Text Categorization and Semantic Browsing with Self-Organizing Maps on non-euclidean Spaces

Jörg Ontrup and Helge Ritter

Neuroinformatics Group
Faculty of Technology
Bielefeld University
D-33501 Bielefeld, Germany

Abstract This paper introduces a new type of Self-Organizing Map (SOM) for Text Categorization and Semantic Browsing. We propose a "hyperbolic SOM" (HSOM) based on a regular tesselation of the hyperbolic plane, which is a non-euclidean space characterized by constant negative gaussian curvature. This approach is motivated by the observation that hyperbolic spaces possess a geometry where the size of a neighborhood around a point increases *exponentially* and therefore provides more freedom to map a complex information space such as language into spatial relations. These theoretical findings are supported by our experiments, which show that hyperbolic SOMs can successfully be applied to text categorization and yield results comparable to other state-of-the-art methods. Furthermore we demonstrate that the HSOM is able to map large text collections in a semantically meaningful way and therefore allows a "semantic browsing" of text databases.

1 Introduction

For many tasks of exploraty data analysis the creation of Self-Organizing Maps (SOM) for data visualization, as introduced by Kohonen more than a decade ago, has become a widely used tool in many fields [7].

So far, the overwhelming majority of SOM approaches have taken it for granted to use (some subregion of) a *flat space* as their data model and, motivated by its convenience for visualization, have favored the (suitably discretized) *euclidean plane* as their chief "canvas" for the generated mappings (for a few notable exceptions using tree- or hypercubical lattices see e. g. [1, 8, 14]).

However, even if our thinking is deeply entrenched with euclidean space, an obvious limiting factor is the rather restricted neighborhood that "fits" around a point on a euclidean 2d surface. Recently, it has been observed that a particular type of *non-euclidean spaces*, the *hyperbolic spaces* that are characterized by uniform negative curvature, are very well suited to overcome this limitation [9] since their geometry is such that the size of a neighborhood around a point increases *exponentially* with its radius r (while in a D-dimensional euclidean space the growth follows the much slower power law r^D). This exponential scaling behavior fits very nicely with the scaling behavior within hierarchical, tree-like

Preprint submitted to PKDD-01 *12th April 2001*

structures, where the number of items r steps away from the root grows as b^r where b is the (average) branching factor. This interesting property of hyperbolic spaces has been exploited for creating novel displays of large hierarchical structures that are more accessible to visual inspection than in previous approaches [10].

Therefore, it appears very promising to use hyperbolic spaces also in conjunction with the SOM. The resulting *hyperbolic SOMs* (HSOMs) are based on a tesselation of the hyperbolic plane (or some higher-dimensional hyperbolic space) and their lattice neighborhood reflects the hyperbolic distance metric that is responsible for the non-intuitive properties of hyperbolic spaces.

Since the notion of non-euclidean spaces may be unfamiliar to many readers, we first give a brief account of some basic properties of hyperbolic spaces that are exploited for hyperbolic SOMs. We then illustrate the properties of hyperbolic SOMs with computer experiments focusing on the field of text-mining.

2 Hyperbolic Spaces

Surfaces that possess *negative gaussian curvature* locally resemble the shape of a "saddle", i. e., the negative curvature shows up as a local bending into opposite normal directions, as we move on orthogonal lines along the surface. This may make it intuitively plausible that on such surfaces the area (and also the circumference) of a circular neighborhood around a point can grow faster than in the uncurved case. Requiring a constant negative curvature everywhere, leads to a space known as the *hyperbolic plane H2* (with analogous generalizations to higher dimensions)[2, 20]. The geometry of H2 is a standard topic in *Riemannian geometry* (see, e. g. [19, 13]), and the relationships for the area A and the circumference C of a circle of radius r are given by

$$A = 4\pi \sinh^2(r/2),\ C = 2\pi \sinh(r)\ . \qquad (1)$$

These formulae exhibit the highly remarkable property that both quantities grow *exponentially* with the radius r (whereas in the limit $r \to 0$ the curvature becomes insignificant and we recover the familiar laws for flat $\mathbb{R}^2$). It is this property that was observed in [9] to make hyperbolic spaces extremely useful for accommodating hierarchical structures: their neighborhoods are in a sense "much larger" than in the non-curved euclidean (or in the even "smaller" positively curved) spaces.

To use this potential for the SOM, we must solve two problems: (i) we must find suitable discretization lattices on H2 to which we can "attach" the SOM prototype vectors. (ii) after having constructed the SOM, we must somehow project the (hyperbolic!) lattice into "flat space" in order to be able to inspect the generated maps.

2.1 Projections of Hyperbolic Spaces

To construct an isometric (i. e., distance preserving) embedding of the hyperbolic plane into a "flat" space, we may use a *Minkowski space* [12]. In such a space, the squared distance d^2 between two points (x, y, u) and (x', y', u') is given by

$$d^2 = (x - x')^2 + (y - y')^2 - (u - u')^2 \tag{2}$$

i. e., it ceases to be positive definite. Still, this is a space with zero curvature and its somewhat peculiar distance measure allows to construct an *isometric* embedding of the hyperbolic plane H2, given by

$$x = \sinh(\rho)\cos(\phi),\ y = \sinh(\rho)\sin(\phi),\ u = \cosh(\rho)\ , \tag{3}$$

where (ρ, ϕ) are polar coordinates on the H2 (note the close analogy of (3) with the formulas for the embedding of a sphere by means of spherical polar coordinates in $I\!R^3$!). Under this embedding, the hyperbolic plane appears as the surface M swept out by rotating the curve $u^2 = 1 + x^2 + y^2$ about the u-axis[1].

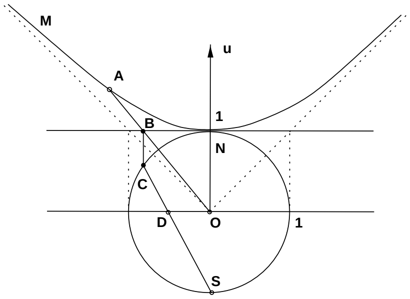


Figure 1: Construction steps underlying *Klein* and *Poincaré*-models of the space H2

From this embedding, we can construct two further ones, the so-called *Klein model* and the *Poincaré model* [2, 3, 6](the latter will be used to visualize hyperbolic SOMs below). Both achieve a projection of the infinite H2 into the unit disk, however, at the price of distorting distances. The Klein model is obtained by projecting the points of M onto the plane $u = 1$ along rays passing through the origin O (see Fig. 1). Obviously, this projects all points of M into the "flat" unit disk $x^2 + y^2 < 1$ of $I\!R^2$. (e. g., $A \mapsto B$). The Poincaré Model results if we add two further steps: first a perpendicular projection of the Klein Model (e. g., a point B) onto the ("northern") surface of the unit sphere centered at the origin (point C), and then a stereographic projection of the "northern" hemisphere onto the unit circle about the origin in the ground plane $u = 0$ (point D). It turns out that the resulting projection of H2 has a number of pleasant properties, among them the preservation of angles and the mapping of shortest paths onto circular arcs belonging to circles that intersect the unit disk at right angles. Distances in the original H2 are strongly distorted in its Poincaré (and also in the Klein) image (cf. Eq. (5)), however, in a rather useful way: the mapping exhibits a strong "fisheye"-effect. The neighborhood of the H2 origin is mapped almost faithfully (up to a linear shrinkage factor of 2), while more distant regions become increasingly "squeezed". Since asymptotically the radial distances and the circumference grow both according to the same exponential law, the squeezing is "conformal", i. e., (sufficiently small) shapes painted onto H2 are not deformed, only their size shrinks with increasing distance from the origin.

[1] the alert reader may notice the absence of the previously described local saddle structure; this is a consequence of the use of a Minkowski metric for the embedding space, which is not completely compatible with our "euclidean" expectations.

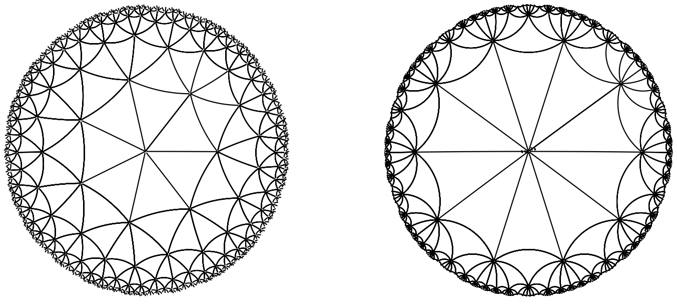


Figure 2: Regular triangle tesselations of the hyperbolic plane, projected into the unit disk using the Poincaré mapping. The leftmost tesselation shows the case where the minimal number ($n = 7$) of equilateral triangles meet at each vertex and is best suited for the hyperbolic SOM, since tesselations for larger values of n (right: $n = 10$) lead to bigger triangles. In the Poincaré projection, only sides passing through the origin appear straight, all other sides appear as circular arcs, although in the original space all triangles are congruent.

By translating the original H2 the fisheye-fovea can be moved to any other part of H2, allowing to selectively zoom-in on interesting portions of a map painted on H2 while still keeping a coarser view of its surrounding context.

2.2 Tesselations of the Hyperbolic Plane

To complete the set-up for a hyperbolic SOM we still need an equivalent of a regular grid in the hyperbolic plane. We use the following results [3, 11]: while the choices for tesselations with congruent polygons on the sphere and even in the plane are such that each grid point is surrounded by the same number n of neighbors are severely limited (the only possible values for n being 3,4,5 on the sphere, and 3,4,6 in the plane), there is an infinite set of choices for the hyperbolic plane. In the following, we will restrict ourselves to lattices consisting of equilateral triangles only. In this case, there is for each $n \geq 7$ a regular tesselation such that each vertex is surrounded by n congruent equilateral triangles. Fig. 2 shows two example tesselations (for the minimal value of $n = 7$ and for $n = 10$), using the Poincaré model for their visualization. While in Fig. 2 these tesselations appear non-uniform, this is only due to the fisheye effect of the Poincaré projection. In the original H2, each tesselation triangle has the same size, and this can be checked by re-projecting any distant part of the tesselation into the center of the Poincaré disk, after which it looks identical (up to a possible rotation) to the center of Fig. 2.

One way to generate these tesselations algorithmically is by repeated application of a suitable set of generators of their symmetry group to a (suitably sized, cf. below) “starting triangle”, for more details cf [15].

3 Hyperbolic SOM Algorithm

We have now all ingredients required for a “hyperbolic SOM”. In the following, we use the regular triangle tesselation with vertex order $n = 7$, which leads to the “finest” tesselation that is possible (in H2, the angles of a triangle uniquely determine its size). Using the construction scheme sketched in the previous section, we can organize the nodes of such a lattice as “rings” around an origin node (i. e., it is simplest to build approximately “circular” lattices). The numbers of nodes of such a lattice grows very rapidly (asymptotically exponentially) with the chosen lattice radius R (its number of rings). For instance, for $n = 7$, Table 1 shows the total number N_R of nodes of the resulting regular hyperbolic lattices with different radii ranging from $R = 1$ to $R = 10$. Each lattice node r carries a prototype vector $\boldsymbol{w}_r \in \mathbb{R}^D$ from some D-dimensional feature space (if we wish to make any non-standard assumptions about the metric structure of this space, we would build this into the distance metric that is used for determining the best-match node). The SOM is then formed in the usual way, e. g., in on-line mode by repeatedly determining the winner node s and adjusting all nodes $r \in N(s,t)$ in a radial lattice neighborhood $N(s,t)$ around s according to the familiar rule

$$\Delta \boldsymbol{w}_r = \eta h_{rs}(\boldsymbol{x} - \boldsymbol{w}_r) \tag{4}$$

with $h_{rs} = \exp(-d^2(r,s)/2\sigma^2)$. However, since we now work on a hyperbolic lattice, we have to determine both the neighborhood $N(s,t)$ and the (squared) node distance $d^2(r,s)$ according to the natural metric that is inherited by the hyperbolic lattice.

Table 1: Node numbers N_R of hyperbolic triangle lattices with vertex order 7 for different numbers R of “node rings” around the origin.

R	1	2	3	4	5	6	7	8	9	10
N_R	8	29	85	232	617	1625	4264	11173	29261	76616

The simplest way to do this is to keep with each node r a complex number z_r to identify its position in the Poincaré model. The node distance is then given (using the Poincaré model, see e. g. [19]) as

$$d = 2\text{arctanh}\left(\left|\frac{z_r - z_s}{1 - \bar{z}_s \cdot z_r}\right|\right) . \tag{5}$$

The neighborhood $N(t,s)$ can be defined as the subset of nodes within a certain graph distance (which is chosen as a small multiple of the neighborhood radius σ) around s.

Like the standard SOM, also the hyperbolic SOM can become trapped in topological defects. Therefore, it is also important here to control the neighborhood radius $\sigma(t)$ from an initially large to a final small value (for details on this and some further means to optimize convergence, see [15]).

4 Experiments

Some introductory experiments where several examples illustrate the favorable properties of the HSOM as compared to the "standard" euclidean SOM can be found in [15].

4.1 Text Categorization

While - similar as for the SOM [7] - a very high classification accuracy is of a secondary importance to visualization, a good classification performance is still important to obtain useful maps of text categories. With the ever growing amount of available information on the Internet, automatic text categorization based on machine learning techniques has become a key task where high-dimensional input spaces with few irrelevant features are involved [4]. Here the goal is the assignment of natural language documents to a number of predefined categories (each document d_j can belong to one, several or none of the categories c_i). Achieving a high classification accuracy is an important prerequisite for automating high volume information organization and management tasks.

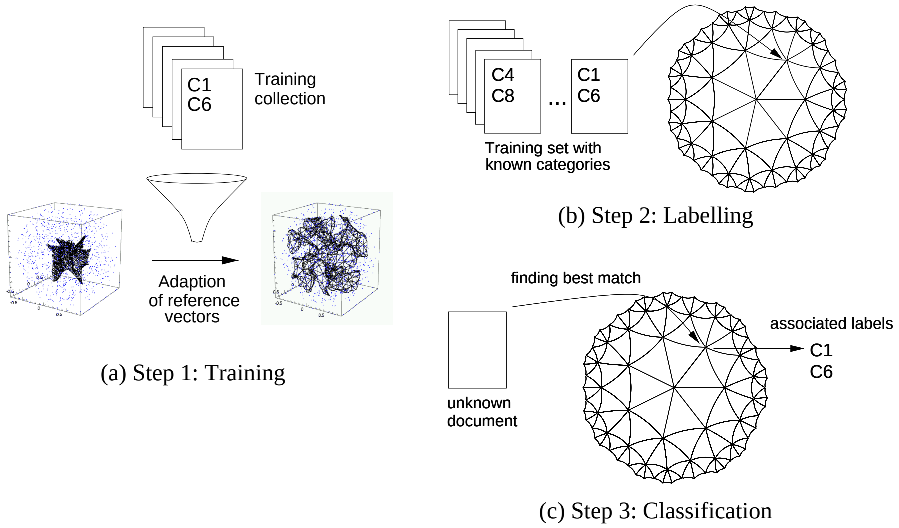
Text Representation. In order to apply the HSOM to natural text categorization, we follow the widely used vector-space-model of Information Retrieval (IR). We applied a word stemming algorithm[2] such that for example the words "retrieved", "retrieval" and "retrieve" are mapped to the term "retrief". The value of f_i of the feature vector $\boldsymbol{f}(d_i)$ for document d_j is then determined by the frequency of which term t_i occurs in that document. Following standard practice [16] we choose a *term frequency* × *inverse document frequency* weighting scheme:

$$f_i = tf(t_i, j)\ log\left(\frac{N}{df(t_i)}\right)\ , \tag{6}$$

where the term frequency $tf(t_i, j)$ denotes the number of times term t_i occurs in d_j, N the number of documents in the training set and $df(t_i)$ the document frequency of t_i, i. e. the number of documents t_i occurs in. Additionally, we built a stop list of the most and least frequent terms specific to the training set and omitted those from the feature vectors, since they have no descriptive function with respect to that text corpus.

HSOM Text Categorization. The HSOM can be utilised for text categorization in the following manner (Fig. 3). In a first step, the training set is used to adapt the weight vectors $\boldsymbol{w}_r$ according to (4). During the second step, the

[2] We applied the *triestem* function of the SMART system by G. Salton and C. Buckley (ftp://ftp.cs.cornell.edu/pub/smart/).



(a) Step 1: Training

(b) Step 2: Labelling

(c) Step 3: Classification

Figure 3: Text categorization with the HSOM: First the training set is used to build an internal model of the collection represented by the HSOM's reference vectors. In *(b)* for each training document the winner nodes are labelled with the document's category. These labels are used in the classification step *(c)* where an unknown document is "thrown" onto the map and labelled with the categories of its corresponding best match node.

training set is mapped onto the HSOM lattice. To this end, for each training example d_j its best match node s is determined such that

$$|\boldsymbol{f}(d_j) - \boldsymbol{w}_s| \leqslant |\boldsymbol{f}(d_j) - \boldsymbol{w}_r| \quad \forall r \, , \tag{7}$$

where $\boldsymbol{f}(d_j)$ denotes the feature vector of document d_j, as described above. After all examples have been presented to the net, each node is labelled with the union U_r of all categories that belonged to the documents that were mapped to this node. A new, unknown text is then classified into the union U_s of categories which are associated with its winner node s selected in the HSOM. In order to evaluate the HSOM's categorization performance, we furthermore use $cos(\boldsymbol{w}_s, \boldsymbol{f}(d_j))$ as a confidence measure for the classification result.

Text Collection. The text collection consists of movie reviews taken from the rec.art.movies.reviews newsgroup. Genre information from the Internet Movie Database (http://www.imdb.com) was used to build a joined database containing the review texts plus the genres from their corresponding movies as the categories. To build the training text collection, for each of the most prominent 17 categories 20 movies were randomly selected. For each of these movies, 3 review texts were chosen by chance. Therefore, the training collection contained 1020 distinct documents. The test text collection was constructed in the same manner with the restriction that it must not contain any document of the training set. After word stemming and stop word removal we arrived at approximately 5000 distinct terms for the construction of the feature vectors.

Performance Evaluation. The classification effectiveness is commonly measured in terms of precision P and recall R [17], which can be estimated as

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i}\ , \tag{8}$$

where TP_i and TN_i are the numbers of documents correctly classified, and correctly not classified to c_i, respectively. Analogous, FP_i and FN_i are the numbers of documents wrongly classified and not classified to c_i, respectively. By adjusting a threshold which is compared with the confidence value $cos(\boldsymbol{w}_s, \boldsymbol{f}(d_j))$ of the classifier, the number of retrieved documents can be controlled. In order to obtain an overall performance measure for all categories, we applied the *microaveraging* method [21]. Furthermore, the *breakeven* point of precision and recall, i. e. the value at which $P = R$ is a frequently given single number to measure the effectiveness determined by both values P and R [17].

In order to assess the HSOM's performance for text categorization, we have used a k-nearest neighbour (k-NN) classifier which was found to show very good results on text categorization tasks [21]. Apart from boosting methods [18] only support vector machines [5] have shown better performances. The confidence level of a k-NN classifier to assign document d_j to class c_i is

$$C_i^{k\text{-NN}}(d_j) = \sum_{d_z \in TR_k(d_j)} a_{iz} \cdot cos(d_j, d_z)\ , \tag{9}$$

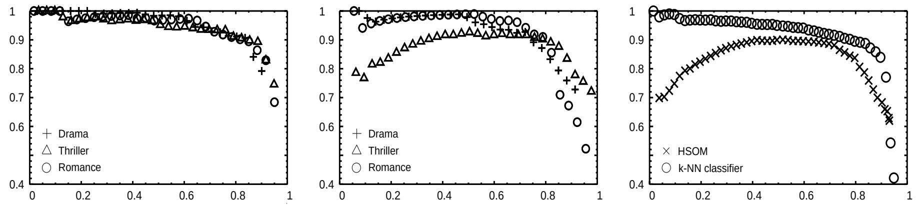
where $TR_k(d_j)$ is the set of k documents d_z for which $cos(d_j, d_z)$ is maximum. The assignment factor a_{iz} is 1, if d_z belongs to category c_i and 0 otherwise. According to [21, 5] we have chosen the $k = 30$ nearest neighbours.

Text Categorization Results. Precision-recall-diagrams for three categories and the microaveraged diagrams for all categories are shown in Fig. 4. The single category and microaveraged break-even points are layed out in Table 2.

It is notable that the HSOM performs significantly worse if only a few documents are recalled, but the precision in cases of high recall values is very close to that of the k-NN classifier. Since one is usually interested in high precision in conjunction with high recall, the suboptimal results for low recall values do not really affect the usefulness of the HSOM for the purpose of text categorization[3]. Thus, our results indicate that the HSOM does not perform better than a k-NN classifier, but it does not play significantly worse either. Since the main

[3] We also believe that a more clever heuristic than the simple distance to the bestmatch node in order to determine the evidence value of a classification will further improve accuracy for low retrieval rates.

purpose of the HSOM is the visualization of relationships between texts and text categories, we believe that the observed categorization performance of the HSOM compares sufficiently well with the more specialized (non-visualization) approaches to warrant its efficient use for creating insightful maps of large bodies of document data.



(a) k-NN **(b)** HSOM **(c)** Microaveraged

Figure 4: Precision-recall-diagrams for the three categories *Drama*, *Thriller* and *Romance*. *(a)* shows the results for the k-NN classifier, *(b)* for the HSOM. In *(c)* the microaveraged diagrams for both methods are shown.

Table 2: Precision-recall breakeven points for the most prominent categories. In most cases the k-NN performs better than the HSOM, but for the categories "Animation", "Fantasy" and "Musical" the HSOM yields better results.

	Action	Advent.	Animation	Comedy	Crime	Docum.	Drama	
HSOM	81.6	75.4	86.9	81.3	84.5	86.7	82.5	
k-NN	87.3	83.0	84.5	87.6	90.5	98.0	85.8	
	Fantasy	Horror	Musical	Mystery	Romance	Sci-Fi	Thriller	μ-avg.
HSOM	81.6	78.6	82.5	84.6	82.8	76.2	86.8	81.1
k-NN	75.0	88.9	81.2	86.1	87.8	89.3	89.1	86.4

4.2 Semantic Browsing

A major advantage of the HSOM is its remarkable capability to map high-dimensional similarity relationships to a low-dimensional space which can be more easily handled and interpreted by the human observer. This feature and the particular "fisheye" capability motivates our approach to visualize whole text collections with the HSOM. With just as little as 5 rings (c.f. Table 1), we can handle well over 500 prototype vectors which are able to represent different types of texts. The nodes are labelled with those document titles which resemble their prototype vectors most closely. We additionally map symbols to the nodes which correspond to the categories associated with the prototypes. We can now interactively change the visual focus to those regions which show an interesting structure. In Fig. 5(a) for example we have "zoomed" into a region of the map which indicated a cluster of “Animation” films. As a closer inspection shows, this region of the map resembles movie reviews all connected to Disney’s typical animations released during Christmas time. In Fig. 5(b) the focal view was moved to a region connected to “Action” and “Fantasy” films. It does not only show the movies of the “Batman” series in a limited area of the map, but also a “Zorro” movie in the neigborhood - which makes a lot of sense, as the main characters of the films indeed have a semantic relation.

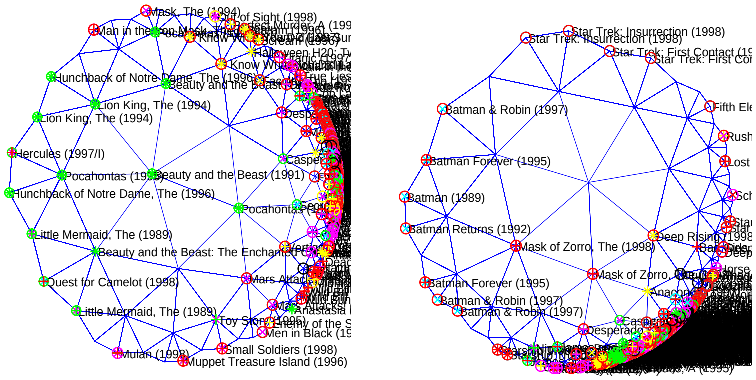


Figure 5: By moving the visual focus of the HSOM, a large text collection can be browsed quite elegantly. In this example, the movie titles of their corresponding review texts (there might be several reviews for one movie) are mapped. In *(a)* reviews of Disney’s animations have been moved into the centre, *(b)* shows a region of the map containing the “Batman” movies. Note, that the HSOM also mirrors the semantical connection between “Zorro” and “Batman”.

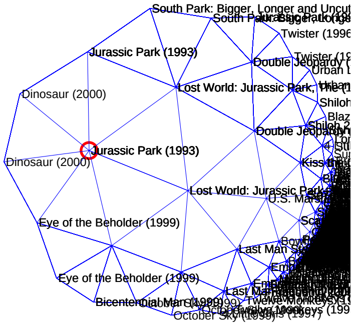


Figure 6: Mapping a new text.

As illustrated in Fig. 6, the HSOM might also be used to classify an unknown text by displaying its relationship to a previously acquired document collection. In this example an unknown review text for the film “Jurassic Park” was mapped. The map was then zoomed to that node which most closely resembled the given input text, which in this case was another review describing the same film. Note, that the neighborhood is occupied by reviews describing the sequel “The Lost World”, respectively the “Dinosaurs” animation. By mapping a complete unknown document collection to a previously formed HSOM, relevant text documents can be discovered. In Fig. 7 the HSOM is used as a filter to display only those documents which belong to a semantic region of interest.

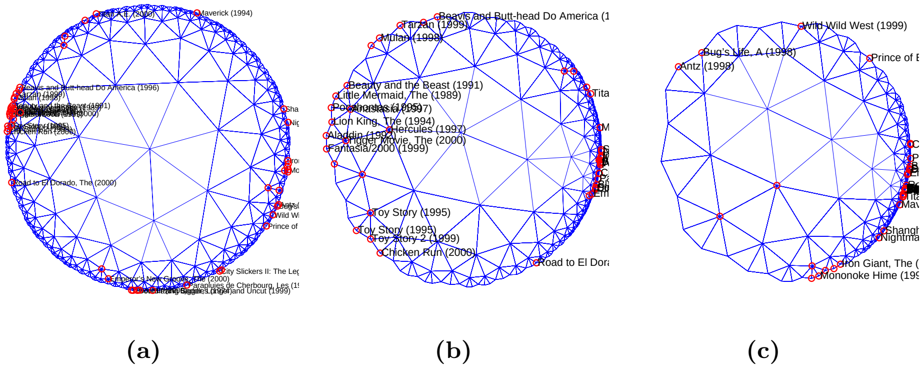


Figure 7: Discovering of relevant documents in whole collections. In this example an unknown text collection is mapped onto the HSOM, but only those items are visualized which belong to a selected category, in this case animation movies. The central view in *(a)* points to a set of document clusters, which can be interactively zoom into. In *(b)* the largest cluster has been moved into focus. It mainly shows Disney animations and films from the "Toy story" series a bit farther down. In *(c)* the cluster in the top left contains "A Bug's Life" and "Antz", whereas the cluster in the bottom right belongs to movies connected to Japanese animations.

5 Discussion

Our results show that the HSOM is not only applicable to automated text categorization, but also specifically well suited to support "semantic browsing" in large document collections. Our first experiments indicate that the HSOM is able to exploit the peculiar geometric properties of hyperbolic space to successfully compress complex semantic relationships between text documents onto a two dimensional projection space. Additionally, the use of hyperbolic lattice topology for the arrangement of the HSOM nodes offers new and highly attractive features for interactive navigation in this way. Large document databases can be inspected at a glance while the HSOM provides additional information which was captured during a previous training step, allowing e. g. to rapidly visualize relationships between new documents and previously acquired collections.

Future work will address more sophisticated visualization strategies based on the new approach, as well as the evaluation for other widely used text collections, such as Reuters, Ohsumed or Pubmed.

References

1. D.S. Bradburn. Reducing transmission error effects using a self-organizing network. In *Proc. of the IJCNN89*, volume II, pages 531–538, San Diego, CA, 1989.
2. H. S. M. Coxeter. *Non Euclidean Geometry.* Univ. of Toronto Press, Toronto, 1957.

3. R. Fricke and F. Klein. *Vorlesungen über die Theorie der automorphen Funktionen*, volume 1. Teubner, Leipzig, 1897. Reprinted by Johnson Reprint, New York, 1965.
4. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. Technical Report LS8-Report 23, Universität Dortmund, 1997.
5. T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998.
6. F. Klein and R. Fricke. *Vorlesungen über die Theorie der elliptischen Modulfunktionen.* Teubner, Leipzig, 1890. Reprinted by Johnson Reprint, New York, 1965.
7. T. Kohonen. *Self-Organizing Maps.* Springer Series in Information Sciences. Springer, second edition edition, 1997.
8. Pasi Koikkalainen and Erkki Oja. Self-organizing hierarchical feature maps. In *Proc. of the IJCNN 1990*, volume II, pages 279–285, 1990.
9. John Lamping and Ramana Rao. Laying out and visualizing large trees using a hyperbolic space. In *Proceedings of UIST'94*, pages 13–14, 1994.
10. John Lamping, Ramana Rao, and Peter Pirolli. 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*, Denver, May 1995. ACM.
11. W. Magnus. *Noneuclidean Tesselations and Their Groups.* Academic Press, 1974.
12. Charles W. Misner, J. A. Wheeler, and Kip S. Thorne. *Gravitation.* Freeman, 1973.
13. Frank Morgan. *Riemannian Geometry: A Beginner's Guide.* Jones and Bartlett Publishers, Boston, London, 1993.
14. H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-organizing Maps.* Addison Wesley Verlag, 1992.
15. Helge Ritter. Self-organizing maps in non-euclidian spaces. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 97–108. Amer Elsevier, 1999.
16. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
17. F. Sebastiani. Machine learning in automated text categorisation: a survey. Technical Report IEI-B4-31-1999, Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, 1999.
18. F. Sebastiani, A. Sperduti, and N. Valdambrini. 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, 2000.
19. Karl Strubecker. *Differentialgeometrie III: Theorie der Flächenkrümmung.* Walter de Gruyter & Co, Berlin, 1969.
20. J.A. Thorpe. *Elementary Topics in Differential Geometry.* Springer-Verlag, New York, Heidelberg, Berlin, 1979.
21. Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1-2(1):69–90, 1999.