “aid infodienst”[1], a non-profit organization co-funded by the German Federal Ministry of Food, Agriculture and Consumer Protection.

The “aid infodienst” offers an information service run by a team of experts for health and food which is open to the general public. After an online registration any user can ask personal questions in one of nine forums titled as “All about Weight”, “Food in Pregnancy”, “Food Allergy” and others. An expert answers the question within a timeframe of 48 hours and question and answer are published in the forum - if the questioner agreed to a publication.

[1] http://www.was-wir-essen.de/impressum.php



	date	title
29	06/01/1987 04:55:29 AM	MOBIL RAISES SINGAPORE PETROL PUMP PRICES
30	06/01/1987 03:39:16 PM	MAJOR U.S. OIL PIPELINE SHUT DOWN BY FLOODS
31	06/18/1987 05:50:22 AM	ARAB HEAVY TERM CRUDE SUPPLIES TO JAPAN UP IN JUL
32	06/18/1987 10:20:21 AM	IMPERIAL OIL RAISES CRUDE OIL POSTINGS 32 CANADIAN (
33	06/18/1987 10:27:06 AM	PHILLIPS RAISES CRUDE OIL POSTED PRICES 50 CTS/BBL
34	06/18/1987 10:45:42 AM	UNION PACIFIC RAISES CRUDE OIL POSTINGS 50 CTS A BE
35	06/18/1987 10:52:41 AM	IMPERIAL OIL <IMO.A> RAISES CRUDE 32 CANADIAN CTS
36	06/18/1987 10:56:24 AM	DIAMOND SHAMROCK RAISES CRUDE OIL POSTED PRICES
37	06/18/1987 10:59:31 AM	COASTAL RAISES CRUDE OIL POSTINGS 50 CTS/BBL YEST
38	06/18/1987 10:59:36 AM	SOUTHLAND CORP RAISED CRUDE OIL POSTINGS 50 CTS/
39	06/18/1987 10:59:57 AM	MURPHY RAISES CRUDE OIL POSTINGS 50 CTS A BBL YES
40	06/18/1987 11:00:03 AM	PERMIAN RAISES CRUDE OIL POSTINGS 50 CTS A BBL, WT
41	06/18/1987 11:20:18 AM	SOUTHLAND <SLC> UNIT RAISES CRUDE 50 CTS/BBL
42	06/18/1987 11:26:22 AM	MURPHY OIL <MUR> RAISES CRUDE POSTINGS
43	06/18/1987 11:29:52 AM	PHILLIPS PETROLEUM <P> RAISES CRUDE POSTINGS
44	06/18/1987 11:38:23 AM	UNION PACIFIC <UNP> RAISES CRUDE OIL PRICES
45	06/18/1987 11:39:59 AM	MARATHON RAISES CRUDE OIL POSTINGS 50 CTS/BBL, WT
46	06/18/1987 11:44:58 AM	COASTAL <CGP> CRUDE POSTING UP 50 CTS/BBL
47	06/18/1987 11:50:59 AM	NATIONAL INTERGROUP <NII> UNIT RAISES CRUDE PRICES
48	06/18/1987 11:59:16 AM	PETRO-CANADA CRUDE UP 32 CTS CANADIAN/BBL
49	06/18/1987 12:18:29 PM	UNOCAL RAISED CRUDE OIL POSTINGS 50 CTS/BBL, WTI N
50	06/18/1987 12:23:41 PM	UNOCAL <UCL> RAISED CRUDE OIL POSTINGS BY 50 CTS/
51	06/18/1987 12:53:43 PM	DUPONT UNIT RAISES CRUDE OIL POSTINGS 50 CTS/BBL, E
52	06/18/1987 01:14:40 PM	DUPONT UNIT RAISES CRUDE OIL POSTINGS

Figure 5.15: Snapshot from an animated news stream showing the evolution of topics in time. At the depicted timestamp the user has stopped the animation and selected a peaking node in the left part of the map.

The general procedure for the members of the expert team so far was to read all questions asked by the Internet users and then giving their answer for those questions for which he or she is the expert. In 2007 the users asked about 10 questions per day in average. Quite commonly the same or similar questions are asked more than once. Either because a search did not return any relevant hits, or because the questioner did not search for already asked questions at all. The “aid infodienst” was therefore looking for an automated text classification system to assist in the following situations:

- Any new question should be categorized and tagged with one of several predefined topics to alleviate the selection process for the experts, i.e. the system should be able to route new questions directly to the responsible expert.

- If a question is selected for answering by the expert, the system should retrieve all similar questions asked in the past to assist the expert in finding already answered questions.

- Directly after a user has submitted a new question, the system should retrieve similar questions from the past. If a similarity threshold is reached, the found question/answer pairs should be presented to the user. If the user then acknowledges the helpfulness of a retrieved answer the new question should be automatically sorted into the old forum thread - disburden the experts to answer this already answered question.

Data Situation

We extracted about 15.000 question/answer pairs from the forum database going back to the year 2002 when the public service started. During the initial project phase the question was raised, what kind of training data would lead to a better generalization:

1. Train the system with the user questions only, hoping for a better generalization for the intended use, i.e. in finding similar old questions to newly asked ones.

2. Train the system with the user questions and their corresponding expert answers. By doing so the system could benefit from the expert's knowledge and vocabulary and "learn" a better internal representation.

In order to systematically evaluate this question, we trained two H^{2}SOM maps with our standard architecture (cf. Section 5.1.2): One was trained with the user questions only, the other was trained with the corresponding question/answer pairs. We then asked the experts to evaluate the quality of the retrieval sets. During the experiment the experts did not know what kind of training set was actually used for the presented results. The outcome of the expert evaluation is given in Chapter 6.

The resulting map obtained with the combined question/answer training data is shown in Fig. 5.16. The training data did not contain any information about the forum to which the question was assigned to by the experts. Nevertheless, the mapping in Fig. 5.16 shows that the documents cluster well with respect to their assigned category. The well-tempered clustering allows for a simple labelling scheme where map sectors are defined by manually identifying border nodes in the deepest hierarchy of the H^{2}SOM. An example for the sector label "Food in Pregnancy" is given in Fig. 5.16 below. All new questions which are mapped to this area can then be automatically assigned to the corresponding expert.

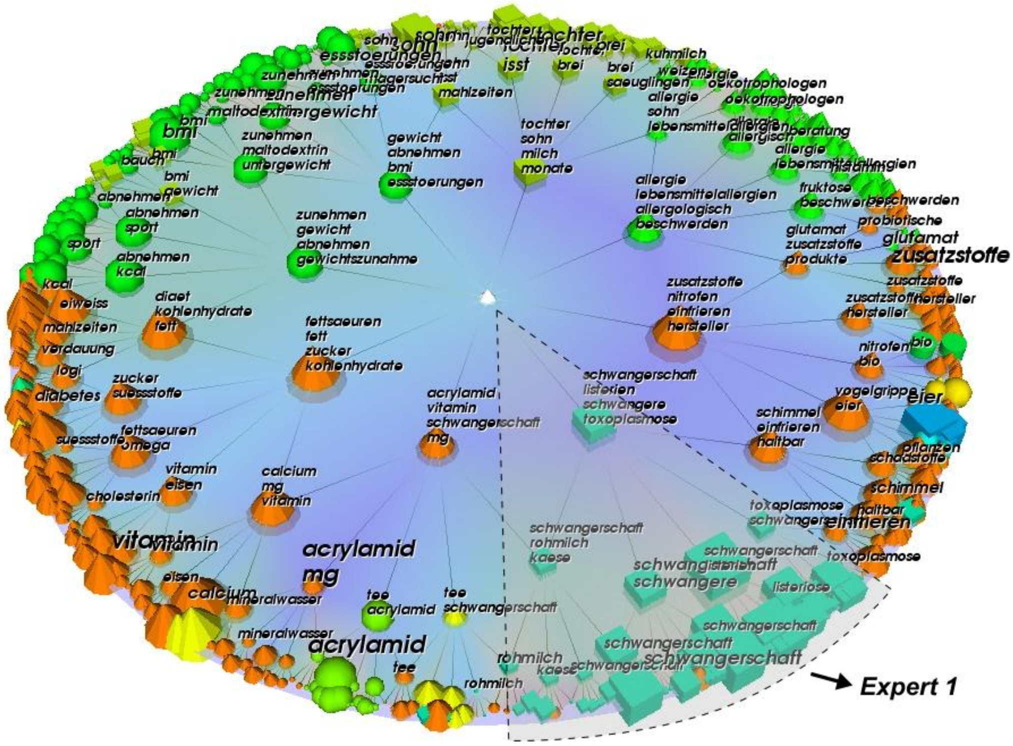
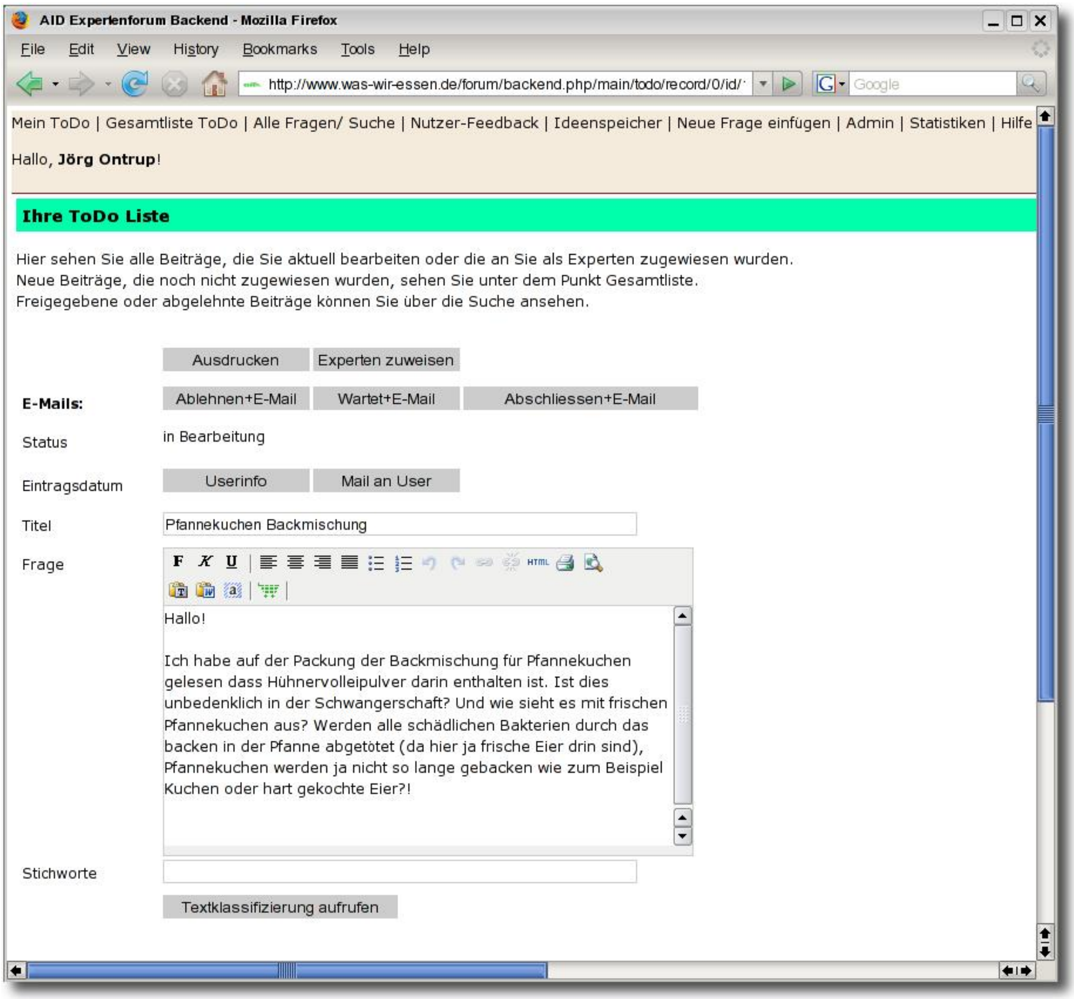


Figure 5.16: Visualization of 15.000 user questions to the "aid-infodienst". The node symbols in the mapping correspond to categories such as *"All about Weight"* (dark green spheres), *"Food in Pregnancy"* (cyan cubes) or *"Food Allergy"* (green cones). The orange cones correspond to the general category *"You ask, aid answers"* and can be divided into several subclusters on the map. Additionally superimposed on the image is a manual sector label, indicating that all messages from that area will get routed to the specific expert for that area.

Web Interface

Since the hyperbolic user interface (cf. Section 5.2) is not yet available as an easily installable software package for the Windows operating systems, we developed a web base application interface allowing to access the key functions of the text mining engine with a standard web browser. By offering the text classification service based on well established standard technologies, the work flow for the experts was only marginally changed. Fig. 5.17 shows a web browser screenshot of an open expert session in the forum backend.


AID Expertenforum Backend - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.was-wir-essen.de/forum/backend.php/main/todo/record/0/id/
Google

Mein ToDo | Gesamtliste ToDo | Alle Fragen/ Suche | Nutzer-Feedback | Ideenspeicher | Neue Frage einfügen | Admin | Statistiken | Hilfe

Hallo, **Jörg Ontrup**!

Ihre ToDo Liste

Hier sehen Sie alle Beiträge, die Sie aktuell bearbeiten oder die an Sie als Experten zugewiesen wurden.
Neue Beiträge, die noch nicht zugewiesen wurden, sehen Sie unter dem Punkt Gesamtliste.
Freigegebene oder abgelehnte Beiträge können Sie über die Suche ansehen.

E-Mails:

Status: in Bearbeitung

Eintragsdatum

Titel: Pfannekuchen Backmischung

Frage: Hallo!

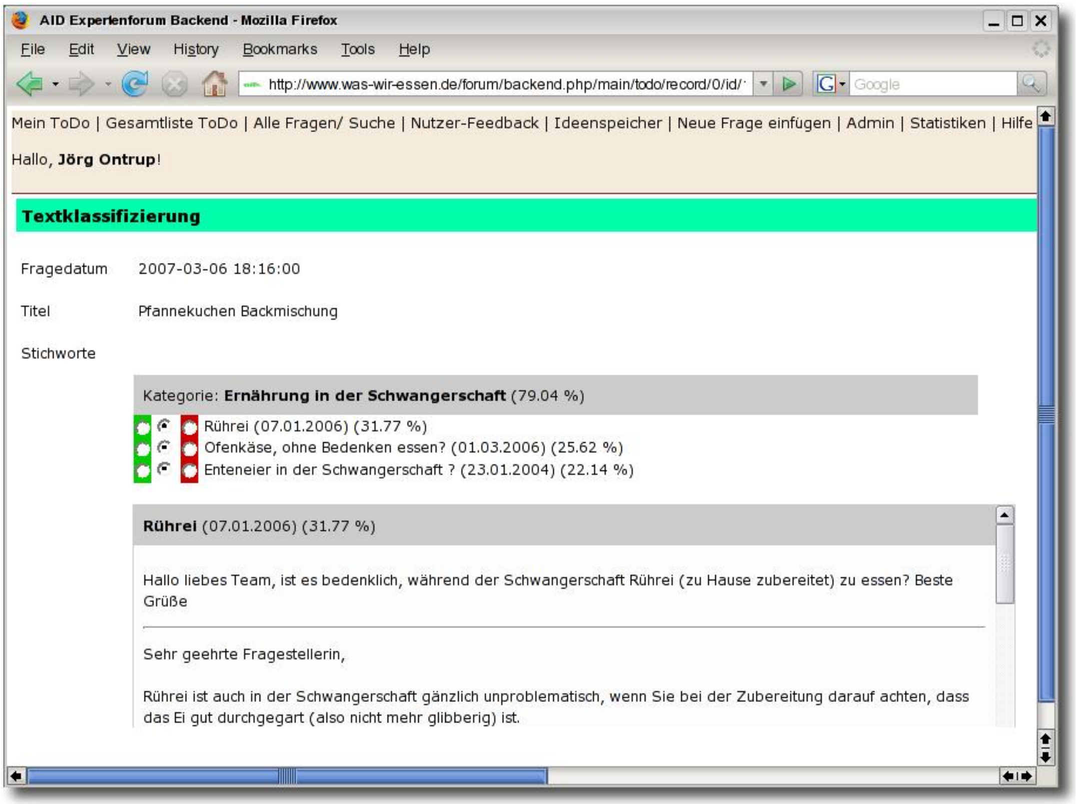
Ich habe auf der Packung der Backmischung für Pfannekuchen gelesen dass Hühnervolleipulver darin enthalten ist. Ist dies unbedenklich in der Schwangerschaft? Und wie sieht es mit frischen Pfannekuchen aus? Werden alle schädlichen Bakterien durch das backen in der Pfanne abgetötet (da hier ja frische Eier drin sind), Pfannekuchen werden ja nicht so lange gebacken wie zum Beispiel Kuchen oder hart gekochte Eier?!

Stichworte

Figure 5.17: The screenshot shows a web browser where the expert is working on an open user question about the risks of pancakes (containing fresh eggs) during pregnancy.

By pressing a button, the open question is submitted to the text mining engine which returns a list of similar questions asked in the past. To this end, the document is converted to the pyramid-of-words representation and the best-match node in the H^{2}SOM is searched using the fast tree search (cf. 4.1.3). Then all previous documents for which the found node is also a best match node are returned together with a relevance score given by $1 - \langle d_q, d_p \rangle$, where d_q and d_p are the vector representations for the query and the previous document, respectively.

Fig. 5.18 shows the results for the example from above. The open question was classified as *"Food in Pregnancy"* (refer to Section 3.5 for the classification process) and three similar questions from the past were returned by the system. Note, that the suggestions for matching similar questions can be marked by the expert as “helpful”, “neutral”, and “not helpful”. The data was collected during a three month evaluation phase and consits of approximately 1200 single expert ratings. The results for this user study are given in Section 6.3 of the next chapter.


AID Expertenforum Backend - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.was-wir-essen.de/forum/backend.php/main/todo/record/0/id/
Google
Mein ToDo | Gesamtliste ToDo | Alle Fragen/ Suche | Nutzer-Feedback | Ideenspeicher | Neue Frage einfügen | Admin | Statistiken | Hilfe

Hallo, **Jörg Ontrup**!

Textklassifizierung

Fragedatum	2007-03-06 18:16:00
Titel	Pfannekuchen Backmischung
Stichworte	

Kategorie: **Ernährung in der Schwangerschaft** (79.04 %)

○ ● ○ Rührei (07.01.2006) (31.77 %)
○ ● ○ Ofenkäse, ohne Bedenken essen? (01.03.2006) (25.62 %)
○ ● ○ Enteneier in der Schwangerschaft ? (23.01.2004) (22.14 %)

Rührei (07.01.2006) (31.77 %)

Hallo liebes Team, ist es bedenklich, während der Schwangerschaft Rührei (zu Hause zubereitet) zu essen? Beste Grüße

Sehr geehrte Fragestellerin,

Rührei ist auch in der Schwangerschaft gänzlich unproblematisch, wenn Sie bei der Zubereitung darauf achten, dass das Ei gut durchgegart (also nicht mehr glibberig) ist.

Figure 5.18: Results obtained from the text classification system for the example in Fig. 5.17.

5.4 Summary

In this chapter we have presented the design elements of an interactive text visualization system consisting of four central building blocks:

1. A database engine responsible to store raw-, feature- and self-organized meta data.
2. A textmining engine for the generation of the pyramid-of-words representation for document collections and its self-organization by the H^{2}SOM.
3. An interactive 3D visualization framework offering a natural *focus & context* navigation scheme allowing the user to browse through documents at a specific chosen semantic resolution. In addition the generated mappings are augmented by self-organizing tag clouds offering semantic information scents.
4. A web application interface allowing to integrate the capacities of the H^{2}SOM framework into third party web applications.

The operation of the system has been demonstrated on three real world case studies. First, more than 20,000 messages from an Internet forum have been analyzed. We have shown that the hierarchical self-organization process and the tag cloud augmented visualization allows the user to quickly obtain an overview of the message structure. We have presented a semantic extension to the classical information retrieval process offering the opportunity to retrieve results which are relevant to the semantic concepts of a user query, but which do not necessarily need to contain any of the used query terms. Second, for the Reuters-21578 newswire collection we have given an example how the time dependent node activation potentials reflect and visualize the evolution of topics within document streams, enabling the user to detect up-and-coming developments. Third, we have shown the employment of the web application interface within the setting of a food & health recommendation system. Experts answering user questions utilize the system for an automated routing of new incoming questions to the responsible expert. In addition they are able to dig out similar questions to newly asked ones from the past. Throughout the employment of the system the experts have performed a system evaluation which is presented in the next chapter.

Chapter 6

Empirical User Studies

In the previous chapter we have presented the overall architecture of the H^{2}SOM based visualization system. We have shown how the interactive 3D visualization engine utilizes the peculiar geometric properties of hyperbolic space as discussed in Chapter 3, and how the focus & context methodology allows a user to browse through the self-organized hierarchical data structure. Additionally, we have given examples on the application to three real world datasets.

In this chapter we present two empirical user studies addressing two questions:

1. How effective is the hyperbolic focus & context interface with respect to navigation tasks in large data structures?
2. Are hierarchical semantic maps able to retrieve knowledge from unstructured text data which is comparable to expert knowledge?

In order to answer the first question we define a task scenario which is constructed to measure two key questions: *(i)* how efficiently are users in reaching a predefined target in a hierarchical search space, and *(ii)* how efficiently are users in exploring a hierarchical data structure to answer task specific questions? We examine two navigation methodologies for changing the focus in the hyperbolic plane and contrast the results to a control group using a standard tree browser. The evaluation shows that for simple tasks there is no significant difference for all groups either in task completion time or task accuracy. However, in the case of complex tasks, the hyperbolic approach was found to be significantly faster and more accurate.

In order to answer the second question we have asked food and health experts from the "aid infodienst" (cf. 5.3.3) to evaluate the helpfulness and quality of retrieval results obtained by the H^{2}SOM.

6.1 Previous User Studies

Although there has been extensive research on designing and implementing focus & context techniques, cf. Section 2.3 for an overview, there have been only a little number of empirical user studies.

6.1.1 Hyperbolic Tree Browser

The most prominent implementation of a hyperbolic focus & context visualization system, the Hyperbolic Tree Browser by Lamping and Rao (1994) has been evaluated in four empirical studies:

(i) In an early experiment, Lamping et al. (1995) set up a scenario where four subjects had to locate a specific node in a hierarchical tree of world-wide-web pages, e.g. by the name of the page. They contrasted the hyperbolic browser with a conventional 2D scrolling tree browser and could not find any significant differences between the two browsers.

(ii) Later, Czerwinski and Larson (1997) evaluated an improved version of the Hyperbolic Tree Browser and could also not find any performance gains for simple browsing tasks relative to a Microsoft-like standard browser.

(iii) Mullet et al. (1997) organized a competitive browser contest at the meeting of the ACM's Special Interest Group on Computer-Human Interaction (CHI '97). They asked 16 subjects who could choose between six browsers to locate items within a large tree hierarchy (see Fig. 6.1). In contrast to the previous two experiments, in this study subjects using the Hyperbolic Tree Browser gained substantially superior results. The runner-up in the contest was the Explorer file browser from Microsoft.

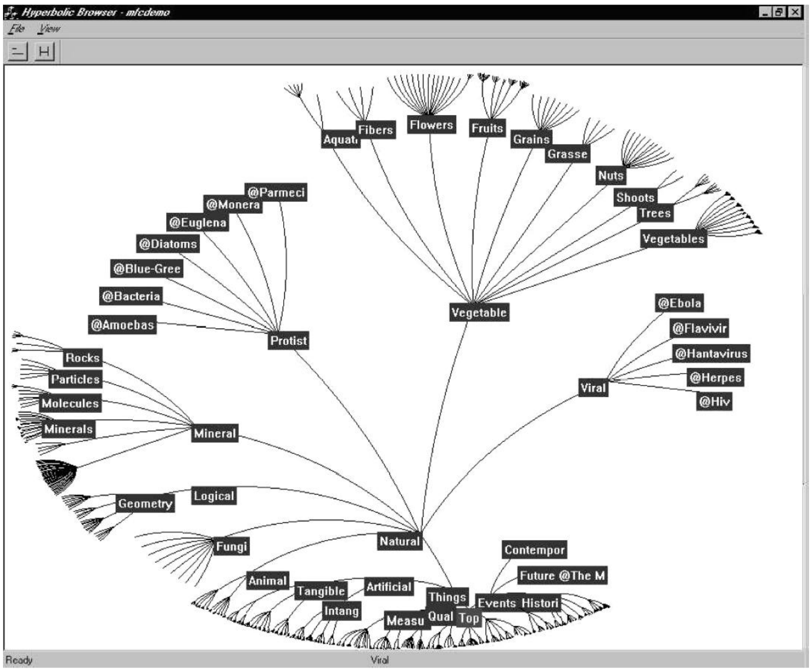


Figure 6.1: The Hyperbolic Tree Browser displaying a portion of the hierarchical dataset used in the studies by Mullet et al. (1997) and Pirolli et al. (2003). Subjects were asked to locate and "double-click" specific nodes by formulating tasks such as "Find the Ebola virus".

(iv) Since the above results were contradictorily, Pirolli et al. (2003) have conducted a more sophisticated study in order to understand why the Hyperbolic Tree Browser performed so well in the competition at CHI '97 but failed to show any significant performance boosts at the previous laboratory experiments. In a first experiment 8 participants were completing the same tasks as in the CHI '97 contest using both the Hyperbolic Tree Browser and the Microsoft Explorer. They found no significant differences between the two browsers, but did find that inter-subject performances varied much stronger than inter-browser performances. A task analysis with further 48 participants showed that some of the CHI '97 tasks were

formulated in a misleading way: The information cues positioned at each node lead many users to explore a wrong subbranch for a given question, therefore not finding a valid answer in the given, limited time. In a second experiment they selected only those tasks for the experiment which were consistent with the given node labels. During the task completion the visual search of the 8 participants was also logged with an eye-tracking apparatus. The results of this second experiment showed that the "Hyperbolic Tree Browser yielded faster overall search performance" than the Microsoft lookalike file browser and that "Hyperbolic users examined more of the tree nodes at a faster rate". Pirolli et al. (2003) conclude "that performance with the Hyperbolic Tree is greatly enhanced by clear information scent [...] (and) the superiority (or lack of superiority) of the Hyperbolic Tree over conventional tree browsers depends on task conditions such as type of task or information scent cues."

6.1.2 Impacts on Study Design

The results of the previous user studies on hyperbolic focus & context browsers are ambivalent: Some studies do find significant differences to conventional browsers, some do not. In the following we discuss impacts the previous experiments had on our study design.

Most of the experiments mentioned above have been based on the *task completion time* as their sole descriptive variable. However, this somehow neglects the *accuracy* with which the tasks have been solved (if they have been solved at all). Especially for hard problems, users tend to click on arbitrary solutions in order to get to the next question. Therefore, we do not only measure the *time*, but also the *correctness* for each task.

Due to a limited resource of participants, some of the above studies have asked subjects to complete the experiments with both browser types in question. This leads to training effects, such that subjects generally perform better when completing a set of tasks for a second time. By randomizing the order of browser presentation the training effect can be leveled out to some extent, but not completely. Thus, participants in our experiment were randomly assigned to one control group and completed the set of tasks only once; therefore, reducing the intra-subject training effect to a minimal level.

6.2 Navigation Efficiency H^{2}SOM vs. Tree Browser

6.2.1 Methodology

Participants

Thirty six undergraduate and PhD students participated in the study. None of them had previous experience with hyperbolic visualization systems. All of them were accustomed to computer-use and were familiar with keyboard and mouse interactions. In order to buildup the subject's motivation it was pointed out that the best participant could either win a book coupon for a local bookstore or a bottle of quality red wine.

Experimental Setup

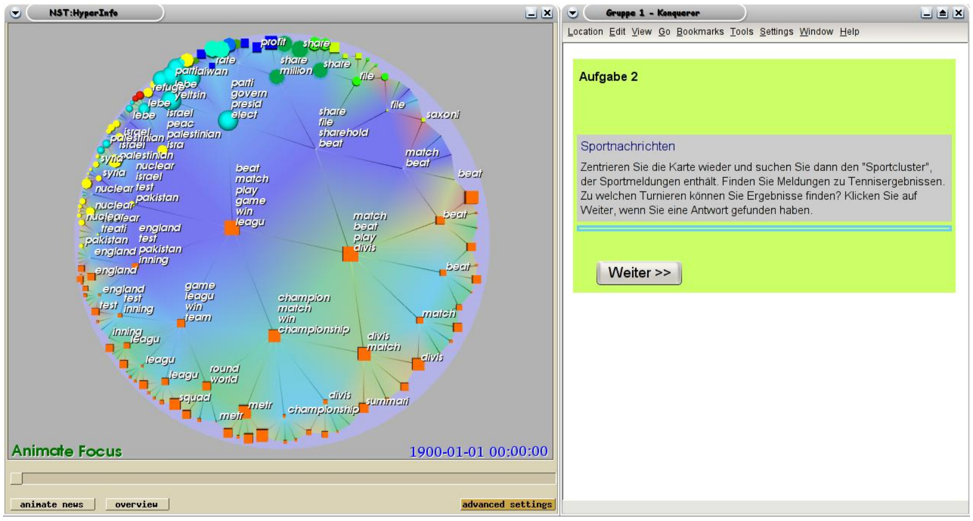
For the user study we deployed an experimental setup constituted of the following parts:

1. The participants were placed at a Pentium 4 based standard-PC connected to a 19-inch LCD flat panel display. Two programs were running simultaneously on this machine.
    a) A web browser displaying an online questionnaire with problem assignments the users had to solve. Additionally, the questionnaire provided answer forms which the users had to fill in.

b) The graphical user interface to solve the assignments. Depending on the control group the users were either presented with the H^2SOM or a traditional tree browser.

2. The online questionnaire was designed with the *Socrates Engine*[1], an open source framework which allows to define questionnaires in XML format. It was run inside a Java servlet container provided by the *Apache Tomcat* [2] system.

3. All information captured by the Socrates engine was directly transferred to a MySQL database server, which allowed for detailed offline evaluation of user answers and reaction times.



Aufgabe 2

Sportnachrichten

Zentrieren Sie die Karte wieder und suchen Sie dann den "Sportcluster", der Sportmeldungen enthält. Finden Sie Meldungen zu Tennisergebnissen. Zu welchen Turnieren können Sie Ergebnisse finden? Klicken Sie auf Weiter, wenn Sie eine Antwort gefunden haben.

Figure 6.2: Screenshot of the user interface presented to the participants. On the left side, the graphical user interface of the evaluated browser is displayed. In this image, the H^2SOM browser is shown with the Reuters dataset. On the right hand side, a task assignment page from the online questionnaire can be seen.

Procedure

The participants started the experiment by clicking on a single link on the screen. They were then randomly assigned to one of three control groups particularized in Section 6.2.2 below. In addition to the online questionnaire either the hyperbolic or the traditional system was then presented to the user depending on the assigned control group.

During the experiment, three different types of pages were used within the questionnaire:

1. Detailed instruction pages described how to operate the user interface on the left hand side of the screen. In a tutorial-like manner the users were familiarized with the presented browsing methodology.

2. Problem assignments asked the users to solve specific tasks.

3. Answer forms provided a means to capture the user's performance on each task.

At the bottom of the questionnaire the users were able to click on a *Next* button in order to get to the next page. Each page transition was recorded with a time stamp, providing a means to determine the duration the users have stayed on each page.

[1] http://socrates-qe.sourceforge.net

[2] http://jakarta.apache.org/tomcat/

6.2.2 Control Groups

Group I & II: H^{2}SOM Browser

Control group I and II used the H^{2}SOM browser as introduced in Chapter 5. The only difference between the two groups was the methodology employed for the adjustment of the fish-eye fovea (cf. Section 5.2.2).

Group I used the “driving” variant, where the fish-eye position is adjusted in a continuous manner by mouse operations similar to the control of a figure in a computer game: When the user moves the mouse, the fish-eye is also moved into the same direction. Depending on the distance the mouse moves, the velocity of the fish-eye movement changes accordingly. For details, see Section 5.2.2.

Group II worked with the “drag” variant. Here, the mouse can be used to drag any position of the hyperbolic plane to a new target destination. That is, an outer region of the map can be moved into the center by dragging it to the origin of the hyperbolic plane. For further details, see Section 5.2.2.

Group III: Traditional Tree Browser

The nodes of the H^{2}SOM are organized in a strict hierarchical way. Therefore, an equivalent way to display the same data is by means of a traditional tree browser. This type of interface where the user can expand or collapse nodes is commonly used to display a computer’s file system. Consequently, most computer users are familiar with this approach to navigate through hierarchic data structures, and we believe it represents a good baseline for an evaluation of alternative approaches such as the H^{2}SOM .

The implementation of the tree browser used in our study is based on the standard tree view widget of the Qt-library[1]. For each node, the user can expand or collapse the corresponding hierarchic level by clicking a “+” or “-” icon displayed in front of the node. Additionally, a context menu provided means to expand or collapse a whole subtree within the hierarchy. A screenshot of the interface displaying the tree browser is shown in Figure 6.3.

6.2.3 Task Assignments

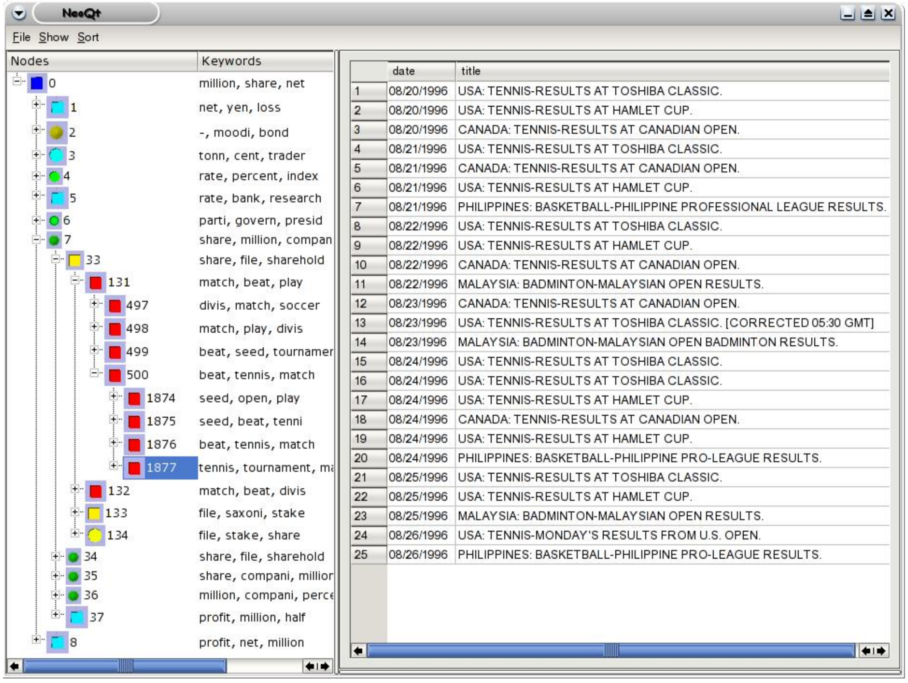
For the first part of the experiment we have created an artificial bag of words dataset with 10,000 items and 20 clusters as described in Section 3.5.2. We trained a H^{2}SOM with a branching factor of $n_b = 8$ and a depth of $r = 6$ rings. The resulting hierarchical data structure of the trained H^{2}SOM has 8,521 nodes and is characterized by the following features:

- At a high hierarchical level there are 20 main clusters present which represent the 20 topic categories.

- The cluster sizes are not equally distributed, but reflect the different category probabilities as shown in Figure 3.25.

- Deeper in the hierarchies, sub clusters emerge which correspond to sub categories within the main topic.

[1] http://www.trolltech.com



Nodes	Keywords
0	million, share, net
1	net, yen, loss
2	-, moodi, bond
3	tonn, cent, trader
4	rate, percent, index
5	rate, bank, research
6	parti, govern, presid
7	share, million, compan
33	share, file, sharehold
131	match, beat, play
497	divis, match, soccer
498	match, play, divis
499	beat, seed, tournamer
500	beat, tennis, match
1874	seed, open, play
1875	seed, beat, tenni
1876	beat, tennis, match
1877	tennis, tournament, ma
132	match, beat, divis
133	file, saxoni, stake
134	file, stake, share
34	share, file, sharehold
35	share, compani, millior
36	million, compani, perce
37	profit, million, half
8	profit, net, million

	date	title
1	08/20/1996	USA: TENNIS-RESULTS AT TOSHIBA CLASSIC.
2	08/20/1996	USA: TENNIS-RESULTS AT HAMLET CUP.
3	08/20/1996	CANADA: TENNIS-RESULTS AT CANADIAN OPEN.
4	08/21/1996	USA: TENNIS-RESULTS AT TOSHIBA CLASSIC.
5	08/21/1996	CANADA: TENNIS-RESULTS AT CANADIAN OPEN.
6	08/21/1996	USA: TENNIS-RESULTS AT HAMLET CUP.
7	08/21/1996	PHILIPPINES: BASKETBALL-PHILIPPINE PROFESSIONAL LEAGUE RESULTS.
8	08/22/1996	USA: TENNIS-RESULTS AT TOSHIBA CLASSIC.
9	08/22/1996	USA: TENNIS-RESULTS AT HAMLET CUP.
10	08/22/1996	CANADA: TENNIS-RESULTS AT CANADIAN OPEN.
11	08/22/1996	MALAYSIA: BADMINTON-MALAYSIAN OPEN RESULTS.
12	08/23/1996	CANADA: TENNIS-RESULTS AT CANADIAN OPEN.
13	08/23/1996	USA: TENNIS-RESULTS AT TOSHIBA CLASSIC. [CORRECTED 05:30 GMT]
14	08/23/1996	MALAYSIA: BADMINTON-MALAYSIAN OPEN BADMINTON RESULTS.
15	08/24/1996	USA: TENNIS-RESULTS AT TOSHIBA CLASSIC.
16	08/24/1996	USA: TENNIS-RESULTS AT TOSHIBA CLASSIC.
17	08/24/1996	USA: TENNIS-RESULTS AT HAMLET CUP.
18	08/24/1996	CANADA: TENNIS-RESULTS AT CANADIAN OPEN.
19	08/24/1996	USA: TENNIS-RESULTS AT HAMLET CUP.
20	08/24/1996	PHILIPPINES: BASKETBALL-PHILIPPINE PRO-LEAGUE RESULTS.
21	08/25/1996	USA: TENNIS-RESULTS AT TOSHIBA CLASSIC.
22	08/25/1996	USA: TENNIS-RESULTS AT HAMLET CUP.
23	08/25/1996	MALAYSIA: BADMINTON-MALAYSIAN OPEN RESULTS.
24	08/26/1996	USA: TENNIS-MONDAY'S RESULTS FROM U.S. OPEN.
25	08/26/1996	PHILIPPINES: BASKETBALL-PHILIPPINE PRO-LEAGUE RESULTS.

Figure 6.3: The interface of the traditional tree browser used in the experiments. With mouse clicks on the "+" and "-" icons, a node can be expanded or collapsed, respectively. For each node, keywords provide cues about their contents. By selecting a node, a drill down within the database is performed and the node's content is displayed on the right hand side of the interface. The situation shown here is similar to that of Figure 6.2, where the user has moved the fish-eye focus towards the sports cluster.

In order to provide visual hints to the users, each topic category is identified by one of 20 unique symbols. Additionally, each node is consecutively labeled. A resulting H^{2}SOM view on this artificial data structure is shown in Figure 6.4.

The first four task assignments were designed to familiarize the users with the interfaces and were formulated in a tutorial-like manner. They were not evaluated in terms of points achieved or task completion time.

Location of Predefined Targets

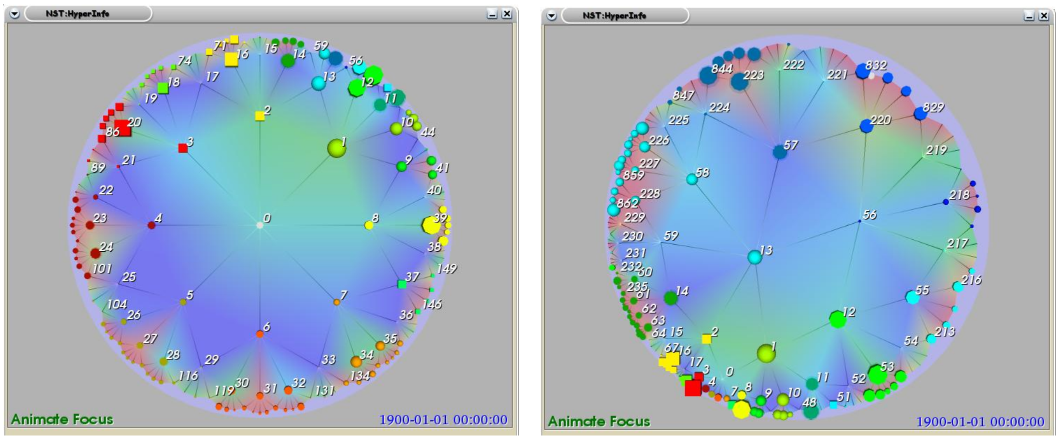
The evaluated section of the study started with a set of three tasks in which the users were asked to locate several predefined targets within the hierarchical data structure. The tasks as such were:

T1. *Locate the nodes along the path 0 - 8 - 39 - 157 and find all child nodes of node number 595. What are the numbers of the nodes below 595?* (1 point)

T2. *Which is the parent node of node number 595?* (1 point)

T3. *Locate the nodes along the path 0 - 3 - 20 - 84 - 325 and find the node with number 1222. Which color and shape does node 1222 have?* (1 point)

For each correct answer the users obtained one point, for a wrong answer they were penalized by a subtraction of one point from their score. By measuring the task completion time for this kind of assignments we can quantitatively assess how effectively users are able



(a) Centered H^{2}SOM view on artificial data.

(b) H^{2}SOM zoomed into upper right region.

Figure 6.4: Two screenshots showing a H^{2}SOM visualization of the artificial dataset. In **(a)** the overall structure with several large clusters can be seen. The upper right region of the map contains those categories which appear with lower probability. **(b)** shows a zoomed view of the H^{2}SOM into the corresponding region of the map and reveals further details about the less frequent categories.

to utilize the given graphical interfaces to navigate within the hierarchic data structure and to locate specific target regions. Typical situations for both types of interfaces are shown in Figure 6.5 below.

Exploring Hierarchies

In real world applications of exploratory data analysis, predefined targets within the data landscape are not necessarily known in advance. Therefore, the next set of task assignments required the users to interactively explore the data hierarchy in order to obtain the desired answers:

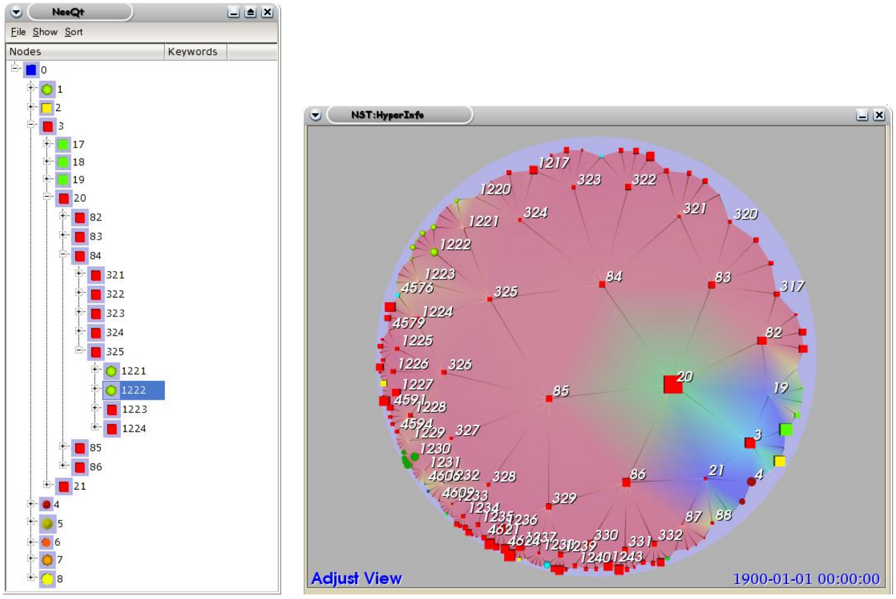
T4. *How many different types of nodes (in terms of color and shape) can you find on the map / in the tree?* (4 points)

T5. *Explore the group of the dark red circles below node number 4. Look for a group of yellow circles within that group. What are the node numbers of the yellow circles?* (4 points)

Especially question five required an exhaustive search within the data structure, since the sought-after sub-cluster was buried deep within the hierarchy of the data set. Again, the users could obtain or loose points for a correct or wrong answer, respectively. Additionally, the task completion time was measured.

Finding Answers in Newswire Articles

For the last set of tasks we constructed a more realistic scenario, in which users had to find answers to specific questions related to news messages from the Reuters newswire agency. To this end, we used a H^{2}SOM trained with 9241 news messages from the Reuters Corpus Volume 1, covering 10 different topic categories. For details on the selection, see Sauren (2005). Similar to the hierarchy for the 20 clusters dataset, we trained a H^{2}SOM with a branching factor of $n_b = 8$ and a depth of $r = 6$ rings, totaling in a hierarchy with 8,521 nodes. The users had to answer the following questions:



(a) Traditional tree browser (b) H^{2}SOM browser

Figure 6.5: Two screenshots showing the situation for both user interfaces when users were solving the 3rd question by locating node number 1222 in the hierarchy tree.

T6. *Search for messages related to food products such as "wheat" or "sugar". Answer the following question: "By what percentage did the Chinese sugar exports increase from January to July?"* (4 points)

T7. *Locate the "sports cluster" in the messages. Find news items related to tennis results. For which tennis tournaments can you find results?* (4 points)

Again, we measured the task completion time it took the users to obtain an answer to the questions, and the correctness of these answers by assigning points to them.

6.2.4 Evaluation Results

Differences for Hyperbolic Navigation Methodologies

User Score. First, we look at the impact the two different H^{2}SOM navigation methodologies had on the achieved user score. In Figure 6.6 the average scores for group I ("driving" of fovea), and group II ("dragging" of focus) are shown.

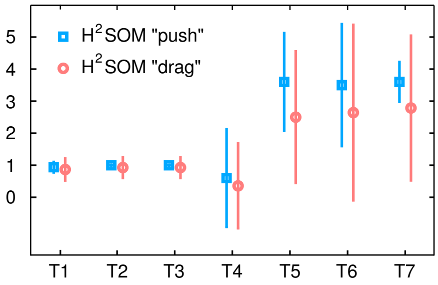


Figure 6.6: The score, i.e., the number of points groups I and II achieved with the two different navigation methodologies for the H^{2}SOM . The blue squares denote the "driving" variant, and the red circles the "dragging" variant of the fish-eye adjustment (cf.Section 5.2.2). The error bars indicate the standard deviation in the data.

For the simple tasks T1 – T3, the achieved results are almost identical. For the more complex tasks T4 – T7, the group using the "driving" variant of the fish-eye adjustment achieved a slightly higher average score. In order to test for any significant differences in the data, we computed the overall score and task completion time for the set of simple and complex tasks T1 – T3 and T4 – T7, respectively. Additionally, we also compared the overall performance by aggregating scores and times over all task assignments. We then applied a two-sided Welch's t-test[1] on these performance indicators.

	"drive" $\mu \pm \sigma^2$	"drag" $\mu \pm \sigma^2$	t	p	95% interval
Score T1 – T3	2.940 ± 0.128	2.725 ± 0.765	-0.249	0.3376	not significant
Score T4 – T7	11.30 ± 1.269	8.286 ± 5.037	2.0649	0.0563	not significant
Total score	14.24 ± 1.258	11.01 ± 5.558	2.0215	0.0616	not significant

Table 6.1: Differences in scores between the two H^{2}SOM groups I and II. The table shows mean and standard deviation for the aggregated scores of the simple and complex tasks T1 – T3 and T4 – T7, respectively. Additionally, the t- and p-values for the two sided Welch's t-test together with the 95% significance interval are given.

As can be seen in Table 6.1, the p-value for the difference in scores for the complex tasks is quite close to 0.05, such that the "driving" operation might indeed be more efficient in terms of the score. However, the data does not justify to call these differences significant. In other words, the H^{2}SOM navigation methodology has no significant impact on the achieved user score.

Task Completion Time. Second, we test whether the two different H^{2}SOM navigation schemes had an impact on task completion time. Figure 6.7 shows the results for group I and II.

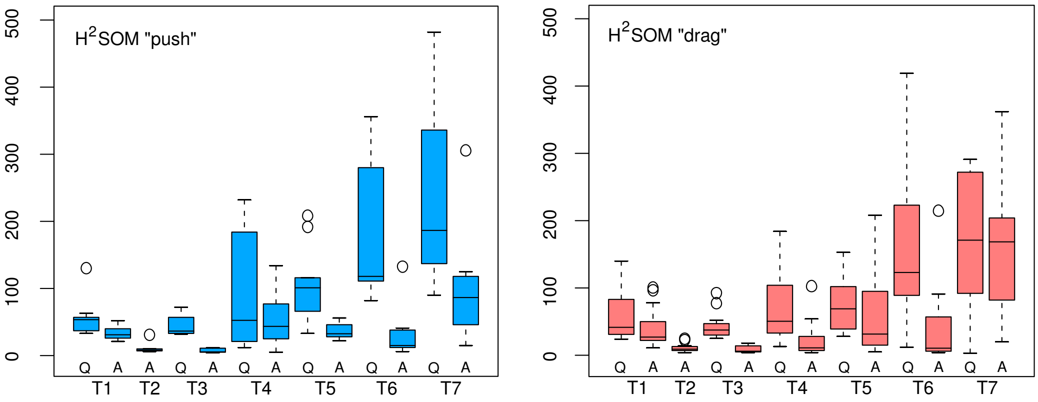


Figure 6.7: Completion times for tasks T1 – T7. The boxplots represent the distribution of seconds the users spend on each page. Q and A denote the question and answer page of the corresponding task. The box edges denote the inter-quartile range, the whiskers 1.5 times the inter-quartile range from upper or lower quartile, and the individual circles mark outliers.

The boxplots represent the distribution of seconds the users spend on each page of the questionnaire. Except for task T2, each assignment was covered by two pages: One page contained a task description and the question itself, and a second page contained the answer form. These two types are abbreviated with Q and A in Figure 6.7, respectively. The users were asked to press the *Next* button only when they found the answer to a question. Therefore,

[1] The Welch's t-test produces slightly smaller t-values as the traditional Student's t-test, but does not assume equal variances and is considered to be the safer one.

we expected to measure the task completion time only by the time the users spend on the question pages Q, since the submission of the answer pages A should have taken a constant small amount of time. But as Figures 6.7 and 6.9 show, especially for the more complex tasks, users decided to advance to the answer form in order to get an idea of the possible solution, before they actually knew the answer. Consequently, we take both types of pages into account when computing the task completion time for each question.

	"drive" $\mu \pm \sigma^2$	"drag" $\mu \pm \sigma^2$	t	p	95% interval
Time T1 – T3	151.9 ± 41.2	162.0 ± 64.9	-0.4461	0.6599	not significant
Time T4 – T7	831.1 ± 320.2	801.8 ± 292.5	0.2184	0.8296	not significant
Total time	983.0 ± 333.3	963.8 ± 306.0	0.1374	0.8922	not significant

Table 6.2: Differences in times between the two H^{2}SOM groups I and II. The table shows mean and standard deviation for the aggregated times for the simple and complex tasks T1 – T3 and T4 – T7, respectively. Additionally, the t- and p-values for the two sided Welch's t-test together with the 95% significance interval are given.

From the boxplots there is no clear distinction between the two groups visible. On some tasks, group I seems to achieve faster completion times, on other tasks group II solved the assignments in shorter time. Again, we used a two-sided Welch's t-test to test for significant differences. As can be seen in Table 6.2, the high p-values indicate, that any differences in time completion time between group I and II are pure random.

We can therefore conclude, that the two navigation methodologies "drive" and "drag" which utilize the mouse in different ways in order to move the focus on the hyperbolic plane of the H^{2}SOM , do not result in different user performances. Consequently, we regard the two H^{2}SOM groups I and II as a single group when comparing their performances to group III which used the traditional tree browser.

Differences between Hyperbolic and Tree Navigation

Since the two H^{2}SOM groups I and II showed no significant performance difference in score or task completion time, all samples from these two groups are merged into one H^{2}SOM group for the comparison to the tree browser group III.

User Score. The averaged scores for the H^{2}SOM group I+II and the traditional tree browser group III are shown in Figure 6.8.

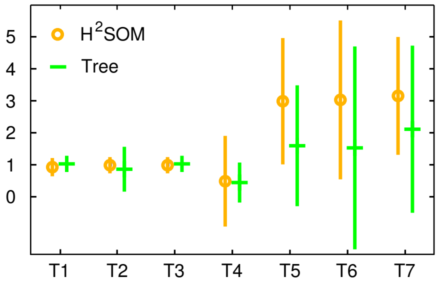


Figure 6.8: Number of points the H^{2}SOM groups I+II and the traditional tree browser group III achieved. The error bars indicate the standard deviation in the data.

For the simple assignments T1 – T3, both groups achieved nearly perfect results with little deviation. For the complex tasks T4 – T7, however, the H^{2}SOM group performed consistently better on the more difficult questions. Similar to the previous section we applied a two-sided Welch's t-test in order to test for significance.

	H^{2}SOM $\mu \pm \sigma^2$	Tree $\mu \pm \sigma^2$	t	p	95% interval
Score T1 – T3	2.815 ± 0.600	2.833 ± 0.552	-0.09	0.929	not significant
Score T4 – T7	9.542 ± 4.205	5.567 ± 3.515	2.8901	0.0078	1.145, 6.805
Total score	12.36 ± 4.606	8.400 ± 3.694	2.6901	0.0122	0.936, 6.977

Table 6.3: Differences in scores between H^{2}SOM and tree group. The table shows mean and standard deviation for the aggregated scores of the simple and complex tasks T1 – T3 and T4 – T7, respectively. Additionally, the t- and p-values for the two sided Welch's t-test together with the 95% significance interval are given.

The results on the Welch's t-test in Table 6.3 shows, that any differences in user scores for the simple tasks are indeed pure random. The p-value for the test on the complex tasks however, is highly significant. The corresponding 95% confidence level interval is given by $[1.145, 6.805]$, i.e. with a probability of 95% the H^{2}SOM group achieved between 1.145 and 6.805 points more for the four difficult questions.

Task Completion Time. For the task completion time, the boxplots in Figure 6.9 show the distribution of times the H^{2}SOM and the tree group spent on each page of the questionnaire.

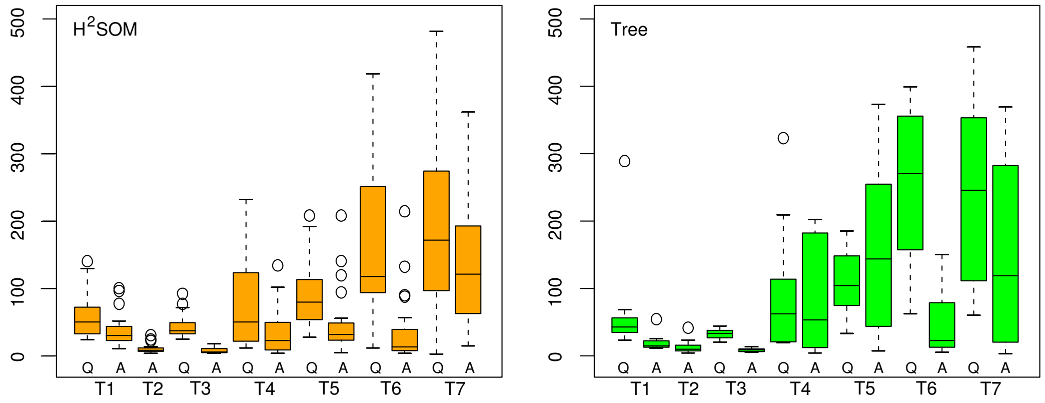


Figure 6.9: Task completion times for the H^{2}SOM group (left), and the tree browser group (right). The boxplots represent the distribution of seconds the users spend on the question (Q) and answer pages (A) of the questionnaire. Note, that the tree browser group III spend more time on the answer pages of the complex tasks A4 – A7.

The plots show that the first three tasks T1 – T3 are solved very fast by both user groups. The complex tasks on the other hand are obviously harder to solve, since both groups needed more time on these pages.

	H^{2}SOM $\mu \pm \sigma^2$	Tree $\mu \pm \sigma^2$	t	p	95% interval
Time T1 – T3	157.8 ± 56.5	136.8 ± 75.3	0.8228	0.4219	not significant
Time T4 – T7	814.0 ± 305.2	1245 ± 482	-2.7155	0.0157	-768.3, -93.3
Total time	917.8 ± 318.8	1382 ± 481	-2.5708	0.02071	-748.1, -71.4

Table 6.4: Differences in times between H^{2}SOM and tree group. The table shows mean and standard deviation for the aggregated times for the simple and complex tasks T1 – T3 and T4 – T7, respectively. Additionally, the t- and p-values for the two sided Welch's t-test together with the 95% significance interval are given.

Table 6.4 shows the results from the two-sided Welch's t-test. The p-value for the simple tasks T1 – T3 indicates, that we have to retain the null hypothesis, i.e. any differences in task completion time between the H^{2}SOM and the tree group are only by chance.

For the tasks T4 – T7 the p-value is distinctively below 0.05, therefore we may conclude, that on the complex assignments, the H^{2}SOM group performs significantly faster than the tree group. The 95% interval is within $[-768.3, -93.3]$, i.e. with a probability of 95% the H^{2}SOM group needed between 1.6 and 12.8 minutes less on the more difficult tasks.

6.2.5 Discussion

Fish-eye Adjustment: "Drive" vs. "Drag" Navigation

From Figure 6.6 one might conclude that the "drive" operation is slightly more efficient, but the data does not justify to call these differences significant. All other differences between the two H^{2}SOM groups are pure random. We can therefore conclude that the two H^{2}SOM groups I and II which used different methodologies to adjust the fish-eye fovea of the hyperbolic map did not show any significant differences either in accuracy or in task completion time.

H^{2}SOM vs. Classical Tree Browser

Previous user studies on hyperbolic focus & context visualization systems resulted in contradictorily results: Lamping et al. (1995) could not find any significant differences between their hyperbolic tree browser and a conventional 2D scrolling tree browser. Czerwinski and Larson (1997) compared Microsoft's Explorer with the hyperbolic tree browser and could also not detect any significant differences in user performances. On the other hand, the experiments by Mullet et al. (1997) and Pirolli et al. (2003) did find that subjects using the hyperbolic tree browser gained substantially superior results.

The results from our user study are consistent with the previous studies when we separate the browsing tasks into two classes:

1. The first class consisted of simple browsing tasks, where the users had to *navigate to a predefined target* within a well structured hierarchical data set. For these kinds of problems the users knew in advance where they had to look for a specific solution, and the different browsing methodologies just represented different means to reach the target.

2. In the second class, the problem assignments were more complex. In order to obtain a valid solution the users had to *explore* the hierarchical data structure and actively search for an answer to the given question. By navigating through the unknown territory, the users build up their own cognitive map of the dataset which finally enabled them to find a solution.

When considering the first class of simple tasks, our user study shows no significant differences between the H^{2}SOM and a classical tree browser whatsoever, and is therefore in line with the results from Lamping et al. (1995) and Czerwinski and Larson (1997).

For the second class of more complex tasks, our results show that the H^{2}SOM group achieved consistently higher scores in less time when compared to a classical tree browser: On average, the tree browser and the H^{2}SOM group achieved 5.567 points in 1245 seconds and 9.542 points in 814 seconds, respectively. That is, the H^{2}SOM group achieved a 71.4% higher score and needed 65.4% of the time the tree browser group did. A statistical analysis establishes these findings as highly significant ($p < 0.01$) and significant ($p < 0.05$) for the scores and the completion times, respectively. These results also mirror the discoveries from Mullet et al. (1997) and Pirolli et al. (2003).

6.3 Evaluation by aid infodienst Experts

Our second user study is targeting the question how useful the generated semantic maps are in real world applications. To answer the question we worked together with a team of food & health experts from the "aid infodienst" who evaluated the H^{2}SOM's performances with respect to an end user recommendation system. The general setting of the application is laid out in Section 5.3.3.

6.3.1 Methodology

Participants

Five professional food & health experts free-lancing for the "aid infodienst" participated in the study. All of them were accustomed to computer-use and were familiar with the forum system and its administration through a web-based backend.

Experimental Setup

The forum backend allows the experts to access all user questions by means of a standard webbrowser. Through a set of webforms these questions can be categorized and answered by the experts. We added an additional form to the backend which connects to the text mining engine web application interface (cf. Section 5.1.4). Each new question can now be submitted to the H^{2}SOM framework and a set of similar questions which have been asked in the past are returned. The experts were asked to evaluate the quality of the returned questions, i.e., they should judge whether the retrieved questions from the past and their corresponding answers would have helped the questioner of the new question. To this end they were asked to label each suggestion made by the text mining engine with "helpful", "neutral" and "not helpful".

As described in Section 5.3.3 we have trained two semantic maps: The first was trained with past questions only, i.e. the training data contained only user questions. The second map was trained with a combination of user question and corresponding expert answer. During the experiment the experts did not know whether the retrieval set was obtained from the first or the second map.

Procedure

The evaluation was carried out by the experts as they were using the forum backend in their standard workflow. During three months the experts answered some 203 user questions and gave 1195 ratings for the suggestions made by the H^{2}SOM system. That is, in average the H^{2}SOM found 5.9 similar questions from the past to a new user question.

6.3.2 Evaluation Results

Table 6.5 shows the numerical evaluation results from the three month evaluation period. The data is split into two parts for the two differently trained H^{2}SOMs. On average, only expert A rated the first map higher than the second one. A two-sided Welch's t-test shows that only the ratings of experts C and E differ significantly for the two semantic maps. The different ratings of experts A, B and D for the two differently trained maps are from a statistical point of view not significant[1].

	training on user questions only (map I)					training on questions plus answers (map II)				
expert	A	B	C	D	E	A	B	C	D	E
ratings	88	139	240	42	24	47	168	64	341	42
positive	19.3%	7.9%	6.3%	11.9%	4.2%	12.8%	14.9%	15.6%	17.6%	28.6%
neutral	23.9%	2.2%	14.6%	50.0%	8.3%	25.5%	0.0%	31.3%	42.2%	11.9%
negative	56.8%	89.9%	79.2%	38.1%	87.5%	61.7%	85.1%	53.1%	40.2%	59.5%
μ	**-0.38**	-0.82	-0.73	-0.26	-0.83	-0.49	**-0.70**	**-0.38**	**-0.23**	**-0.31**
σ	±0.79	±0.55	±0.57	±0.66	±0.47	±0.71	±0.71	±0.74	±0.73	±0.89

Table 6.5: Distribution of expert ratings for the two differently trained semantic maps. μ and σ denote mean and standard deviation of the ratings if counted as +1 for "helpful", 0 for "neutral" and -1 for "not helpful".

Table 6.6 shows the aggregated results over all experts. By looking at the total data, we see that on average the experts found 9.2% and 75.4% of all recommendations made by the system on the base of map I to be helpful and not helpful for the questioner, respectively. The data on map II shows an increase in helpful, and a decrease in not-helpful results: 17.1% vs. 55.6%, respectively. A two-sided Welch's t-test yielded a very low $p = 1.25\ 10^{-11}$ with a 99% confidence interval of [-0.38, -0.17].

	map I	map II
positive	9.2%	17.1%
neutral	15.4%	27.3%
negative	75.4%	55.6%
μ	-0.662	**-0.385**
σ	±0.638	±0.760

Table 6.6: Average ratings aggregated over all experts.

6.3.3 Discussion

The results show a significant better rating for the H^{2}SOM when it was trained not only with the original user questions but additionally with their corresponding expert answers. This result is rather interesting. From a machine learning viewpoint we initially expected a different outcome: The language used in the questions is typically not as rich as that in the more elaborate experts answers. During the recommendation process, a new question - for which there is no expert answer readily available - is mapped onto the H^{2}SOM trained with data from the past. Therefore, in case for map I which was trained with question data only, we map a new question onto a pure "questions" map. In case of map II which was trained with a combination of questions and answers we compare slightly different text types when mapping a question only onto the map trained with questions and their answers. We therefore initially expected, that map I would lead to a better approximation of the "question-space" and therefore to a better quality with respect to the ability to match new questions to previously asked ones. The expert evaluation shows the opposite: Obviously the internal representation of the H^{2}SOM obtained from question/answer pairs is able to benefit from the experts knowledge, such that if the H^{2}SOM is deployed as a recommendation system, the quality of its suggestions is improved.

[1] $p > 0.05$

The rate of 55.6% non-helpful answers found by the machine seems to be rather disappointing at first glance. That was also the notion of the experts who expected a better rate from the machine learning approach. However, 17.1% of the suggested results where rated as “helpful” by the experts. Since the system retrieved in average 5.9 similar questions to each new question, this is almost exactly a ratio of one helpful answer for each newly asked user question.

6.4 Summary

In this chapter we have presented two user studies addressing two questions: First, how effective is the hyperbolic focus & context visualization scheme with respect to navigation tasks in large data structures? And second, is the H^2SOM able to retrieve knowledge from unstructured text data which is comparable to expert knowledge?

In the first study thirtysix students were asked to either use the H^2SOM based navigation system or a conventional tree browser to solve navigational tasks. For simple browsing problems, i.e. when the users had to navigate to a predefined target known in advance, the results show no significant difference with respect to completion time and user score between both methodologies. However, for more complex tasks where the users had to explore an unknown data territory and actively search for information, the H^2SOM group achieved significantly higher scores in significantly less time than the tree browser control group.

In the second study five food & health experts evaluated the H^2SOM framework in a recommendation system setting. The system was trained with 15,000 user questions from the past. In the context of the application the system should classify new user questions and retrieve similar ones from the past. On average the system returned 5.9 hits for newly asked questions, and from those suggestions the experts rated 1.0 to be helpful, 1.6 to be neutral and 3.3 not to be helpful for the questioner. At first, the experts were slightly disappointed by this achievement of the system. On second thought however, they acknowledged that the system in average found one helpful answer for each new question. Regarding that the questioner is able to get that helpful answer immediately, the H^2SOM based recommendation system can be considered to be a valuable service.

Chapter 7

Conclusion and Outlook

7.1 Summary

In this thesis we have presented a novel approach for the semantic visualization of large bodies of document sets. The suggested new methodology is aiming at the alleviation of information overload which is caused be the ever increasing amount of data available to us. We have proposed a technical framework build upon the integration of four different perspectives which have been highlighted by previous work. These are *(i)* techniques from *information retrieval* for the discovery of data items related to specific information needs, *(ii) machine learning* approaches concerned with algorithms which are able to learn from data, *(iii) information visualization* targeting the transformation of data into visual form permitting the viewer to look and browse through large amounts of information and *(iv)* the incorporation of *semantic networks* to imprint additional layers of meaningful semantics onto the data.

In a first step we have investigated the possibility to exchange the geometrical substrate which is used for the construction of a regular lattice of formal neurons in the self-organizing map (SOM). While the standard SOM uses the flat Euclidean space, we employ the non-Euclidean hyperbolic plane as first suggest by Ritter (1999). We have demonstrated that the peculiar geometric properties of the hyperbolic plane $I\!H^2$ offer two distinctive advantages: First, the neighborhood around any point in that space grows asymptotically exponentially, allowing the formal neurons of the hyperbolic self-organizing map (HSOM) to “feel” more freedom for adaptation. Second, the projection of $I\!H^2$ onto the two-dimensional Poincaré Disk allows for a natural *focus & context* navigation scheme within very large map spaces.

Based on two intuitively graspable datasets and the Reuters-21578 benchmark we have demonstrated a slightly superior performance of the HSOM as compared to the standard SOM with respect to quantization error, neighborhood preservation and classification accuracy. However, the HSOM approach suffers from one distinctive drawback: For high-dimensional data, the user is confronted with mainly empty space in the center of the map. From a statistical point of view, this reflects the data distribution rather well, since the vast majority of data items in high-dimensional spaces actually resides in the “periphery”. However, from a visualization perspective this property is not beneficial, since the user is forced to exhaustively navigate within the map in order to examine the whole dataset.

The potential of $I\!H^2$ to naturally embed arbitrary large hierarchical structures opens up the possibility to construct a hierarchically growing variant of the HSOM, the H^{2}SOM.

The peculiar, intrinsically "uniformly hierarchical" structure of the hyperbolic grid offers an intriguing possibility to significantly accelerate the most time-consuming step of the self-organizing approach: By approximating the global search for the winner neuron by a *fast tree search*, we obtain a search path with a complexity of $\mathcal{O}(logN)$ instead of $\mathcal{O}(N)$ for a global search. As a result, not only the training time is drastically reduced from several hours to several minutes, but also the generalization capabilities of the network seem to improve: In the context of text categorization tasks our experiments reveal that for maps consisting of the same number of neurons the H^{2}SOM generally outperforms the HSOM in terms of neighborhood preservation and classification accuracy. Only in terms of quantization error the H^{2}SOM performs worse. This is explained by the different utilization of the hyperbolic lattice structure: While the HSOM uses all nodes to approximate the data, the H^{2}SOM uses the first "inner" levels of the grid for its hierarchical structuring, such that less nodes are available for representing the data as compared to the "flat" node utilization of the HSOM.

The hierarchical organization of the H^{2}SOM provides us also with the opportunity to extend the standard model mostly used for the representation of unstructured text data. By borrowing the concept of pyramidal feature spaces from image processing we expand the classical *bag-of-words* to a hierarchically organized *pyramid-of-words*. We have integrated the semantic lexicon offered by WordNet to achieve a semantically guided representation of documents: On a coarse semantic level documents can be represented with fewer dimensions by more generalizing terms. This allows documents sharing the same cognitive concepts to be similar to each other without the need to necessarily share the same words. In order to deal with linguistic polysemy we have introduced a simple neural model which performs a probabilistic word sense disambiguation. The results on the Reuters-21578 corpus indicate that overall classification performance is slightly increased by the pyramid-of-words. For low frequently appearing topics the precision at equal recall levels is significantly increased between 30% and 130% as compared to the standard bag-of-words representation. The comparison to other classifier schemes has shown that the H^{2}SOM trained with the pyramid-of-words achieves very competitive results. In terms of overall precision-recall accuracy, the H^{2}SOM is significantly better than a *naives Bayes* classifier and almost on par with a *k-nearest neighbor* approach. A *support vector machine* (SVM) still performs better, however at the cost of larger computational complexity and not offering a visualization framework as the H^{2}SOM does.

Based upon the results obtained with the H^{2}SOM we have presented the design of a technical system realizing an interactive text mining and visualization instrument. The operation of the system was demonstrated on three real world cases. First, more than 20,000 messages from an Internet forum have been analyzed. We have shown that the hierarchically self-organization process and the tag cloud augmented visualization allows the user to quickly obtain an overview of the message structure. We have presented a semantic extension to the classical information retrieval process offering the opportunity to retrieve results which are relevant to the semantic concepts of a user query, but which do not necessarily need to contain any of the used query terms. Second, for the Reuters-21578 newswire collection we have given an example how the time dependent node activation potentials reflect and visualize the evolution of topics within document streams, enabling the user to detect up-and-coming developments. Third, we have shown the employment of a web application interface within the setting of a food & health recommendation system.

To corroborate the findings from our investigations we designed and carried out two user studies. In the first study 36 students were asked to use either a conventional tree browser or the hyperbolic framework in order to answer the question how effectively the H^{2}SOM focus & context system is with respect to navigation tasks in large data structures. In the second study five experts working for an Internet forum related to food and health were asked to evaluate our framework within a recommendation system setting.

The results from the first study have shown, that for simple browsing tasks both navigation frameworks, i.e. the conventional and the hyperbolic, do not expose any significant difference with respect to completion time and user score. However, for more complex tasks where the users had to explore an unknown data territory and actively search for information, the H^{2}SOM group achieved significantly higher scores in significantly less time than the tree browser control group.

During the second study, the experts initially were disappointed by the achievements of the text mining system: On average the system returned 5.9 hits for newly asked questions, and from those suggestions the experts rated 1.0 to be helpful, 1.6 to be neutral and 3.3 not to be helpful for the questioner. On second thought, the experts acknowledged that the system found one helpful answer for each new user question. Regarding that the questioner is able to get that helpful answer immediately, the H^{2}SOM based recommendation system was considered to be a valuable service.

Our a novel approach for exploring structure in large data sets therefore offers the following advantages:

- The computational complexity of the H^{2}SOM allows to build large semantic information maps with the order of $\mathcal{O}(log\ N)$ with N being the number of nodes in the map. In comparison to the standard SOM this allows the mapping of large data sets within several minutes as opposed to several hours.

- With the incorporation of the lexical database WordNet, we achieve a semantically guided representation of documents allowing for a matching of cognitive concepts instead of words only.

- A node labelling and ranking scheme offers the possibility to employ the H^{2}SOM as a text categorization tool. The system can therefore be used twofold: First as an unsupervised clustering tool. Second, if there is labelled data available, as a classification tool.

- The generated mappings can be displayed by an interactive visualization engine offering a natural focus & context framework. The maps are embedded in 3D and augmented by a self-organizing tag cloud providing semantic information scents to the user. Additionally, a time dependent node activation might be visualized offering the user a way of tracking the evolution of topics in document streams.

- Classical information retrieval searches can be performed by using the underlaying MySQL fulltext indexing engine. By visualizing the results as accumulated hits on the hierarchical node structure of the H^{2}SOM, additional semantic hints are added to the retrieval set.

7.2 Perspectives

The results we have reported in this theses induce a series of possible future research directions:

The construction of the hierarchically growing hyperbolic self-organizing map was largely directed by the lattice structure created by the regular tessellation with equilateral triangles. This forces a fixed number of child nodes for each hierarchical level. Due to the self-organizing character this poses no hard limitation, since the network is able to fill gaps by “interpolating nodes”, but for future research a freely growing topology could provide even more freedom to the self-organizing process.

Another open question is motivated by the successful employment of the hierarchical feature representation of text documents. We have used an external knowledge repository, i.e. WordNet, in order to construct the semantically guided pyramid-of-words. On the basis of very large document databases we expect that also the more general terms from the upper pyramidal levels should be found within the text data. The question is, how could a semantical lexicon such as WordNet be *learned* from the data? The exploration of this direction is also highly relevant for non textual domains. For example, in the domain of bioinformatics: Due to recently available technologies, we are able to produce large amounts of genomic sequences. It would be fascinating to investigate the question whether a hierarchically semantic structuring could also be learned from that data.

A further question was touched when we analyzed WordNet’s hypernym statistics. Very interestingly, the average hypernym depth of nouns is significantly larger than that of verbs. This poses the question, whether this difference is due to the human expert input, or if there is a fundamental cognitive disparity between things and actions. The data in WordNet suggests that we cognitively organize things on a much finer granularity than actions. This finding might be relevant when considering cognitive systems, i.e. in robotics.

Bibliography

Agirre, E. and Rigau, G. (1996). Word sense disambiguation using conceptual density. In *International Conference On Computational Linguistics*, pages 16–22. Copenhagen, Denmark.

Anderson, J. (2001). *Hyperbolic Geometry*. 2nd printing. Springer-Verlag.

Aronson, A., Bodenreider, O., Chang, H., Humphrey, S., Mork, J., Nelson, S., Rindflesh, T., and Wilbur, W. (2000). The NLM indexing initiative. In *Proceedings of the Annual AMIA Symposium*, pages 17–21.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI*. Acapulco, Mexico.

Bauer, H.-U. and Pawelzik, K. R. (1992). Quantifying the neighborhood preservation of self-organizing feature maps. In *IEEE Transactions on Neural Networks*, 3(4):570–579.

Bederson, B., Shneiderman, B., and Wattenberg, M. (2002). Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. In *ACM Transactions on Graphics*, 21(4):833–854.

Belman, R. E. (1957). *Dynamic Programming*. Princeton University Press, New Jersey.

Beltrami, E. (1868). Saggio di interpretazione della geometria non-euclidea. In *Giornale di Matematiche*, 6:284–312.

Berners-Lee, T., Hendler, J., and O.Lassila (2001). The semantic web. In *Scientific American*, pages 34–43.

Berthold, M. and Hand, J., editors (1999). *Intelligent Data Analysis*. Springer, Berlin.

Bethesda (1999). *MeSH. Medical Subject Headings*. National Library of Medicine.

Bezdek, J. and Pal, N. (1995). An index of topological preservation for feature extraction. In *Pattern Recognition*, 28(3):381–391.

Bishop, C. M., Svenson, M., and Williams, C. K. I. (1998). GTM: The generative topographic mapping. In *Neural Computation*, 10(1):215–234.

Blakemore, C. and Cooper, G. (1970). Development of the brain depends on the visual environment. In *Nature*, 228:477–478.

Bloehdorn, S. and Hotho, A. (2004). Boosting for text classification with semantic features. In *Proceedings of the MSW 2004 workshop at the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 70–87.

Bollacker, K. D., Lawrence, S., and Giles, C. L. (2000). Discovering relevant scientific literature on the web. In *IEEE Intelligent Systems*, 15(2):42–47.

Bolyai, F. (1832). *Tentamen juventutem studiosam in elementa matheseos purae elementis ac sublimioris, methodo intuitiva, evidentiaque huic propria, introducendi*, chapter Scientia absoluta spatii. Maros Vasarhely.

Borst, A. and Theunissen, F. E. (1999). Information theory and neural coding. In *Nature neuroscience*, 2(11):947–957.

Bradburn, D. (1989). Reducing transmission error effects using a self-organizing network. In *Proc. of the IJCNN89*, volume II, pages 531–538. San Diego, CA.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Burt, P. and Adelson, E. (1983). The laplacian pyramid as a compact image code. In *IEEE Transactions on Communication*, 31(4):532–540.

Bush, V. (1945). As we may think. In *Atlantic Monthly*.

Cannon, J., Floyd, W., Kenyon, R., and Parry, W. (1997). *Flavours of Geometry*, chapter Hyperbolic Geometry. 31. MSRI Publication.

Card, S., MacKinlay, J., and Shneidermann, B. (1999). *Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco.

Carey, M., Kriwaczek, F., and Rüger, S. M. (2000). A visualization interface for document searching and browsing. Technical report, Department of Computing, Imperial College London.

Chavel, I. (1994). *Riemannian Geometry: A Modern Introduction*. Cambridge University Press.

Cox, T. F. and Cox, M. A. (1994). *Multidimensional Scaling*. Monographs on Statistics and Appied Probability. Chapman and Hall.

Coxeter, H. (1988). The trigonometry of Escher's woodcut "Circle Limit III". In *The Mathematical Intelligencer*, 18:42–46.

Coxeter, H. S. M. (1957). *Non Euclidean Geometry*. Univ. of Toronto Press, Toronto.

Coxeter, H. S. M. (1979). The non-Euclidean symmetry of Escher's picture "Circle Limit III". In *Leonardo*, 12:19–25.

Craven, M. and Kumlien, J. (1999). Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*, pages 77–86. Heidelberg.

Cugini, J., Laskowski, S., and Sebrechts, M. (1999). Design of 3-d visualization of search results: Evolution and evaluation. Technical report, National Institute of Standards and Technology.

Czerwinski, M. and Larson, K. (1997). The new web browsers: They're cool but are they useful? In *People and Computers XII: Proceedings of HCI'97*.

Dalva, M. and Katz, L. (1994). Rearrangements of synaptic connections in visual cortex revealed by laser photostimulation. In *Science*, 265:255–258.

Decker, R., Wagner, R., and Scholz, S. (2005). Environmental scanning in marketing planning. In *Marketing Intelligence and Planning*, 23:189–199.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A.

(1990). Indexing by latent semantic analysis. In *Journal of the American Society of Information Science*, 41(6):391–407.

do Carmo, M. (1976). *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

Dunham, D. (1988). *M.C. Escher: Art and Science*, chapter Creating Hyperbolic Escher Patterns, pages 241–248. Elsevier Science Publishers B.V.

Durbin, R. and Mitchison, G. (1990). A dimension reduction framework for understanding cortical maps. In *Nature*, 343:644–647.

Edmunds, A. and Morris, A. (2000). The problem of information overload in business organisations: a review of the literature. In *International Journal of Information Management*, 20(1):17–28.

Efron, M. (2007). Model-averaged latent semantic indexing. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR*, pages 755–756. Amsterdam, The Netherlands.

Endsley, M. and Hoffman, R. R. (2002). The Sacagawea principle. In *IEEE Intelligent Systems*, 17:80–85.

Erwin, E., Obermayer, K., and Schulten, K. (1992). Self-organizing maps: Ordering, convergence properties and energy functions. In *Biological Cybernetics*, 67:47–55.

Faloutsos, C. and Lin, K.-I. (1995). FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In M. J. Carey and D. A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174. San Jose, California.

Fellbaum, C., editor (2001). *WordNet - An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Freeman, R. T. and Yin, H. (2004). Adaptive topological tree structure for document organisation and visualisation. In *Neural Networks*, pages 1255–1271.

Fricke, R. and Klein, F. (1897). *Vorlesungen über die Theorie der automorphen Funktionen*, volume 1. Teubner, Leipzig. Reprinted by Johnson Reprint, New York, 1965.

Funk, M., Reid, C., and McGoogan, L. (1983). Indexing consistency in MEDLINE. In *Bull. Med. Libr. Assoc.*, pages 176–183.

Furnas, G. W. (1986). Generalized fisheye views. In *Proceedings of the ACM SIGCHI 86 Conference on Human Factors in Computing Systems*, pages 16–23.

Gaskett, C. and Cheng, G. (2003). Online learning of a motor map for humanoid robot reaching. In *Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*. Singapore.

Gonzalez, R. and Woods, R. (2008). *Digital Image Processing*. Prentice Hall.

Gonzalo, J., Verdejo, F., Chugur, I., and Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP*. Montral, Canada.

Goodhill, G., Finch, S., and Sejnowski, T. (1996). Optimizing cortical mappings. In *Advances in Neural Information Processing Systems (NIPS)*, volume 8, pages 330–336.

Goodhill, G. J. and Sejnowski, T. (1997). A unifying objective function for topographic mappings. In *Neural Computation*, 9:1291–1303.

Graepel, T. and Obermayer, K. (1999). A stochastic self-organizing map for proximity data. In *Neural Computation*, 11(1):139–155.

Grahl, M., Hotho, A., and Stumme, G. (2007). Conceptual clustering of social bookmarking sites. In *7th International Conference on Knowledge Management (I-KNOW '07)*, pages 356–364. Graz, Austria.

Guan, Z. (2007). An eye tracking study of the effect of target rank on web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems.*, pages 417–420. San Jose, California, USA.

Hand, D., Mannila, H., and Smyth, P. (2001). *Principles of Data Mining*. MIT press, Cambridge.

Haney, D. M., McKeough, L. T., Smith, B. M., and Broughton, A. (2000). Divisible tilings in the hyperbolic plane. In *New York Journal of Mathematics*, 6:237–283.

Hassan-Montero, Y. and Herrero-Solana, V. (2006). Improving tag-clouds as visual information retrieval interfaces. In *Proceedings of International Conference on Multidisciplinary Information Sciences and Technologies*. Merida, Spain.

Havre, S., Hetzler, E., Whitney, P., and Nowell, L. (2002). Themeriver: Visualizing thematic changes in large document collections. In *IEEE Transactions on Visualization and Computer Graphics*, 8(1).

Herman, I., Melancon, G., and Marshall, M. S. (2000). Graph visualization and navigation in information visualization: a survey. In *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43.

Herrero, J., Valencia, A., and Dopazo, J. (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. In *Bioinformatics*, 17(2):126–136.

Hetzler, B., Harris, W., Havre, S., and Whitney, P. (1998). Visualizing the full spectrum of document relationships. In *Proceedings of the Fifth International Society for Knowledge Organization (ISKO)*.

Hilbert, D. (1901). Über Flächen constanter Gaußscher Krümmung. In *Transactions of the American Mathematical Society*, 2:87–99.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. In *Machine Learning*, 42:177–196.

Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1996). Newsgroup exploration with websom method and browsing interface. Technical Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science.

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

Hotho, A., Staab, S., and Stumme, G. (2003a). Explaining text clustering results using semantic structures. In *Principles of Data Mining and Knowledge Discovery, PKDD*.

Hotho, A., Staab, S., and Stumme, G. (2003b). WordNet improves text document clustering. In *In Proceedings of the Semantic Web Workshop at SIGIR-2003, 26th Annual International ACM SIGIR Conference*. Toronto, Canada.

Hubel, D. and Wiesel, T. (1959). Receptive fields of single neurones in the cat's striate cortex. In *Journal of Physiology*, 148:574–591.

Humphrey, S. M. (1992). Indexing biomedical documents: From thesaural to knowledge-based retrieval systems. In *Artificial Intelligence in Medicine*, 4(5):343–371.

Hung, C. and Wermter, S. (2004). Neural network based document clustering using WordNet ontologies. In *International Journal of Hybrid Intelligent Systems*, 1(3):127–142.

Hung, C., Wermter, S., and Smith, P. (2004). Hybrid neural document clustering using guided self-organization and WordNet. In *IEEE Intelligent Systems*, 19(2):68–77.

Jankun-Kelly, T. J. and Ma, K.-L. (2001). Visualization exploration and encapsulation via a spreadsheet-like interface. In *IEEE Transactions on Visualization and Computer Graphics*, 7(3):275–287.

Jansen, H. (1909). Abbildung der hyperbolischen Geometrie auf ein zweischaliges Hyperboloid. In *Mitteilungen der mathematischen Gesellschaft Hamburg*, 4:409–440.

Jenssen, T.-K., Laegreid, A., Komorowski, J., and Hovig, E. (2001). A literature network of human genes for high-troughput analysis of gene expression. In *Nature Genetics*, 28:21–28.

Joachims, T. (1997). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Technical Report LS8-Report 23, Universität Dortmund.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 1398, pages 137–142. Chemnitz, DE.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, Edmonton, Canada.

Kaski, S. (1997). Data exploration using self-organizing maps. In *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series*, 82.

Kaski, S., Kangas, J., and Kohonen, T. (1998a). Bibliography of self-organizing map (SOM) papers: 1981-1997. In *Neural Computing Surveys*, 1:102–350.

Kaski, S. and Lagus, K. (1996). Comparing self-organizing maps. In *Proceedings of ICANN'96, International Conference on Artificial Neural Networks*, pages 809–814.

Kaski, S., Lagus, K., Honkela, T., and Kohonen, T. (1998b). WEBSOM – self-organizing maps of document collections. In *Neurocomputing*, 21:101–117.

Kehagias, A., Petridis, V., Kaburlasos, V., and Fragkou, P. (2001). A comparison of word- and sense-based text categorization using several classification algorithms. In *Journal of Intelligent Information Systems*, 21(3):227–247.

Kiang, M. and Kumar, A. (2001). An evaluation of self-organizing map networks as a robust alternative to factor analysis in data mining applications. In *Information Systems Research*, 12:177–194.

Killing, W. (1880). Die Rechnung in den Nicht-Euklidischen Raumformen. In *Journal der Reinen Angewandten Mathematik*, 89:265–287.

Kiviluoto, K. (1998). Comparing 2D and 3D self-organizing maps in financial data visualization. In *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems (IIZUKA'98)*, pages 68–71. Singapore.

Klein, F. (1871). Über die sogenannte Nicht-Euklidische Geometrie. In *Mathematische Annalen*, 4:573–625.

Klein, F. and Fricke, R. (1890). *Vorlesungen über die Theorie der elliptischen Modulfunktionen*. Teubner, Leipzig. Reprinted by Johnson Reprint, New York, 1965.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. In

Biological Cybernetics, 43:59–69.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer Series in Information Sciences. 3rd edition.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Paatero, V., and Saarela, A. (2000). Organization of a massive document collection. In *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585.

Kohonen, T., Kaski, S., and Lappalainen, H. (1997). Self-organized formation of various invariant-feature filters in the adaptive-subspace som. In *Neural Computation*, 9(6):1321–1344.

Koikkalainen, P. (1994). Progress with the tree-structured self-organizing map. In *11th European Conference on Artificial Intelligence (ECAI 1994)*, pages 211–215.

Koikkalainen, P. and Oja, E. (1990). Self-organizing hierarchical feature maps. In *Proc. of the IJCNN 1990*, volume II, pages 279–285.

Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. (1996). Self-organizing maps of document collections: A new approach to interactive exploration. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 238–243.

Lagus, K., Kaski, S., and Kohonen, T. (2004). Mining massive document collections by the websom method. In *Information Sciences*, 163:135–156.

Lamping, J. and Rao, R. (1994). Laying out and visualizing large trees using a hyperbolic space. In *ACM Symposium on User Interface Software and Technology*, pages 13–14.

Lamping, J., Rao, R., and Pirolli, P. (1995). A focus+content technique based on hyperbolic geometry for viewing large hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 401–408. ACM, Denver.

Lawrence, S. (2001). Access to scientific literature. In D. Butler, editor, *The Nature Yearbook of Science and Technology*. Macmillan, London.

Lawrence, S., Bollacker, K., and Giles, C. L. (1999). Indexing and retrieval of scientific literature. In *Eight International Conference on Information and Knowledge Management, CIKM 99*, pages 139–146. Kansas City.

Lewis, D. (1996). Dying for information? In *Reuters Business Information*.

Lewis, D. (2004). Reuters-21578 collection 1.0. Technical report, AT&T Labs Research. Http://www.daviddlewis.com/resources/testcollections/reuters21578/.

Liu, T., Chen, Z., Zhang, B., ying Ma, W., and Mu, G. (2004). Improving text classification using local latent semantic indexing. In *Fourth IEEE International Conference on Data Mining, ICDM*, pages 162–169.

Lobachevsky, N. (1829). On the principles of geometry. Kazan Messenger.

Lobachevsky, N. (1837). Géométrie imaginaire. In *Journal für die reine und angewandte Mathematik*, 17:295–320.

Lobachevsky, N. (1840). *Geometrische Untersuchungen zur Theorie der Parallellinien*. F. Fincke, Berlin.

Lodhi, H., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2001). Text classification using string kernels. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 563–569. MIT Press.

Luttrell, S. (1994). A bayesian analysis of self-organizing maps. In *Neural Computation*, 6:767–794.

Magnus, W. (1974). *Noneuclidean Tesselations and Their Groups*. Academic Press.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Marcotte, E. M., Xenarios, I., and Eisenberg, D. (2001). Mining literature for protein-protein interactions. In *Bioinformatics*, 17(4):359–363.

Martin, G. E. (1975). *The Foundations of Geometry and the Non-Euclidean Plane*. Springer-Verlag.

Merkel, D. and Rauber, A. (1998). CIA's view of the world and what neural networks learn from it. In *Proceedings of the Int'l Conference on Database and Expert Systems Applications (DEXA'98)*. Springer Verlag, Vienna.

Merkl, D. (1997). Exploration of text collections with hierarchical feature maps. In *Proceedings of the Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*. ACM Press, Philadelphia.

Merkl, D. (1998). Text classification with self-organizing maps: Some lessons learned. In *Neurocomputing*, 21.

Miller, N. E., Wong, P. C., Brewster, M., and Foote, H. (1998). Topic Islands - A wavelet-based text visualization system. In D. Ebert, H. Hagen, and H. Rushmeier, editors, *IEEE Visualization '98*, pages 189–196.

Misner, C. W., Wheeler, J. A., and Thorne, K. S. (1973). *Gravitation*. Freeman.

Morgan, F. (1993). *Riemannian Geometry: A Beginner's Guide*. Jones and Bartlett Publishers, Boston, London.

Mullet, K., Fry, C., and Schiano, D. (1997). On your marks, get set, browse! In *Proceedings of Human Factors in Computing Systems, CHI ´97*. Atlanta, USA.

Munzner, T. (1997). H3: Laying out large directed graphs in 3D hyperbolic space. In *Proceedings of the 1997 IEEE Symposium on Information Visualization, Phoenix, AZ*, pages 2–10.

Munzner, T. (1998). Exploring large graphs in 3D hyperbolic space. In *IEEE Computer Graphics and Applications*, 18(4):18–23.

Munzner, T. and Burchard, P. (1995). Visualizing the structure of the World Wide Web in 3D hyperbolic space. In *Proc. VRML*, pages 33–38. ACM Press.

Munzner, T., Guimbretiere, F., Tasiran, S., Zhang, L., and Zhou, Y. (2003). TreeJuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. In *SIGGRAPH 2003*, pages 453–462.

Obermayer, K., Ritter, H., and Schulten, K. (1990). A neural network model for the formation of topographic maps in the CNS: Development of receptive fields. In *IJCNN-90*, volume II, pages 423–429. San Diego.

Oja, M., Kaski, S., and Kohonen, T. (2003). Bibliography of self-organizing map (SOM) papers: 1998-2001 addendum. In *Neural Computing Surveys*, 3:1–156.

Ontrup, J., Nattkemper, T., Gerstung, O., and Ritter, H. (2003). A mesh term based distance measure for document retrieval and labeling assistance. In *Proceedings of the 25th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society (EMBS)*. Cancun, Mexiko.

Ontrup, J. and Ritter, H. (2001a). Hyperbolic self-organizing maps for semantic navigation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 1417–

1424.

Ontrup, J. and Ritter, H. (2001b). Text categorization and semantic browsing with self-organizing maps on non-Euclidean spaces. In *Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 338–349. Springer, LNAI 2168.

Ontrup, J. and Ritter, H. (2005a). *Clinical Knowledge Management: Opportunities and Challenges*, chapter Interactive Information Retrieval as a Step Towards Effective Knowledge Management in Healthcare, pages 52–71. Idea Group Publishing.

Ontrup, J. and Ritter, H. (2005b). A hierarchically growing hyperbolic self-organizing map for rapid structuring of large data sets. In *Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM 05)*. Paris, France.

Ontrup, J. and Ritter, H. (2006). Large-scale data exploration with the hierarchically growing hyperbolic SOM. In *Neural Networks*, 19(6):751–761.

Ontrup, J., Wersing, H., and Ritter, H. (2004). A computational feature binding model of human texture perception. In *Cognitive Processing*, 5(1).

Pakkanen, J. (2003). The Evolving Tree, a new kind of self-organizing neural network. In *Proceedings of the Workshop on Self-Organizing Maps '03*, pages 311–316. Kitakyushu, Japan.

Pakkanen, J., Iivarinen, J., and Oja, E. (2004). The Evolving Tree – a novel self-organizing network for data analysis. In *Neural Processing Letters*, 20(3):199–211.

Pal, N. and Eluri, V. (1998). Two efficient connectionist schemes for structure-preserving dimension reduction. In *IEEE TNN*, pages 1142–1154.

Penfield, W. and Boldrey, E. (1937). Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation. In *Brain*, 60:389–443.

Pirolli, P. and Card, S. (1999). Information foraging. In *Psychological Review*, pages 643–675.

Pirolli, P., Card, S., and Van der Wege, M. (2000). The effect of information scent on searching information: visualizations of large tree structures. In *Proceedings of the working conference on Advanced visual interfaces (AVI)*, pages 161–172. ACM Press, Palermo, Italy.

Pirolli, P., Card, S., and Van Der Wege, M. (2003). The effects of information scent on visual search in the hyperbolic tree browser. In *ACM Transactions on Computer-Human Interaction (TOCHI)*, 10(1):20–53.

Pirolli, P., Card, S. K., and Van Der Wege, M. M. (2001). Visual information foraging in a focus + context visualization. In *ACM Conference on Human Factors in Computing Systems, CHI Letters*, 3(1):506–513.

Playfair, J. (1861). *Elements of Geometry: Containing the First Six Books of Euclid, with a Supplement on the Circle and the Geometry of Solids to which are added Elements of Plane and Spherical Trigonometry*. W.E. Dean, New York.

Poincaré, H. (1881). Sur les applications de la géométrie non euclidienne à la théorie des formes quadratiques. In *Compte Rendu de l'association Francaise pour l'Avancement des Sciences, 10^{e} Session*, pages 132–138. Alger.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical recipies in C: the art of scientific computing*. Cambridge University Press, New York, 2nd edition.

Purcell, G., Rennels, G., and Shortliffe, E. (1997). Development and evaluation of a context-based document representation for searching the medical literature. In *International Journal of Digital Libraries*, 1:288–296.

Ramsay, A. and Richtmyer, R. (1995). *Introduction to Hyperbolic Geometry*. Springer-Verlag.

Raskutti, B., Ferra, H. L., and Kowalczyk, A. (2001). Second order features for maximising text classification performance. In *Proceedings of ECML-01, 12th European Conference on Machine Learning*, pages 419–430. Springer, Freiburg.

Rauber, A. and Merkl, D. (2001). Automatic labeling of self-organizing maps for information retrieval. In *Journal of Systems Research and Information Systems (JSRIS)*, 10(10):23–45.

Rauber, A., Merkl, D., and Dittenbach, M. (2002). The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. In *IEEE Transactions on Neural Networks*, 13(6):1331–1341.

Rauber, A., Tomsich, P., and Merkl, D. (2000). parSOM: A parallel implementation of the self-organizing map exploiting cache effects: Making the som fit for interactive high-performance data analysis. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'00)*. Como, Italy.

Reynolds, W. (1993). Hyperbolic geometry on a hyperboloid. In *American Mathematical Monthly*, 100:442–455.

Risden, K., Czerwinski, M. P., Munzner, T., and Cook, D. (2000). An initial examination of ease of use for 2D and 3D information visualizations of web content. In *International Journal of Human Computer Studies*, 53(5):695–714.

Ritter, H. (1999). Self-organizing maps in non-Euclidian spaces. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 97–110. Amer Elsevier.

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

Ritter, H., Martinetz, T., and Schulten, K. (1992). *Neural Computation and Self-organizing Maps*. Addison Wesley Verlag.

Ritter, H. and Schulten, K. (1988). Kohonen's self-organizing maps: exploring their computational capabilities. In *Proceedings of the ICNN'88, IEEE International Conference on Neural Networks*, volume 1, pages 109–116. San Diego.

Russel, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, Inc., 2nd edition.

Saalbach, A., Ontrup, J., Ritter, H., and Nattkemper, T. W. (2005). Image fusion based on topographic mappings using the hyperbolic space. In *Information Visualization*, 4(4):266–275.

Salton, G. (1965). Progress in automatic information retrieval. In *IEEE Spectrum*, 2(28):90–103.

Salton, G. (1987). Historical note: The past thirty years in information retrieval. Technical report, Department of Computer Science, Cornell University, Ithaca, New York.

Salton, G. (1991). The smart document retrieval project. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 356–358.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval.

In *Information Processing and Management*, 24(5):513–523.

Salton, G., Wong, A., and Yang, C. (1975). A vector space model for automatic indexing. In *Communications of the ACM*, 26(1):33–44.

Sammon, Jr., J. W. (1969). A non-linear mapping for data structure analysis. In *IEEE Transactions on Computers*, 18:401–409.

Sangole, A. and Knopf, G. (2002). Representing high-dimensional data sets as close surfaces. In *Journal of Information Visualization*, 1:111–119.

Sangole, A. and Knopf, G. (2003). Visualization of randomly ordered numeric data sets using spherical self-organizing feature maps. In *Computer & Graphics*, 27(6):963–976.

Sarkar, M. and Brown, M. H. (1994). Graphical fisheye views. In *Communications of the ACM*, 37(12):73–84.

Sauren, D. (2005). *Document retrieval and categorization with hyperbolic self organizing mas*. Master's thesis, Faculty of Technology, Bielefeld University.

Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., and Roseman, M. (1998). Navigating hierarchically clustered networks through fisheye and full-zoom methods. In *ACM Transactions on Computer-Human Interaction*, 3(2):162–188.

Schroeder, W., Martin, K., and Lorensen, B. (1997). *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Prentice-Hall, 2nd edition.

Scott, S. and Matwin, S. (1998). Text classification using WordNet hypernyms. In S. Harabagiu, editor, *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 38–44. Association for Computational Linguistics, Somerset, New Jersey.

Sebastiani, F. (2002). Machine learning in automated text categorization. In *ACM Computing Surveys*, 34(1):1–47.

Sebastiani, F., Sperduti, A., and Valdambrini, N. (2000). An improved boosting algorithm and its application to automated text categorization. In *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 78–85.

Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The sematic web revisited. In *IEEE Intelligent Systems*, pages 96–101.

Shepard, R. (1980). Multidimensional scaling, tree-fitting and clustering. In *Science*, 210:390–398.

Shima, K., Todoriki, M., and Suzuki, A. (2004). Svm-based feature selection of latent semantic features. In *Pattern Recognition Letters*, 25(9):1051–1057.

Shneiderman, B. (1992). Tree visualization with Treemaps: a 2D space-filling approach. In *ACM Transactions on Graphics*, 11(1):92–99.

Sieno, D. D. (1988). Adding a conscience to competitive learning. In *Proc. of the ICNN88*, volume I, pages 117–124. San Diego, CA.

Simon, H. A. (1971). Designing organizations for an information-rich world. In M. Greenberger, editor, *Computers, Communication, and the Public Interest*, pages 37–52. The Johns Hopkins Press, Baltimore, MD.

Siolas, G. and d'Alche Buc, F. (2000). Support vector machines based on a semantic kernel for text categorization. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN*. IEEE, Como, Italy.

Skupin, A. (2002). A cartographic approach to visualizing conference abstracts. In *IEEE*

Computer Graphics and Applications, 22(1):50–58.

Skupin, A. (2004). The world of geography: Visualizing a knowledge domain with cartographic means. In *Proceedings of the National Academy of Sciences*, 101(1):5274–5278.

Sommerfeld, A. (1909). Über die Zusammensetzung der Geschwindikeiten in der Relativitätstheorie. In *Physikalische Zeitschrift*, 10:826–829.

Spence, R. (2000). *Information Visualization*. ACM Press/Addison Wesley.

Stasko, J., Catrambone, R., Guzdial, M., and McDonald, K. (2000). An evaluation of space-filling information visualizations for depicting hierarchical structures. In *International Journal of Human-Computer Studies*, 53(5):663–694.

Strasberg, H., Manning, C., Rindfleisch, T., and Melmon, K. (2000). What's related? generalizing approaches to related articles in medicine. In *Proc. AMIA Symp.*, pages 838–42.

Strickert, M. and Hammer, B. (2003). Neural gas for sequences. In T. Yamakawa, editor, *Proceedings of the Workshop on Self-Organizing Networks (WSOM 2003)*, pages 53–58. Kyushu.

Strubecker, K. (1969). *Differentialgeometrie III: Theorie der Flächenkrümmung*. Walter de Gruyter & Co, Berlin.

Thomas, J., Cook, K., Crow, V., Hetzler, B., May, R., McQuerry, D., McVeety, R., Miller, N., Nakamura, G., Nowell, L., Whitney, P., and Wong, P. (1999). Human computer interaction with global information spaces - beyond data mining. In *Proceedings of British Computer Society Conference*.

Thorpe, J. (1979). *Elementary Topics in Differential Geometry*. Springer-Verlag, New York.

Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of ECML-01, 12th European Conference on Machine Learning*, pages 491–502. Springer, Freiburg.

Venna, J. and Kaski, S. (2001). Neighborhood preservation in nonlinear projection methods: An experimental study. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Artificial Neural Networks—ICANN 2001*, pages 485–491. Springer, Berlin.

Venna, J. and Kaski, S. (2005). Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In *Proceedings of WSOM'05, 5th Workshop On Self-Organizing Maps*, pages 695–702. Paris.

Vesanto, J. (1999). SOM-based data visualization methods. In *Intelligent Data Analysis*, 3:111–126.

Villmann, T., Der, R., Herrmann, J., and Martinetz, M. (1994). Topology preservation in self-organizing feature maps: General definition and efficient measurement. In B. Reusch, editor, *Informatik Aktuell - Fuzzy-Logik*, pages 159–166. Springer-Verlag.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. In *Kybernetik*, 14:85–100.

Wagner, R., Ontrup, J., and Scholz, S. W. (2005). Innovative technologies for clustering, monitoring, and evaluation of up-and-coming topics in business information. In *Japanese-German Symposium on Classification*. Tokyo, Japan.

Walter, J. (2003). H-MDS: a new approach for interactive visualization with multidimensional scaling in the hyperbolic space. In *Information Systems, Elsevier*.

Walter, J., Ontrup, J., Wessling, D., and Ritter, H. (2003). Interactive visualization and navigation in large data collections using the hyperbolic space. In *Proceedings of the*

Third IEEE International Conference on Data Mining. IEEE.

Walter, J. and Ritter, H. (1996). Rapid learning with parametrized self-organizing maps. In *Neurocomputing*, pages 131–153.

Walter, J. A. and Ritter, H. (2002). On interactive visualization of high-dimensional data using the hyerbolic plane. In *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 123–131. Edmonton.

Ware, C. (2004). *Information visualization*. Morgan Kaufmann Publishers Inc., San Francisco, USA, 2nd edition.

Wilbur, W. J. and Yang, Y. (1996). An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology. In *Comput. Biol. Med.*, 26(3):209–222.

Williams, R. (1979). *The geometrical foundation of natural structure: A source book of design*. Dover.

Wise, J. (1999). The ecological approach to text visualization. In *J. Am. Soc. f. Information Science*, 50(13):1224–1233.

Wise, J., Thomas, J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V. (1995). Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceedings on IEEE Information Visualization, 1995*, pages 51–58.

Wong, P. C., Foote, H., Leung, R., Adams, D., and Thomas, J. (2000). Data signatures and visualization of scientific data sets. In *IEEE Computer Graphics and Applications*, 20(2).

Wu, Y. and Takatsuka, M. (2005). Fast spherical self organizing map–use of indexed geodesic data structure. In *Proceedings of the 5th Workshop on Self-Organizing Maps 2005, WSOM05*, pages 455–462. Paris.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. In *Information Retrieval*, 1-2(1):69–90.