Automated text categorization of bibliographic records

Sándor Darányi, Johan Eklund
Swedish School of Library and Information Science

2006-09-26

1 Introduction

LIVA (Library Information Analysis and Visualization) is a research and development project of the Swedish School of Library and Information Science in Borås (SSLIS), Bibliotekscentrum Sverige AB, and BTJ, in cooperation with a number of project libraries and with funding for 2005-2007 from the Knowledge Foundation (KKS). Based on analysis of data from the project libraries and content providers – prominently Lund Public Library, Nordiska museet, SCB (Statistics Sweden), TPB (The Swedish Library of Talking Books and Braille), Southern Älvsborg Hospital Library and Dandelon, a German portal and search engine – the goal of the project is to bring competitive functionality in terms of language technology, classification research, information retrieval (IR) and information visualization to special/public library OPACs.

This interaction between the above components is one of the delicate areas of study where active research is going on worldwide. As a point of departure we assume that the quality of classification has an impact both on IR results and on information searching by browsing, as enabled by information visualization. In an automated environment, classification quality, on the other hand, heavily depends on linguistic pre-processing of the input material, this being provided for example by language technology tools. One might say that the outcome of knowledge organization depends on the abilities of the linguistic and statistical tools applied.

2 Traditional classification

Library classification can be described as a procedure involving the coding and organizing library materials (books, serials, audiovisual materials, computer files, maps, manuscripts etc.) according to their conceived subject. A classification consists of tables of subject headings and classification schedules used to assign a class number to each item being classified, based on that item's subject [6].

Whereas as late as until modern times, manual indexing and classification for cataloguing were seen as inventories designed for controlling the collection and for information needed for new acquisitions, during the 20th century they have become indispensable tools for subject searches as well [15]. In this sense, we can say that classification is the organization of library materials by a hierarchy of subject categories [7].

3 Motivation

The motivation for research in LIVA was due to several developments in different fields during the past four decades. Prominently we must name the following:

- The birth and evolution of the World Wide Web. With the exponential growth of the number of web pages, and potential direct access to document databases, huge and ever-expanding document collections came into existence. Text documents being the most

important of them, manual classification in local or virtual collections cannot keep up with the proliferation of digitized documents. Worldwide efforts at digitization add a new dimension to this problem. Unless computerized solutions are worked out, users lose access to the majority of these collections;

- Due to the enormously increased need to handle larger and larger quantities of documents, there is a growing need emphasized by increased connectivity and availability of document bases of all types at all levels in the information chain. But this interest is also due to the fact that text categorization techniques have reached accuracy levels that rival the performance of trained professionals, and these accuracy levels can be achieved with high levels of efficiency on standard hardware/software resources. This means that more and more organizations are automating all their activities that can be cast as text categorization tasks [28];

- Interaction between natural language processing, information retrieval and machine learning results in an ongoing exchange of issues, solutions, data sets and evaluation methods; and

- Findings from the above fields for commercial product development are increasingly available.

4 Roadmap

Since the project paves the way for software product development, working out interaction between the different research components is crucial to the success of LIVA. This means finding a conceptual and methodological common denominator enabling the automatic indexing, automatic classification, information visualization and information retrieval modules working together. To this end, we concentrate on the vector space model in information retrieval for document content representation, and the energy metaphor for content grouping.

The implementation of this integration happens on three levels. The first research problem, now solved, was to find ways and means for the automatic indexing and automatic classification of bibliographic records, in the context of the SAB classification system. We will refer to these two procedures as text categorization. The solution involves the use of natural language processing (NLP) tools for Swedish and the concept of information gain in machine learning [8], both handling textual elements of such records, and support vector machines (see below).

A next level of integration demanded interaction between the text categorization and the information visualization components. This is now in the experimental phase. We are experimenting with three visualization metaphors: document galaxies [34], force-directed placement [33] such as in "cauliflower space" [14, 2], and contour maps or thematic landscapes [34]. Whereas document galaxies and contour maps support navigation in a database, force-directed placement methods give the user an overview of both the information searching process and single steps of information retrieval.

Finally, information visualization and information retrieval need to be integrated as well. Apart from trivial solutions such as the visualization of the vector space model or the latent semantic indexing (LSI) IR model for navigation parallel to retrieval [3, 22], we are working on the implementation of the force-directed placement methods and their matching with the relevance feedback IR model [25]. To this end, we conjecture that a query retrieving documents from a database manifests a classification suitable as a training sample for supervised learning, and relevance feedback for query modification leads to a changing sequel of such classification samples [23].

5 Definitions

In order to give the reader a clear perception of the central concepts in this article a few definitions follow.

5.1 Text categorization

Informally, *text categorization* can be defined as the process of assigning text documents to categories or classes. Let $\mathcal{D}$ be a set of documents and $\mathcal{C}$ a set of categories. We can then formalize the categorization of $\mathcal{D}$ by defining a target function $\Psi : \mathcal{D} \times \mathcal{C} \rightarrow \{0, 1\}$, where 1 indicates membership and 0 non-membership of a particular category.

5.2 Automated text categorization

By *automated text categorization* we refer to text categorization performed by an algorithm contained in a computer program.

5.3 Classification

By *classification* we refer to categorization of documents into predefined and labeled categories.

5.4 Clustering

By *clustering* we refer to categorization of documents into a set of groups which arise from inter-similarities between the documents in $\mathcal{D}$. Formally, clustering is a partitioning of $\mathcal{D}$. We distinguish between *partitional* clustering, where no relations between the obtained clusters are stored, and *hierarchical* clustering, where relations between the clusters are stored in a hierarchical (tree) structure.

5.5 Feature space

A *feature space* is a vector space used to store and analyze objects (for instance documents) by representing them with vectors containing measured feature values. In the case of documents, term frequencies within documents as well as document collections are often used to calculate suitable vector coordinates. This representation form easily allows measurement of document similarities as well as detecting the cluster structure of a collection. Both "ordinary" Euclidean vector spaces as well as more abstract Hilbert spaces can be used for this purpose.

5.6 Cross-validation

A common evaluation method in supervised machine learning, *cross-validation* is a procedure that involves partitioning the document set $\mathcal{D}$ into a collection of subsets $\mathcal{D}' = \{D_1, \ldots, D_k\}$ followed by choosing one subset D_i for training, i.e. for inducing a model for the classification of $\mathcal{D}$, while the remaining subsets $D_j \in (\mathcal{D}' - D_i)$ are used for validation. This process can be repeated over all k subsets in $\mathcal{D}'$ – so called *k-fold cross-validation* – followed by the calculation of the average over all k measurements.

5.7 Machine learning

The field of *machine learning* methods generally involves computerized learning of a task [24]. This task can consist of finding a syntactic model for a language, inducing a regressive model for economical change, or categorizing information objects (for instance documents and terms). The process of learning formally entails choosing a hypothesis h of a space of hypotheses $\mathcal{H}$ about the phenomenon at hand.

Machine learning methods are generally categorized as either *supervised*, where h is chosen to fit a set of labeled input patterns, and *unsupervised*, for which no data labels are available (or used). In many machine learning applications the system uses a subset X_t (called a *training set*) of the entire data set X to induce a model for X.

5.7.1 Overfitting

A machine learning hypothesis is characterized by *overfitting* if it is too granular (has too many parameters) in relation to the data set. This typically means that the model fits the training set well, but is not capable of generalizing to the underlying data population.

6 Automatic indexing

Because of its seminal importance for decades of consecutive research, including the ideas we are going to introduce and combine here as well, we must start our treatise with the brief description of the *vector space model* (VSM).

In the field of information retrieval (IR), the vector space model is an important, well-understood and extensively researched classical model, which has been widely used to process texts efficiently and retrieve information for some forty years [26]. The VSM is called so because each document and query is mapped to a point in the feature space based on frequencies of keywords appearing in the text. The feature space is mathematically modelled by the orthonormal Euclidean space, i.e., the space (or geometry) defined by a system of pairwise orthogonal coordinate axes corresponding to index terms. So far, the Euclidean geometry is the only type of space used in the VSM in general, but non-Euclidean geometry is becoming increasingly important in modern science and technology.

As Salton *et al.* suggest in a later article, we can consider a document space $\mathcal{D}$ consisting of documents d_i, each identified by one or more index terms t_j, which may be weighted according to their importance. If they are not, their weights are restricted to 0 and 1. Whereas such a document space can be easily visualized in three dimensions of Euclidean space if the document has three index terms, the example can be extended to t dimensions when t different index terms are present. In that case, each document d_i is represented by a t-dimensional vector $\mathbf{d}_i = (w_1, \ldots, w_t)$, where w_i represents the weight of the ith term.

Given the index vectors for two documents, it is possible to compute a similarity coefficient between them, $\sigma : \mathcal{D} \times \mathcal{D} \to [0, 1]$, which reflects the similarity in the corresponding terms and term weights. Such a similarity measure might be the inner product of the two vectors, or alternatively an inverse function of the angle between the corresponding vector pairs; when the term assignment for two vectors is identical, the angle will be zero, producing a maximum similarity measure [27]. Documents are regarded relevant to a query vector, and ranked with respect to their similarity values. The similarity measure σ can be expressed in the general form

$$\sigma(d_i, d_j) = \frac{\langle \mathbf{d}_i, \mathbf{d}_j \rangle}{\Delta} \tag{1}$$

where Δ is a function that normalizes σ to the interval $[0, 1]$.

7 Support vector machines

Support vector machines (SVM) [18] is a set of supervised machine learning methods used for both classification and regression tasks, based upon the ideas in the Vapnik-Chervonenkis theory of structural risk minimization. Let $h \in \mathcal{H}$ be hypothesis on a given classification problem and $P(error(h))$ the probability that h will incorrectly classify a previously unseen document. This

error can be expressed using the following inequality [31]:

$$P(error(h)) \leq train\_error(h) + \varepsilon(d, n) \tag{2}$$

where d denotes the Vapnik-Chervonenkis dimension (VC) of h, n the number of training examples and ε a function monotonically increasing with d. The task is to find an optimal h which minimizes $P(error(h))$. If the VC dimension is low the training error (the first right-hand side term in (1)) will be high and if the VC dimension is high the second right-hand term in (1) will be high, indicating *overfitting* to the training examples.

Given a binary classification situation with $+1$ and -1 as class labels we let $\mathcal{H}$ consist of hyperplane classifiers with the following general definition:

$$h(\mathbf{d}) = sign(\mathbf{w} \cdot \mathbf{d} + b) = \begin{cases} +1 & \text{if } \mathbf{w} \cdot \mathbf{d} + b > 0 \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

Let R be the radius of a ball containing all training examples $\mathbf{d}_i$ If we require that $|\mathbf{w} \cdot \mathbf{d}_i + b| \geq 1$ for all examples and let $\Lambda = ||\mathbf{w}||$ we can according to [32] delimit the values for d according to

$$d \leq min([R^2\Lambda^2], n) + 1 \tag{4}$$

From (3) it follows that the VC dimension can be minimized by searching for the smallest value for $||\mathbf{w}||$, which in turn implies a maximal hyperplane separating the two classes. This hyperplane will be constructed as a linear combination of the training examples being closest to the hyperplane, called the *support vectors*.

Combined with the idea of a maximum-margin hyperplane classifier SVMs normally apply a (often non-linear) transformation Φ into a Hilbert space, associated with a kernel function K in the following fashion (let $\mathbf{x}_i$ and $\mathbf{x}_j$ be vectors):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \tag{5}$$

where $\langle \cdot, \cdot \rangle$ denotes *inner product*. This procedure, termed the "kernel trick" makes it possible to use a linear classifier in a high-dimensional (possibly infinite-dimensional) feature space, since it is defined by the inner product in the space at hand. The point of doing so is to find a feature space that is optimized for separating the classes and provide maximum classification accuracy. A popular choice for Φ, also applied in this article, is a radial basis function (RBF):

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2} \tag{6}$$

To investigate the possibilities of automatizing the process of classification with SVM, based on the knowledge contained in pre-classified documents, we have commenced processing MARC bibliographic records from BURK-sök®, BTJ. The records are manually categorized according to the SAB classification scheme and contain value-adding information like reviews, content descriptions, and tables of contents. In the initial phase the sample records were processed using various language technology tools (stopword elimination, stemming, latent semantic analysis) to obtain an optimal representation of the information content in the records. The data was subsequently mapped into a high-dimensional feature space by a kernel function optimized for the task, and classified using a support vector machine. When performing cross-validation of the classification results the *accuracy*, i.e. the proportion of correctly classified items, obtained for the most frequent SAB classes in the material was found to be in the range of 93-96%.

7.1 Optimizing the input to improve SVM performance

One commonly encountered problem, called the "curse of dimensionality" [29], in automatized text categorization is that the number of features is much higher than the number of examples

available for training a classification model. Beside the immediate risk for overfitting this also puts high demands on the computational resources. Within the LIVA project several approaches to overcome this problem have been tested. One method is to use statistical dimension reduction techniques such as *singular value decomposition* (SVD)[12]. This technique is also a powerful tool to identify hidden (or latent) patterns in a set of data and has therefore been successfully applied in document indexing (latent semantic indexing, [5]) with the aim to bring together semantically related terms.

Another strategy is to use a feature selection technique to reduce the number of terms in the vocabulary before the training commences. Contrary to how SVD works the reduced term space will not be based on a statistical compression of all available data, but rather a filtering of the index terms based on a measure of their class discrimination ability. One measure that has been particularly useful to this end is *information gain* (IG) [8], which is an important tool in the creation of decision trees. This measure is closely related to Shannon's *information entropy* which was designed to quantify the amount of information in a communication signal. The rationale behind both measures is that the amount of information in a message about an event X is positively related to the uncertainty about the outcome of X. Formally, let X be an event and $x_1, \ldots, x_n$ its possible outcomes. Then the entropy of X is defined

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

where $p(x_i)$ denotes the probability of x_i. If an event X is conditioned on an event $Y = v$, where v is a specific state we write the *conditional entropy* $H(X|Y = v)$ as follows:

$$H(X|Y = v) = -\sum_{i=1}^{n} p(x_i|Y = v) \log_2 p(x_i|Y = v)$$

The information gain of an event X conditioned on a state v is defined as difference in entropy

$$IG(X, v) = H(X) - H(X|v)$$

Our strategy has been to compute the IG for each term and rank the terms in descending order according to their IG value, after which a cutoff value such as 1000 or 2000 terms has been applied (i.e. we keep the 1000 highest-ranked terms). Initial tests with SVM in combination with "class targeted" term lists yield a cross-validation accuracy of above 99% after parameter optimization of the SVM kernel.

8 Quantum clustering

An example for integrated information visualization is how we utilize contour maps. In the first step, we use dimension reduction techniques to limit the list of word forms extracted from the records to only those which impact the index term distribution to the greatest extent. Then we generate a contour map of the documents indexed by these terms. Put it simply, the "longitude" and the "latitude" of the map are computed by SVD, whereas its "altitude" – based on 2- or more-dimensional distances between document coordinates – is estimated by quantum clustering (QC) [16, 17]. The result is a three-dimensional potential map. The task is to find the optimal landscape in which terms and their documents inhabit their respective contour zones. An example of a map amplified with QC is given below in *Figure 1.*

For the intellectual background of this method, physical energy "driving" the arrangement of document content in document space as a metaphor has been in practice for at least two decades now, with good results. As a general framework for this way of thinking, we mention the concept of data physics [21, 10]. One particular such method called force-directed placement uses the force of a spring [33, 4] or the strong nuclear force between subatomic particles [19, 11]

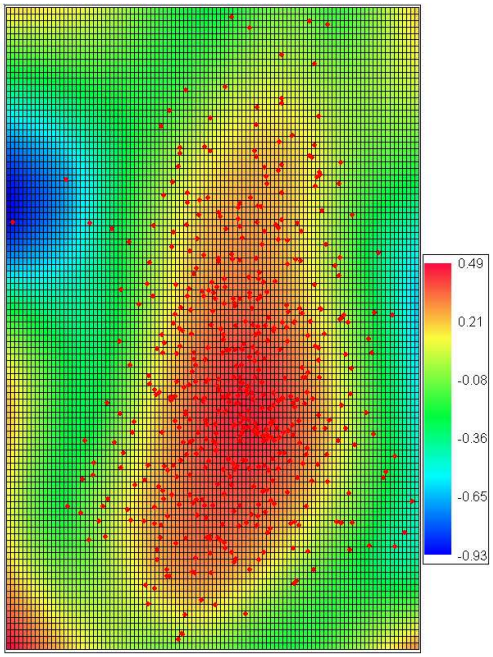

Figure 1: *BURK-sök® sample, class Oh [Sociala frågor och socialpolitik], 544 documents x 8928 terms, potential computation based on factors 1 and 2, terrain by minimum curvature*

as a metaphor for computing the location of document representations. A next example is the computation of eigenvalues in latent semantic indexing methods, underlying the calculation of document and term coordinates – namely eigenvalues in real physical systems, for example in quantum physics, represent the energy content of the system. Finally, there are researchers who, for different reasons, consider quantum mechanics as the model proper for information retrieval [9, 30], and move over to non-Euclidean geometries such as Hilbert space for understanding relevance [30], or to hyperbolic geometry for improved information retrieval [13]. In short, the future will be partly non-linear, non-Euclidean and non-Newtonian.

QC represents documents and terms by Gaussian wave functions whose sum is $\psi(\mathbf{x})$. This means that ψ is modelled as a Parzen window estimator of the form

$$\psi(\mathbf{x}) = \sum_j e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2}$$

By using the Schrödinger equation from quantum mechanics, i.e.

$$H\psi \equiv \left(-\frac{\sigma^2}{2}\nabla^2 + V(\mathbf{x})\right)\psi = E\psi \tag{7}$$

where $V(\mathbf{x})$ is the potential and E is the eigenvalue of ψ, we search for the Schrödinger potential for which $\psi(\mathbf{x})$ is the ground state. The minima of the potential function define our cluster centers. In a supervised learning situation, if the number of classes is known beforehand, QC can be fine-tuned for this number and reproduce the original classification by automatic means. The potential $V(\mathbf{x})$ can be derived from equation (7) to the following expression:

$$V(\mathbf{x}) = E - \frac{d}{2} + \frac{1}{2\sigma^2\psi}\sum_j ||\mathbf{x}-\mathbf{r}_j||^2 e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} \tag{8}$$

For a proof of this, please consult the Appendix of this article.

In our reciprocal implementation of the QC algorithm, for the sake of visual convenience, document cluster centroids, i.e. the most frequent and typical documents are peaks in the

content landscape; whereas outliers, i.e. rare, atypical documents are placed at the deepest points of the catchment basin *(Figures 2-3)*.

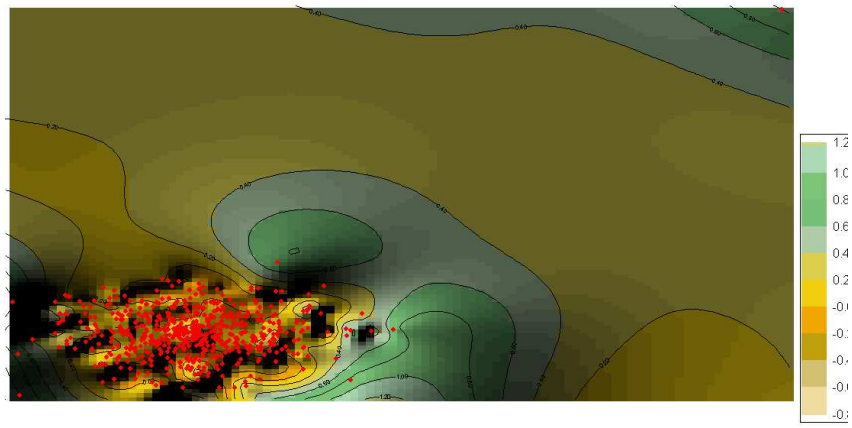

Figure 2: *Contour landscape, BURK-sök® sample, class Qb [Företagsekonomi] 594 documents x 12407 terms, potential computation based on factors 1 and 2, colour relief map by minimum curvature*

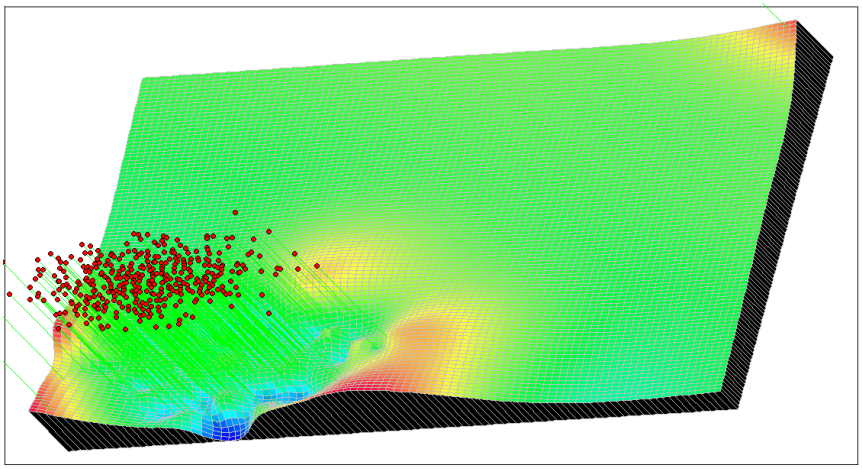

Figure 3: *Three-dimensional view, BURK-sök® sample, class Qb [Företagsekonomi] 594 documents x 12407 terms, potential computation based on factors 1 and 2, colour relief map by minimum curvature*

We also mention in passing that one can generate joint contour maps of documents and their index terms by QC, using e.g. SVD for computing the document and index term coordinates. The reason for this is that in SVD and related methods, term and document spaces share a joint basis, i.e. they correspond to different linear combinations of the same unit vectors.

9 Self-organizing maps

In the early 80's Teuvo Kohonen at the Helsinki University of Technology published a new method, the *self-organizing map* (SOM), for visualizing a multi-dimensional data set in 2 dimensions, suitable for a computer screen or printed material. It is an example of unsupervised learning and is designed with the human brain as a template, in that sense that different input patterns are mapped to different regions of the map. This is analogous to how different sensory inputs are handled by different parts of the brain. The map is arranged as a 2-dimensional grid and implemented as a neural network with connections between all adjacent nodes. One particularly attractive feature of the SOM algorithm is its capacity to preserve the topological features of the original data.

In order to be fit to categorize a certain data set the map is first trained with a sample of data to create a structure of nodes "attracting" certain data patterns. This process can briefly be describe as follows:

1. Initialize the map by associating weight vectors with each of its nodes. These vectors can be randomized or based on the input.

2. Select a vector $\mathbf{x}$ from the set of input data. Calculate $||\mathbf{x} - \mathbf{w}_j||$ for every node vector $\mathbf{w}_j$. The node having the most similar vector to the input is called the *best matching unit* (BMU).

3. Update the vectors of the BMU and its neighboring nodes according to

$$\mathbf{w}_j^{t+1} = \mathbf{w}_j^t + \theta(j,t)\alpha(t)[\mathbf{x} - \mathbf{w}_j^t]$$

   where $\theta(j,t)$ is a function decreasing with distance between the node j and the BMU and $\alpha(t)$ is a function monotonically decreasing with each time step t.

For a more detailed treatment of this procedure, please see [20]. The actual distribution of data on the map is simply performed by for each data point $\mathbf{x}$ identifying the node in the map having the smallest distance to $\mathbf{x}$. In the originally proposed algorithm the number of nodes are fixed and the map consists of only one layer. There also exist extensions to the algorithm involving growing and even hierarchical layers of maps.

In the LIVA project the self-organizing map has been studied as one of several approaches to arranging a set of documents to prepare them for visualization, optionally with a third dimension added indicating data density. The density can be measured by for instance a Parzen window estimator [1] or the QC algorithm.

10 Conclusions

The steady increase in the amount of digital information and the demands for cataloging and access tools to manage the information overload have led to an interest in automatic text categorization with the overall expectation of reducing the human labor involved in traditional classification or even replacing it to a limited extent. There have been a few research projects and some related studies on the feasibility of the Library of Congress Classification (LCC) and the Dewey Decimal Classification (DDC) schemes as a framework for the automatic classification of digital information. Their latest overview can be found in a recent paper by Yi [35].

With its aim to upgrade existing library software by the integration of automatic indexing and classification amounting to automatic text categorization, information visualization and information retrieval, the LIVA project is looking for alternative solutions to the same problem on a smaller scale. The combination of the methods we have selected by the half lifetime of the project holds the promise that a reasonable working solution will emerge next year.

Acknowledgements

We are grateful to Åke Viberg (Uppsala University) for making Svenskt OrdNät available to LIVA, and to Martin Volk (Stockholm University) and Dimitrios Kokkinakis (Göteborg University) for their permission to use the natural language processing tools they have developed.

References

[1] Babich, G.A. and Camps, O.I. (1996). Weighted Parzen Windows for Pattern Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):567-570.
Available at: http://dx.doi.org/10.1109/34.494647

[2] Beale, R., McNab, R.J. and Witten, I.H. (1997). Visualizing sequences of queries: A new tool for information retrieval. *Proceedings of the IEEE Conference on Information Visualization*, 57-62.

[3] Berry, M.W., Dumais, S.T. and O'Brien, G.W. (1994). *Using linear algebra for intelligent information retrieval*. CS-94-270. Knoxville: Computer Science Department, University of Tennessee.

[4] Chalmers, M. and Chitson, P. (1992). Bead: Explorations in Information Visualization. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM Press, 330-337.

[5] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *JASIS*, 41(6):391-407.

[6] Definition of library classification. At: http://en.wikipedia.org/wiki/Library_classification

[7] Definition of classification in digital libraries. At:
http://www.cs.cornell.edu/wya/DigLib/MS1999/glossary.html

[8] Definition of information gain.
At: http://en.wikipedia.org/wiki/Information_gain_in_decision_trees

[9] Dominich, S. (1994). Interaction Information Retrieval. *Journal of Documentation*, 50(3):197-212.

[10] Fayyad, U. (2001). The digital physics of data mining. *Communications of the ACM*, 44(3):62-65.

[11] Fruchterman, T.M.J. and Reingold, E.M. (1991). Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(1):1129-1164.

[12] Golub, G.H. and Van Loan, C.F. (1996). *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press.

[13] Góth, J. (2006). *Effective methods in the practice of information retrieval*. PhD dissertation. Veszprém: Department of Computer Science, Pannon University.

[14] Hendley, R.J., Drew, N.S., Wood, A.M. and Beale, R. (1995). Case study. Narcissus:

Visualising Information. *Proceedings of the IEEE International Conference on Information Visualization*, 90-96.

[15] Hjörland, B. (2003). Fundamentals of knowledge organization. In Frías, J. A. and Travieso, C. (Eds.): *Trends in knowledge organization research.* Salamanca: Ediciones Universidad de Salamanca, 83-116.

[16] Horn, D. and Gottlieb, A. (2001). The method of quantum clustering. In Dietterich, T.G., Becker, S. and Ghahramani, Z. (Eds.): *Advances in Neural Information Processing Systems 14* [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]. Cambridge, MA.: MIT Press, 769-776.

[17] Horn, D. and Gottlieb, A. (2001). Algorithm for Data Clustering in Pattern Recognition Problems Based on Quantum Mechanics. *Physical Review Letters* 88(1):018702.

[18] Joachims, T. (1997). *Text Categorization with Support Vector Machines: Learning with Many Relevant Features.* LS8-Report 23. Dortmund: Universität Dortmund.

[19] Kamada, T. and Kawai, S. (1988). *Automatic display of network structures for human understanding.* Technical report 88-077. Tokyo: Department of Information Science, University of Tokyo.

[20] Kohonen, T. (2001). *Self-organizing maps.* New York: Springer.

[21] Korfhage, R.R., Day, W.H.E., Beck, L.L. and Appelbe, W.F. (1978). Data physics - an unorthodox view of data and its implications in data processors. *Proceedings of the fourth workshop on Computer architecture for non-numeric processing.* New York: ACM Press, 1-7.

[22] Landauer, T.K., Laham, D. and Derr, M. (2004). From paragraph to graph: Latent semantic analysis for information visualization. *PNAS* [April 6] 101(1):5214-5219.
At: http://www.pnas.org/cgi/doi/10.1073/pnas.0400341101

[23] Losee, R.M. (1988). Parameter estimation for probabilistic document-retrieval models. *JASIS* 39(1):8-16.

[24] Mitchell, T.M. (1997). *Machine learning.* New York : McGraw-Hill.

[25] Rocchio, J.J. (1971). Relevance feedback in information retrieval. In Salton, G. (Ed.): *The SMART retrieval system - experiments in automatic document processing.* Englewood Cliffs: Prentice Hall.

[26] Salton, G. (1966). Automatic Phrase Matching. In Hayes, D.G. (Ed.): *Readings in Automatic Language Processing.* New York: American Elsevier Publishing Company, Inc., 169-188.

[27] Salton, G., Wong, A. and Yang, C.S. (1975). A vector space model for automatic indexing. *Communications of the ACM* 18(11):613-620.

[28] Sebastiani, F. (2005). Text categorization. In Alessandro Zanasi (Ed.): *Text Mining and its Applications.* Southampton: WIT Press, 109-129.
At: http://www.math.unipd.it/ fabseb60/Publications/Publications.html

[29] van Rijsbergen, K. (1979). *Information Retrieval*. London: Butterworths.
Available at: http://www.dcs.gla.ac.uk/Keith/Preface.html

[30] van Rijsbergen, K. (2004). *The geometry of information retrieval*. Cambridge: Cambridge University Press.

[31] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer.

[32] Vapnik, V. (1982). *Estimation of Dependencies Based on Empirical Data*. New York: Springer.

[33] Walshaw, C. (2001). A multilevel algorithm for force-directed graph drawing. In Marks, J. (Ed.): *Graph drawing. 8th International Symposium, Colonial Williamsburg 2000*. Berlin: Springer, 171-182.

[34] Wise, J.A. (1999). The Ecological Approach to Text Visualization. *JASIST* 50(13):1224-1233.

[35] Yi, K. (2006). Challenges in automated classification using library classification schemes. *World Library and Information Congress: 72nd IFLA General Conference and Council, 20-24 August 2006, Seoul, Korea*. At: www.ifla.org/IV/ifla72/papers/097-Yi-en.pdf

Appendix: The derivation of quantum clustering

The time-independent Schrödinger equation can be expressed

$$\left(-\frac{\sigma^2}{2}\nabla^2 + V(\mathbf{x})\right)\psi = E\psi \tag{1}$$

By rearranging the terms in (1) we get

$$V(\mathbf{x}) = E + \frac{\frac{\sigma^2}{2}\nabla^2\psi}{\psi} \tag{2}$$

The ∇^2 operator is called the *Laplacian* operator and is defined as the sum of all second-order partial derivatives. Our task is therefore to find these derivatives for ψ. We begin with the following definitions:

$$\begin{aligned}
\psi(\mathbf{x}) &= \sum_j e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} \\
f(\mathbf{x}) &= e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} \\
h(\mathbf{x}) &= -||\mathbf{x}-\mathbf{a}||^2/2\sigma^2 \\
&= -\sum_i (x_i - a_i)^2/2\sigma^2 \\
&= -\sum_i (x_i^2 - 2x_i a_i + a_i^2)/2\sigma^2
\end{aligned}$$

We now search for the second-order partial derivatives of h

$$\begin{aligned}
\frac{\partial}{\partial x_i}h &= -(2x_i - 2a_i)/2\sigma^2 \\
&= -(x_i - a_i)/\sigma^2 \\
\frac{\partial^2}{\partial x_i^2}h &= -\frac{1}{\sigma^2}
\end{aligned}$$

and continue with the second-order partial derivatives of f

$$\begin{aligned}
\frac{\partial}{\partial x_i}f &= -\frac{x_i - a_i}{\sigma^2}e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} \\
\frac{\partial^2}{\partial x_i^2}f &= -\frac{1}{\sigma^2}e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} + \frac{(x_i - a_i)^2}{\sigma^4}e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2}
\end{aligned}$$

We go on to derive an expression for $\frac{\sigma^2}{2}\nabla^2 f$. In correspondence with [19] we let d denote the number of dimensions in our feature space.

$$\begin{aligned}
\frac{\sigma^2}{2}\frac{\partial^2}{\partial x_i^2}f &= -\frac{1}{2}e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} + \frac{1}{2\sigma^2}(x_i - a_i)^2 e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} \\
\frac{\sigma^2}{2}\nabla^2 f &= -\frac{d}{2}e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} + \sum_i \frac{1}{2\sigma^2}(x_i - a_i)^2 e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} \\
&= -\frac{d}{2}e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} + \frac{1}{2\sigma^2}||\mathbf{x}-\mathbf{r}_j||^2 e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2}
\end{aligned}$$

We continue with the corresponding expression for $\frac{\sigma^2}{2}\nabla^2\psi$

$$\begin{aligned}\frac{\sigma^2}{2}\nabla^2\psi &= -\frac{d}{2}\sum_j e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} + \frac{1}{2\sigma^2}\sum_j ||\mathbf{x}-\mathbf{r}_j||^2 e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} \\ &= -\frac{d}{2}\psi + \frac{1}{2\sigma^2}\sum_j ||\mathbf{x}-\mathbf{r}_j||^2 e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2}\end{aligned}$$

and finally insert the missing components of equation (2)

$$\begin{aligned}\frac{\frac{\sigma^2}{2}\nabla^2\psi}{\psi} &= -\frac{d}{2} + \frac{1}{2\sigma^2\psi}\sum_j ||\mathbf{x}-\mathbf{r}_j||^2 e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2} \\ V(\mathbf{x}) &= E - \frac{d}{2} + \frac{1}{2\sigma^2\psi}\sum_j ||\mathbf{x}-\mathbf{r}_j||^2 e^{-||\mathbf{x}-\mathbf{r}_j||^2/2\sigma^2}\end{aligned}$$

which is the expression for V given in [16].