Martin-Luther-University
Halle-Wittenberg
Institute for Informatics
Databases and Information Systems

LWA 2007
Lernen – Wissen – Adaption

Halle, September 2007

Workshop Proceedings

Alexander Hinneburg

(Editor)

Martin-Luther-University Halle-Wittenberg
September 2007, Germany
ISBN: 978-3-86010-907-6

We thank our sponsors:

Gesellschaft für Informatik e.V.
`http://www.gi-ev.de`

Martin-Luther-Universität Halle-Wittenberg
`http://www.uni-halle.de`

Rapid-I
`http://www.rapid-i.com`

Preliminary

The joint workshop event ”Lernen, Wissen und Adaptivität” (Learning, Knowledge, and Adaptivity) is held from Sep. 24-26, 2007 in Halle, Germany. Like in the years before, the LWA hosts several interesting workshops organized by different special interest groups of the German Society of Informatics (GI). They provide insights to current trends, technologies, applications, and, most important, a forum where established and beginning researchers can share their ideas. The workshops of this years LWA are organized by the following special interest groups

- FG-ABIS (Adaptability and User Modelling in Interactive Software Systems)
- FG-IR (Information Retrieval)
- FG-WM (Knowledge Management)
- FG-KDML (Knowledge Discovery, Data Mining and Machine Learning)
- FG-BIOINF (Bioinformatics)

We are happy to have the invited talks covering the areas of bioinformatics, knowledge discovery and user modelling.

- Dr. Alexander Schliep, MPI for Molecular Genetics, Berlin, Detecting functional modules from heterogenous mass data
- Prof. Dr. Myra Spiliopoulou, Otto-von-Guericke-University Magdeburg, Understanding Change before Adaptation: Tracing Changes over a Stream of Data
- Dr. Michael Fahrmair, DoCoMo Communications Laboratories Europe GmbH, Munich, Germany, Exploiting User Context Models in Large Scale Mobile Context Adaptive Systems

Halle (Saale), September 2007

Alexander Hinneburg

Contents

FGWM 2007 **273**

ABIS 2007 **342**

Detecting functional modules from heterogenous mass data

Alexander Schliep
Max Planck Inst. for Molecular Genetics
Ihnestrasse 73, 14195 Berlin, Germany
schliep@molgen.mpg.de

Abstract

Due to advances in experimental techniques we have seen a vast increase in exciting new data reflecting the spatio-temporal aspects of life on the molecular level. It augments static information about DNA sequences, transcripts and proteins and their interactions. The integration of these heterogeneous sources of data in order to arrive at conclusive information about a biological process is typically performed manually.

We revisit classical mixture models, or convex combinations of density functions, and show how they can effectively model high-dimensional data by use of sufficiently constrained component models, for example for gene expression time-courses. One approach to combining abundant primary data with labels from possibly sparse secondary data is semi-supervised learning. We will use the detection of groups of syn-expressed genes from in-situ images and gene expression time-courses during embryogenesis using a semi-supervised approach as a case study.

Understanding Change before Adaptation: Tracing Changes over a Stream of Data

Myra Spiliopoulou
Otto-von-Guericke University Magdeburg
Germany

Abstract

In recent years, it has been recognized that the models discovered in many real applications are affected by changes in the underlying population. This includes changes in customer preferences, in network intrusion patterns, in the formulation of spam mails, in the topics being emphasized in the news. There is much research devoted in *adapting* models and patterns to the current state of the changing population. However, tracing and understanding the changes themselves is an issue of no less importance: It delivers valuable insights on the trends of the population and contributes in well-informed strategic decisions. We will discuss the issue of pattern change detection for evolving data, including stream data, with special emphasis on cluster evolution.

Pattern change detection can be studied from different perspectives: In the simplest case, it involves comparing a pair of patterns drawn from distinct population samples and testing whether the two samples refer to the same population. These samples may be collected at different time intervals, as is e.g. the case for sales data for two consecutive months. They may also differ with respect to some property of the population, such as the purchase location, the type of payment or the gender of the customers. Research methods that compare patterns for the detection of differences include [6; 2; 5; 12].

Pattern change detection can also be observed as a spatiotemporal phenomenon. This perspective is adopted among else in [1; 9]. In [1], a cluster is defined as a spatial object - a densification of the topological space. This can be best perceived in the context of geographic information systems, where a group of physically proximal trees forms a wood (cluster) - an area that is more densely filled with trees than e.g. a plain or a field. Cluster change detection corresponds then to the discovery of changes caused e.g. by urban development, wood fires and deforestation. Not surprisingly, this perspective can be found also by incremental density-based clustering methods, such as [4; 3].

Ultimately, pattern change detection is a temporal phenomenon, which is not limited to spatial data. For example, Mei and Zhai investigate topic evolution in [11], while Hinneburg et al study linguistic change [8]. The temporal approach involves observing the data as a constant stream or as a sequence of batches collected and analyzed at predefined intervals. This approach is studied among else in [7; 11; 12] and in [10], where the emphasis is on pattern management.

In the talk, the different perspectives to pattern change detection will be discussed in more detail, together with a selection of the above methods.

References

[1] C. Aggarwal. On Change Diagnosis in Evolving Data Streams. *Knowledge and Data Engineering, IEEE Transactions on*, 17(5):587–600, 2005.

[2] I. Bartolini, P. Ciaccia, I. Ntoutsi, M.o Patella, and Y.s Theodoridis. A unified and flexible framework for comparing simple and complex patterns. In *PKDD*, pages 496–499, 2004.

[3] F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. *Proc. of the SIAM Conf. on Data Mining (SDM)*, 2006.

[4] M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental clustering for mining in a data warehousing environment. In *VLDB*, pages 323–333, 1998.

[5] W. Fan. Systematic data selection to mine concept-drifting data streams. In *KDD*, pages 128–137, 2004.

[6] V. Ganti, J. Gehrke, and R. Ramakrishnan. A framework for measuring changes in data characteristics. In *PODS*, pages 126–137, 1999.

[7] V. Ganti, J. Gehrke, and R. Ramakrishnan. Demon: Mining and monitoring evolving data. In *ICDE*, pages 439–448, 2000.

[8] A. Hinneburg, H. Mannila, S. Kaislaniemi, T. Nevalainen, and H. Raumolin-Brunberg. How to Handle Small Samples: Bootstrap and Bayesian Methods in the Analysis of Linguistic Change. *Lit Linguist Computing*, 22(2):137–150, 2007.

[9] P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatio-temporal data. In *SSTD*, pages 364–381, 2005.

[10] A. Maddalena and B. Catania. Towards an interoperable solution for pattern management. In *3rd Int. Workshop on Database Interoperability INTERDB07 (in conjunction with VLDB07)*, 2007.

[11] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD*, pages 198–207, 2005.

[12] M. Spiliopoulou, I. Ntoutsi, Y. Theodoridis, and R. Schult. Monic: modeling and monitoring cluster transitions. In *KDD*, pages 706–711, 2006.

Exploiting User Context Models in Large Scale Mobile Context Adaptive Systems

Michael Fahrmair
DoCoMo Communications Laboratories Europe GmbH
Landsbergerstrasse 312, 80687 Munich, Germany
fahrmair@docomolab-euro.com

Abstract

Developing context adaptive systems for mobile phones has several specific constraints. This is especially true if the system spans a large number of mobile phones. The presentation shows the lessons learned from a 2-year case study and describes a special discovery technology called scoping that is based on user context models and was developed for large scale global object finding application as well as development and evaluation methods that have been used. Finally the presentation summarizes future challenges when applying scoping technology not only specifically for finding lost objects but in a generic way to discover arbitrary context information on a large global scale in near real-time.

Workshop on Knowledge Discovery, Data Mining, and Machine Learning and Workshop on Bioinformatics

Alexander Hinneburg
Martin-Luther-University Halle-Wittenberg

Ralf Klinkenberg and **Ingo Mierswa**
University of Dortmund

Stefan Posch
Martin-Luther-University Halle-Wittenberg

Steffen Neumann
Institute for Plant Biochemistry Halle

KDML 2007

The workshop Knowledge Discovery, Data Mining, and Machine Learning (KDML) in 2007 is organized by the special interest group of the GI on Knowledge Discovery, Data Mining and Maschine Learning (FG-KDML, former FGML). The goal of the workshop is to provide a forum for database and machine learning oriented researchers with interests in knowledge discovery and data mining.

The workshop is part of the workshop week Learning – Knowledge Discovery – Adaptivity (LWA 2007). This provides the chance to meet researchers from the special interest groups on Adaptivity and Interaction, on Information Retrieval, on Knowledge Management, and on Bioinformatics. The program consists of nine regular paper, nine short paper and five poster paper. The program committee had the following members

- Ralf Klinkenberg, University of Dortmund
- Ingo Mierswa, University of Dortmund
- Alexander Hinneburg, Martin-Luther University Halle-Wittenberg
- Martin Atzmüller, University of Würzburg
- Timm Euler, University of Dortmund / viadee Unternehmensberatung GmbH
- Felix Jungermann, University of Dortmund
- Regis Newo, University of Hildesheim
- Michael Wurst, University of Dortmund

September 2007,
Ralf Klinkenberg, Ingo Mierswa, Alexander Hinneburg

BIOINF 2007

The workshop covers applications of machine learning to problems in bioinformatics. Techniques of interest include, but are not limited to, clustering of high-dimensional data, text-mining in biomedical literature, reasoning in large ontologies, data integration, statistical modelling. Applications of interest are whole genome analysis, comparative genomics, gene regulation, protein-protein interaction, signal transduction, proteomics, expression analysis, metabolic networks.

Out of seven submissions four papers have been accepted, which will be presented in a joint session with the KDML 2007 workshop. We thank the following colleagues who supported us as members of the program committee

- U. Bohnebeck (ttz Bremerhaven)
- I. Grosse (Institute for Plant Genetics and Crop Plant Research Gatersleben)
- A. Hinneburg (Martin-Luther-University Halle-Wittenberg)
- O. Kohlbacher (Eberhard Karls Universität Tübingen)
- B. Morgenstern (Georg-August-Universität Göttingen)
- G. Rätsch (Friedrich Miescher Laboratory of the Max Planck Society, Tübingen)
- A. Schliep (MPI for Molecular Genetics Berlin)
- F. Schreiber (Institute for Plant Genetics and Crop Plant Research Gatersleben, Martin-Luther-University Halle-Wittenberg)

September 2007,
Stefan Posch, Steffen Neumann

Tag Recommendations in Folksonomies*

Robert Jäschke[1]**, Leandro Marinho**[2]**, Andreas Hotho**[1]**, Lars Schmidt-Thieme**[2] **and Gerd Stumme**[1]
1: Knowledge & Data Engineering Group (KDE), University of Kassel,
Wilhelmshöher Allee 73, 34121 Kassel, Germany
http://www.kde.cs.uni-kassel.de
2: Information Systems and Machine Learning Lab (ISMLL), University of Hildesheim,
Samelsonplatz 1, 31141 Hildesheim, Germany
http://www.ismll.uni-hildesheim.de

Abstract

Collaborative tagging systems allow users to assign keywords—so called "tags"—to resources. Tags are used for navigation, finding resources and serendipitous browsing and thus provide an immediate benefit for users. These systems usually include tag recommendation mechanisms easing the process of finding good tags for a resource, but also consolidating the tag vocabulary across users. In practice, however, only very basic recommendation strategies are applied.

In this paper we present two tag recommendation algorithms: an adaptation of user-based collaborative filtering and a graph-based recommender built on top of FolkRank, an adaptation of the well-known PageRank algorithm that can cope with undirected triadic hyperedges. We evaluate and compare both algorithms on large-scale real life datasets and show that both provide better results than non-personalized baseline methods. Especially the graph-based recommender outperforms existing methods considerably.

1 Introduction

Social resource sharing systems are web-based systems that allow users to upload their resources, and to label them with arbitrary words, so-called *tags*. The systems can be distinguished according to what kind of resources are supported. Flickr, for instance, allows the sharing of photos, del.icio.us the sharing of bookmarks, CiteULike[1] and Connotea[2] the sharing of bibliographic references, and Last.fm[3] the sharing of music listening habits. Our own system, *BibSonomy*,[4] allows to share bookmarks and BibTeX based publication entries simultaneously.

In their core, these systems are all very similar. Once a user is logged in, he can add a resource to the system, and assign arbitrary tags to it. The collection of all his assignments is his *personomy*, the collection of all personomies constitutes the *folksonomy*. The user can explore his personomy, as well as the personomies of the other users, in all dimensions: for a given user one can see all resources he has uploaded, together with the tags he has assigned to them; when clicking on a resource one sees which other users have uploaded this resource and how they tagged it; and when clicking on a tag one sees who assigned it to which resources. Based on the tags that are assigned to a resource, users are able to search and find her own or other users resources within such systems.

To support users in the tagging process and to expose different facets of a resource, most of the systems offered some kind of tag recommendations already at an early stage. Del.icio.us, for instance, had a tag recommender in June 2005 at the latest,[5] and also included resource recommendations.[6] However, no algorithmic details were published. We assume that these recommendations basically rely on tag–tag-co-occurrences. As of today, nobody has empirically shown the quantitative benefits of recommender systems in such systems. In this paper, we will quantitatively evaluate a tag recommender based on collaborative filtering (introduced in Sec. 3) and a graph based recommender using our ranking algorithm FolkRank (see Sec. 4) on three real world folksonomy datasets. We make the BibSonomy dataset publicly available for research purposes to stimulate research in the area of folksonomy systems (details in Section 5).

The results we are able to present in Sec. 6 are very encouraging as the graph based approach outperforms all other approaches significantly. As we will see later, this is caused by the ability of FolkRank to exploit the information that is pertinent to the specific user together with input from other users via the integrating structure of the underlying hypergraph.

2 Recommending Tags—Problem Definition and State of the Art

Recommending tags can serve various purposes, such as: increasing the chances of getting a resource annotated, reminding a user what a resource is about and consolidating the vocabulary across the users. In this section we formalize the notion of folksonomies, formulate the tag recommendation problem and briefly describe the state of the art on tag recommendations in folksonomies.

A Formal Model for Folksonomies.

A folksonomy describes the users, resources, and tags, and the user-based assignment of tags to resources. Formally, a *folksonomy* is a tuple $\mathbb{F} := (U, T, R, Y)$ where U, T, and R are finite sets, whose elements are called *users*, *tags* and *resources*, resp., and Y is a ternary relation between them, i. e., $Y \subseteq U \times T \times R$, whose elements are called tag as-

[*] A shortened version of this work has been published at PKDD 2007. [1] http://www.citeulike.org [2] http://www.connotea.org [3] http://www.last.fm [4] http://www.bibsonomy.org

[5] http://www.socio-kybernetics.net/saurierduval/archive/2005_06_01_archive.html [6] http://blog.del.icio.us/blog/2005/08/people_who_like.html

signments (*tas* for short).[7] Users are typically described by their user ID, and tags may be arbitrary strings. What is considered a resource depends on the type of system. For instance, in del.icio.us, the resources are URLs, in BibSonomy URLs or publication references, and in last.fm, the resources are artists.

In this paper, we will use an equivalent view on the folksonomy structure. We will consider it as a tripartite (undirected) hypergraph $G = (V, E)$, where $V = U \dot{\cup} T \dot{\cup} R$ is the set of nodes, and $E = \{\{u, t, r\} \mid (u, t, r) \in Y\}$ is the set of hyperedges.

For convenience we also define, for all $u \in U$ and $r \in R$, $\mathrm{tags}(u, r) := \{t \in T \mid (u, t, r) \in Y\}$, i. e., $\mathrm{tags}(u, r)$ is the set of all tags that user u has assigned to resource r. The set of all *posts* of the folksonomy is then $P := \{(u, S, r) \mid u \in U, r \in R, S = \mathrm{tags}(u, r)\}$. Thus, each *post* consists of a user, a resource and all tags that this user has assigned to that resource.

Tag Recommender Systems.

Recommender systems (RS) in general recommend interesting or personalized information objects to users based on explicit or implicit ratings. Usually RS predict ratings of objects or suggest a list of new objects that the user hopefully will like the most. In tag recommender systems the recommendations are, for a given user $u \in U$ and a given resource $r \in R$, a set $\tilde{T}(u, r) \subseteq T$ of tags. In many cases, $\tilde{T}(u, r)$ is computed by first generating a ranking on the set of tags according to some quality or relevance criterion, from which then the top n elements are selected.

Related work.

General overviews on the rather young area of folksonomy systems and their strengths and weaknesses are given in [Hammond *et al.*, 2005; Lund *et al.*, 2005; Mathes, 2004]. In [Mika, 2005], Mika defines a model of semantic-social networks for extracting lightweight ontologies from del.icio.us. Recently, work on more specialized topics such as structure mining on folksonomies—e. g. to visualize trends [Dubinko *et al.*, 2006] and patterns [Schmitz *et al.*, 2006] in users' tagging behavior—as well as ranking of folksonomy contents [Hotho *et al.*, 2006a], analyzing the semiotic dynamics of the tagging vocabulary [Cattuto *et al.*, 2006], or the dynamics and semantics [Halpin *et al.*, 2006] have been presented.

The literature concerning the problem of tag recommendations in folksonomies is still sparse. The existent approaches usually lay in the collaborative filtering and information retrieval areas. AutoTag [Mishne, 2006], e.g., is a tool that suggests tags for weblog posts using information retrieval techniques. Xu et al. [Xu *et al.*, 2006] introduce a collaborative tag suggestion approach based on the HITS algorithm [Kleinberg, 1999]. A goodness measure for tags, derived from collective user authorities, is iteratively adjusted by a reward-penalty algorithm. Benz et al. [Benz *et al.*, 2006] introduce a collaborative approach for bookmark classification based on a combination of nearest-neighbor-classifiers. There, a keyword recommender plays the role of a collaborative tag recommender, but it is just a component of the overall algorithm, and therefore there is no information about its effectiveness alone. The standard tag recommenders, in practice, are services that provide the most-popular tags used for a particular resource. This is usually done by means of tag clouds where the most frequent used tags are depicted in a larger font or otherwise emphasized.

The approaches described above address important aspects of the problem, but they still diverge on the notion of tag relevance and evaluation protocol used. Xu et al. [Xu *et al.*, 2006], e.g., present no quantitative evaluation, while in [Mishne, 2006], the notion of tag relevance in not entirely defined by the users but partially by experts.

3 Collaborative Filtering

Due to its simplicity and promising results, collaborative filtering (CF) has been one of the most dominant methods used in recommender systems. In the next section we recall the basic principles and then present the details of the adaptation to folksonomies.

Basic CF principle.

The idea is to suggest new objects or to predict the utility of a certain object based on the opinion of like-minded users [Sarwar *et al.*, 2001]. In CF, for m users and n objects, the user profiles are represented in a user-object matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. The matrix can be decomposed into row vectors:

$$\mathbf{X} := [\vec{x}_1, ..., \vec{x}_m]^\top \text{ with } \vec{x}_u := [x_{u,1}, ..., x_{u,n}], \text{ for } u := 1, \ldots, m,$$

where $x_{u,o}$ indicates that user u rated object o by $x_{u,o} \in \mathbb{R}$. Each row vector $\vec{x}_u$ corresponds thus to a user profile representing the object ratings of a particular user. This decomposition leads to user-based CF. (The matrix can alternatively be represented by its column vectors leading to item-based recommendation algorithms.)

Now, one can compute, for a given user u, the recommendation as follows. First, based on matrix $\mathbf{X}$ and for a given k, the set N_u^k of the k users that are most similar to user $u \in U$ are computed: $N_u^k := \arg\max^k_{v \in U} \mathrm{sim}(\vec{x}_u, \vec{x}_v)$ where the superscript in the $\arg\max$ function indicates the number k of neighbors to be returned, and sim is regarded (in our setting) as the cosine similarity measure. Then, for a given $n \in \mathbb{N}$, the top n recommendations consist of a list of objects ranked by decreasing frequency of occurrence in the ratings of the neighbors (see Eq. 1 below for the folksonomy case).

This brief discussion refers only to the user-based CF case, moreover, we consider only the recommendation task since in collaborative tagging systems there are usually no ratings and therefore no prediction. For a detailed description about the item-based CF algorithm see [Deshpande and Karypis, 2004].

CF for Tag Recommendations in Folksonomies.

Because of the ternary relational nature of folksonomies, traditional CF cannot be applied directly, unless we reduce the ternary relation Y to a lower dimensional space. To this end we consider as matrix $\mathbf{X}$ alternatively the two 2-dimensional projections $\pi_{UR} Y \in \{0, 1\}^{|U| \times |R|}$ with $(\pi_{UR} Y)_{u,r} := 1$ if there exists $t \in T$ s. t. $(u, t, r) \in Y$ and 0 else and $\pi_{UT} Y \in \{0, 1\}^{|U| \times |T|}$ with $(\pi_{UT} Y)_{u,t} := 1$ if there exists $r \in R$ s. t. $(u, t, r) \in Y$ and 0 else. The projections preserve the user information, and lead to log-based like recommender systems based on occurrence or non-occurrence of resources or tags, resp., with the users. Notice that now we have two possible setups in which the k-neighborhood N_u^k of a user u can be formed, by considering either the resources or the tags as objects.

[7] In the original definition [Hotho *et al.*, 2006a], we introduced additionally a subtag/supertag relation, which we omit here.

Having defined matrix $\mathbf{X}$, and having decided whether to use $\pi_{UR}Y$ or $\pi_{UT}Y$ for computing user neighborhoods, we have the required setup to apply collaborative filtering. For determining, for a given user u, a given resource r, and some $n \in \mathbb{N}$, the set $\tilde{T}(u,r)$ of n recommended tags, we compute first N_u^k as described above, followed by:

$$\tilde{T}(u,r) := \underset{t \in T}{\operatorname{arg} \overset{n}{\max}} \sum_{v \in N_u^k} \operatorname{sim}(\vec{x}_u, \vec{x}_v)\delta(v,t,r) \qquad (1)$$

where $\delta(v,t,r) := 1$ if $(v,t,r) \in Y$ and 0 else.

4 A Graph Based approach

The seminal PageRank algorithm [Brin and Page, 1998] reflects the idea that a web page is important if there are many pages linking to it, and if those pages are important themselves.[8] In [Hotho *et al.*, 2006a], we employed the same underlying principle for Google-like search and ranking in folksonomies. The key idea of our FolkRank algorithm is that a resource which is tagged with important tags by important users becomes important itself. The same holds, symmetrically, for tags and users, thus we have a graph of vertices which are mutually reinforcing each other by spreading their weights. In this section we briefly recall the principles of the FolkRank algorithm, and explain how we use it for generating tag recommendations. More details can be found in [Hotho *et al.*, 2006a].

Because of the different nature of folksonomies compared to the web graph (undirected triadic hyperedges instead of directed binary edges), PageRank cannot be applied directly on folksonomies. In order to employ a weight-spreading ranking scheme on folksonomies, we will overcome this problem in two steps. First, we transform the hypergraph into an undirected graph. Then we apply a differential ranking approach that deals with the skewed structure of the network and the undirectedness of folksonomies, and which allows for topic-specific rankings.

Folksonomy-Adapted Pagerank.
First we convert the folksonomy $\mathbb{F} = (U, T, R, Y)$ into an *un*directed tri-partite graph $G_{\mathbb{F}} = (V, E)$. The set V of nodes of the graph consists of the disjoint union of the sets of tags, users and resources. All co-occurrences of tags and users, users and resources, tags and resources become edges between the respective nodes (more details in [Hotho *et al.*, 2006a]).

The rank of the vertices of the graph are the entries in the fixed point $\vec{w}$ of the weight spreading computation

$$\vec{w} \leftarrow dA\vec{w} + (1-d)\vec{p} \ , \qquad (2)$$

where $\vec{w}$ is a weight vector with one entry for each node, A is the row-stochastic version of the adjacency matrix of the graph $G_{\mathbb{F}}$ defined above, $\vec{p}$ is the preference vector, and $d \in [0,1]$ is determining the influence of $\vec{p}$.

For a global ranking, one will choose $\vec{p} = \mathbf{1}$, i. e., the vector composed by 1's. In order to generate recommendations, however, $\vec{p}$ can be tuned by giving a higher weight to the user and to the resource for which one currently wants to generate a recommendation. The recommendation $\tilde{T}(u,r)$ is then the set of the top n nodes in the ranking, restricted to tags. In the experiments presented below, we will see that this version performs reasonable, but not exceptional. This is in line with our observation in [Hotho *et al.*, 2006a] which showed that the topic-specific rankings are biased by the global graph structure. As a consequence, we developed the following differential approach.

FolkRank—Topic-Specific Ranking.
As the graph $G_{\mathbb{F}}$ that we created in the previous step is undirected, we face the problem that an application of the original PageRank would result in weights that flow in one direction of an edge and then 'swash back' along the same edge in the next iteration, so that one would basically rank the nodes in the folksonomy by their degree distribution. This makes it very difficult for other nodes than those with high edge degree to become highly ranked, no matter what the preference vector is.

This problem is solved by the *differential* approach in FolkRank, which computes a topic-specific ranking of the elements in a folksonomy. Let $\vec{w}_0$ be the fixed point from Equation (2) without preference vector and $\vec{w}_1$ be the fixed point with preference vector $\vec{p}$ and in this case $d = 0.7$. Then $\vec{w} := \vec{w}_1 - \vec{w}_0$ is the final weight vector. Thus, we compute the winners and losers of the mutual reinforcement of nodes when a user/resource pair is given, compared to the baseline without a preference vector. We call the resulting weight $\vec{w}[x]$ of an element x of the folksonomy the *FolkRank* of x.[9] For generating a tag recommendation for a given user/resource pair, we compute the ranking as described above, and then restrict the result set $\tilde{T}(u,r)$ to the top n tag nodes.

5 Evaluation

In this section we first describe the datasets we used, how we prepared the data, the methodology deployed to measure the performance, and which algorithms we used, together with their specific settings.

Datasets.
To evaluate the proposed recommendation techniques we have chosen datasets from three different folksonomy systems: *del.icio.us*, *Last.fm* and *BibSonomy*. They have different sizes, different resources to annotate and are probably used by different people. Therefore they form a good basis to test our tag recommendation scenario in a general setting. Table 1 gives an overview on the datasets. For all datasets we disregarded if the tags had lower or upper case since this is the behaviour of most systems when querying them for posts tagged with a certain tag (although often they store the tags as entered by the user).

Del.icio.us. One of the first and most popular folksonomy systems is del.icio.us [10] which exists since the end of 2003. It allows users to tag bookmarks (URLs) and had according to its blog around 1.5 Mio. users in February 2007. We used a dataset from del.icio.us we obtained from July 27 to 30, 2005 [Hotho *et al.*, 2006a]. Since del.icio.us allows its users to *not* tag resources at all (they can be accessed by

[8] This idea was extended in a similar fashion to bipartite subgraphs of the web in HITS [Kleinberg, 1999] and to n-ary directed graphs in [Xi *et al.*, 2004].

[9] In [Hotho *et al.*, 2006a] we showed that $\vec{w}$ provides indeed valuable results on a large-scale real-world dataset while $\vec{w}_1$ provides an unstructured mix of topic-relevant elements with elements having high edge degree. In [Hotho *et al.*, 2006b], we applied this approach for detecting trends over time in folksonomies.

[10] http://del.icio.us

the tag "system:unfiled") we added those posts with the tag "system:unfiled" to the dataset.

Last.fm. Audioscrobbler[11] is a music engine based on a collection of music profiles. These profiles are built through the use of the company's flagship product, Last.fm,[12] a system that provides personalized radio stations for its users and updates their profiles using the music they listen to. Audioscrobbler exposes large portions of data through their web services API. The data was gathered during July 2006, partly through the web services API (collecting user nicknames), partly crawling the Last.fm site. Here the resources are artist names, which are already normalized by the system.

BibSonomy. This system allows users to manage and annotate bookmarks and publication references simultanously. Since three of the authors have participated in the development of BibSonomy, [13] we were able to create a complete snapshot of all users, resources (both publication references and bookmarks) and tags publicly available at April 30, 2007, 23:59:59 CEST.[14] From the snapshot we excluded the posts from the DBLP computer science bibliography[15] since they are automatically inserted and all owned by one user and all tagged with the same tag (*dblp*). Therefore they do not provide meaningful information for the analysis.

Core computation.

Many recommendation algorithms suffer from sparse data or the "long tail" of items which were used by only few users. Hence, to increase the chances of good results for all algorithms (with exception of the most popular tags recommender) we will restrict the evaluation to the "dense" part of the folksonomy, for which we adapt the notion of a p-core [Batagelj and Zaversnik, 2002] to tri-partite hypergraphs. The p-core of level k has the property, that each user, tag and resource has/occurs in at least k posts.

To construct the p-core, recall that a folksonomy (U, T, R, Y) can be formalized equivalently as tri-partite hypergraph $G = (V, E)$ with $V = U \dot\cup T \dot\cup R$. First we define, for a subset $\tilde V$ of V (with $\tilde V = \tilde U \dot\cup \tilde T \dot\cup \tilde R$ and $\tilde U \subseteq U, \tilde T \subseteq T, \tilde R \subseteq R$), the function

$$\operatorname{posts}(v, \tilde V) = \begin{cases} \{(v, S, r) \mid r \in \tilde R, S = \operatorname{tags}_{\tilde V}(v, r)\} & \text{if } v \in \tilde U \\ \{(u, v, r) \mid u \in \tilde U, r \in \tilde R\} & \text{if } v \in \tilde T \\ \{(u, S, v) \mid u \in \tilde U, S = \operatorname{tags}_{\tilde V}(u, v)\} & \text{if } v \in \tilde R \end{cases} \tag{3}$$

which assigns to each $v \in \tilde V$ the set of all posts in which v occurs. Here, $\operatorname{tags}_{\tilde V}$ is defined as in Section 2, but restricted to the subgraph $(\tilde V, E_{|\tilde V})$. Let $p(v, \tilde V) := |\operatorname{posts}(v, \tilde V)|$. The p-core at level $k \in \mathbb{N}$ is then the subgraph of (V, E) induced by $\tilde V$, where $\tilde V$ is a maximal subset of V such that, for all $v \in \tilde V$, $p(v, \tilde V) \geq k$ holds.

Since $p(v, \tilde V)$ is, for all v, a monotone function in $\tilde V$, the p-core at any level k is unique [Batagelj and Zaversnik, 2002], and we can use the algorithm presented in [Batagelj and Zaversnik, 2002] for its computation. An overview on the p-cores we used for our datasets is given in Table 2. For BibSonomy, we used $k = 5$ instead of 10 because of its smaller size. The largest k for which a p-core exists is listed, for each dataset, in the last column of Table 1.

Evaluation methodology.

To evaluate the recommenders we used a variant of the leave-one-out hold-out estimation [Herlocker *et al.*, 2004] which we call *LeavePostOut*. In all datasets, we picked, for each user, one of his posts p randomly. The task of the different recommenders was then to predict the tags of this post, based on the folksonomy $\mathbb{F} \setminus \{p\}$.

As performance measures we use precision and recall which are standard in such scenarios [Herlocker *et al.*, 2004]. With r being the resource from the randomly picked post of user u and $\tilde T(u, r)$ the set of recommended tags, recall and precision are defined as

$$\operatorname{recall}(\tilde T(u, r)) = \frac{1}{|U|} \sum_{u \in U} \frac{|\operatorname{tags}(u, r) \cap \tilde T(u, r)|}{|\operatorname{tags}(u, r)|} \tag{4}$$

$$\operatorname{precision}(\tilde T(u, r)) = \frac{1}{|U|} \sum_{u \in U} \frac{|\operatorname{tags}(u, r) \cap \tilde T(u, r)|}{|\tilde T(u, r)|}. \tag{5}$$

For each of the algorithms of our evaluation we will now describe briefly the specific settings used to run them.

Most popular tags. For each tag we counted in how many posts it occurs and used the top tags (ranked by occurence count) as recommendations.

Most popular tags by resource. For a given resource we counted for all tags in how many posts they occur together with that resource. We then used the tags that occured most often together with that resource as recommendation.

Adapted PageRank. With the parameter $d = 0.7$ we stopped computation after 10 iterations or when the distance between two consecutive weight vectors was less than 10^{-6}. In $\vec p$, we gave higher weights to the user and the resource from the post which was chosen. While each user, tag and resource got a preference weight of 1, the user and resource from that particular post got a preference weight of $1 + |U|$ and $1 + |R|$, resp.

FolkRank. The same parameter and preference weights were used as in the adapted PageRank.

Collaborative Filtering UT. Collaborative filtering algorithm where the neighborhood is computed based on the user-tag matrix $\pi_{UT} Y$. The only parameter to be tuned in the CF based algorithms is the number k of best neighbors. For that, multiple runs where performed where k was successively incremented until a point where no more improvements in the results were observed. For this approach the best values for k were 80 for the deli.icio.us, 60 for the Last.fm, and 20 for the BibSonomy dataset.

[11] http://www.audioscrobbler.net
[12] http://www.last.fm
[13] http://www.bibsonomy.org
[14] On request to bibsonomy@cs.uni-kassel.de a snapshot of BibSonomy is available for research purposes.
[15] http://www.informatik.uni-trier.de/~ley/db/

Table 1: Characteristics of the used datasets.

dataset	$\|U\|$	$\|T\|$	$\|R\|$	$\|Y\|$	$\|P\|$	date	$k_{\max}$
del.icio.us	75,245	456,697	3,158,435	17,780,260	7,698,653	2005-07-30	77
Last.fm	3,746	10,848	5,197	299,520	100,101	2006-07-01	20
BibSonomy	1,037	28,648	86,563	341,183	96,972	2007-04-30	7

Table 2: Characteristics of the p-cores at level k.

dataset	k	$\|U\|$	$\|T\|$	$\|R\|$	$\|Y\|$	$\|P\|$
del.icio.us	10	37,399	22,170	74,874	7,487,319	3,055,436
Last.fm	10	2,917	2,045	1,853	219,702	75,565
BibSonomy	5	116	412	361	10,148	2,522

Collaborative Filtering UR. Collaborative Filtering algorithm where the neighborhood is computed based on the user-resource matrix $\pi_{UR}Y$. For this approach the best values for k were 100 for the deli.icio.us, 100 for the Last.fm, and 30 for the BibSonomy dataset.

6 Results

In this section we present and describe the results of the evaluation. We will see that all three datasets show the same overall behavior: 'most popular tags' is outperformed by all other approaches; the CF-UT algorithm performs slightly better than and the CF-UR approach approx. as good as the 'most popular tag by resource', and FolkRank uniformly provides significantly better results.

We will now study the results in detail. There are two types of diagrams. The first type of diagram (Figure 1) shows in a straightforward manner how the recall depends on the number of recommended tags. In the other diagrams with usual precision-recall plots (Figures 2 and 3) a datapoint on a curve stands for the number of tags used for recommendation (starting with the highest ranked tag on the left of the curve and ending with ten tags on the right). Hence, the steady decay of all curves in those plots means that the more tags of the recommendation are regarded, the better the recall and the worse the precision will be.

Del.icio.us. Figure 1 shows how the recall increases, when more tags of the recommendation are used. All algorithms perform significantly better than the baseline based on the most popular tags—whereas it is much harder to beat the resource specific most popular tags. The surprising result is that the graph based recommendations of FolkRank have superior recall—independent of the number of regarded tags. The second best results come from the collaborative filtering approach based on user tag similiarities. For ten recommended tags it reaches 89% of the recall of FolkRank (0.71 of 0.80)—a significant difference. The idea to suggest the top most popular tags of the resource gives a recall which is very similiar to using the CF recommender based on users resource similiarities—both perform worse than the aforementioned approaches. Between most popular tags by resource and most popular tags is the adapted PageRank which is influenced by the most popular tags, as discussed earlier.

The precision-recall plot in Figure 2 again reveals clearly the quality of the recommendations given by FolkRank compared to the other approaches. The top 10 tags given by FolkRank contained in average 80 % of the tags the users decided to attach to the selected resource. Nevertheless, the precision is rather poor with values below 0.2. So why do we call this a good result anyhow?

A post in del.icio.us contains only 2.45 tags in average. A precision of 100 % can therefore not be reached when recommending ten tags. However, from a subjective point of view, the additional 'wrong' tags may even be considered as highly relevant, as the following example shows, where the user *tnash* has tagged the page http://www.ariadne.ac.uk/issue43/chudnov/ with the tags *semantic, web,* and *webdesign.* Since that page discusses the interaction of publication reference management systems in the web by the OpenURL standard, the tags recommended by FolkRank (*openurl, web, webdesign, libraries, search, semantic, metadata, social-software, sfx, seo*) are adequate and capture not only the user's point of view that this is a webdesign related issue in the semantic web, but also provide him with more specific tags like *libraries* or *metadata* which the users nevertheless did not use. The CF based on user/tag similiarities recommends very similiar tags (*openurl, libraries, social-software, sfx, metadata, me/toread, software, myndsi, work, 2read*). The additional tags may thus animate users to use more tags and/or tags from a different viewpoint for describing resources, and thus lead to converging vocabularies.

The essential point in this example is, however, that FolkRank is able to predict—additionally to globally relevant tags—the exact tags of the user which CF could not. This is due to the fact that FolkRank considers, via the hypergraph structure, also the vocabulary of the user himself, which CF by definition doesn't do.

Last.fm. For this dataset, recall and precision for FolkRank are considerably higher than for the del.icio.us dataset, see Table 3. Even when just two tags are recommended, the recall is close to 60 %. Though the precision of the user-resource collaborative filtering approach is always slightly better than on the del.icio.us dataset, the recall is only better until the 7th tag where it falls below the recall reached on the del.icio.us dataset. Again, the graph based approach outperforms all other methods (CF-UT reaches at most 76 % of the recall of FolkRank). An interesting observation can be made about the adapted PageRank: its recall now is the second best after FolkRank for larger numbers of recommended tags. This shows the overall importance of general terms in this dataset—which have a high influence on the adapted PageRank (cf. Section 4).

BibSonomy. For the BibSonomy dataset the precision for FolkRank is similiar to the Last.fm dataset (see Table 3), but the recall (omitted here because of space restrictions)

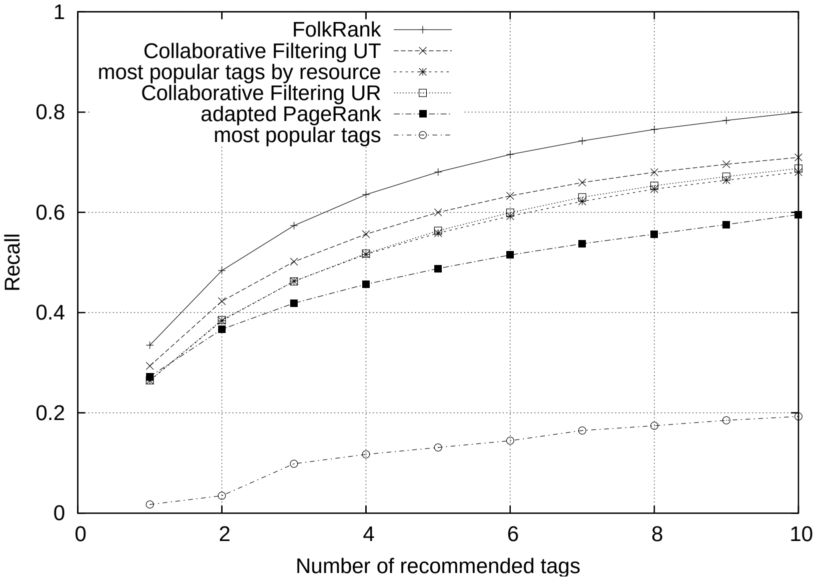


Figure 1: Recall for del.icio.us p-core at level 10

Table 3: Precision for BibSonomy p-core at level 5

Number of recommended tags	1	2	3	4	5	6	7	8	9	10
FolkRank	0.724	0.586	0.474	0.412	0.364	0.319	0.289	0.263	0.243	0.225
Collaborative Filtering UT	0.569	0.483	0.411	0.343	0.311	0.276	0.265	0.257	0.243	0.235
most popular tags by resource	0.534	0.440	0.382	0.350	0.311	0.288	0.267	0.250	0.241	0.234
Collaborative Filtering UR	0.509	0.478	0.408	0.341	0.311	0.285	0.267	0.252	0.241	0.234

reaches only values comparable to the del.icio.us dataset. We will focus here on a phenomenon which is unique for that dataset. With an increasing number of suggested tags, the precision decrease is steeper for FolkRank than for the collaborative filtering and the 'most popular tags by resource' algorithm such that the latter two approaches for ten suggested tags finally overtake FolkRank. The reason is that the average number of tags in a post is around 4 for this dataset and while FolkRank can always recommend the maximum number of tags, for the other approaches there are often not enough tags available for recommendation. This is because in the p-core of order 5, for each post, often tags from only four other posts can be used for recommendation with these approaches. Consequently this behaviour is even more noticeable in the p-core of order 3 (which is not shown here).

7 Conclusion

In this paper we presented two methods for tag recommendations in folksonomies, a straightforward collaborative filtering adaptation based on projections and an adaptation of the well-known PageRank algorithm named FolkRank. We conducted experiments in three real-life datasets and showed that FolkRank outperforms the other methods. Some conclusions of our experiment were:

- The exploitation of the hypergraph structure in FolkRank yields a significant advantage.
- Despite its simplicity and non-personalized aspect, the 'most popular tags' achieved reasonable precision and recall on the small datasets (Last.fm and BibSonomy) what indicates its adequacy for the cold start problem.
- The adapted PageRank profits also from this good performance of the 'most popular tags' on small datasets.

Currently, our approach for FolkRank always returns a fixed number of tags, often yielding low precision. Future work will include a method to determine a good cut-off point automatically.

Acknowledgement. Part of this research was funded by the European Commission in the Nepomuk[16] (FP6-027705) and the X-Media[17] (IST-FP6-026978) projects.

References

[Batagelj and Zaversnik, 2002] V. Batagelj and M. Zaversnik. Generalized cores, 2002. cs.DS/0202039, http://arxiv.org/abs/cs/0202039.

[Benz *et al.*, 2006] D. Benz, K. Tso, and L. Schmidt-Thieme. Automatic bookmark classification: A collaborative approach. In *Proceedings of the Second Workshop on Innovations in Web Infrastructure (IWI 2006)*, Edinburgh, Scotland, 2006.

[Brin and Page, 1998] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, April 1998.

[Cattuto *et al.*, 2006] Ciro Cattuto, Vittorio Loreto, and Luciano Pietronero. Collaborative tagging and semiotic dynamics, May 2006. http://arxiv.org/abs/cs/0605015.

[16] http://nepomuk.semanticdesktop.org

[17] http://www.x-media-project.org,

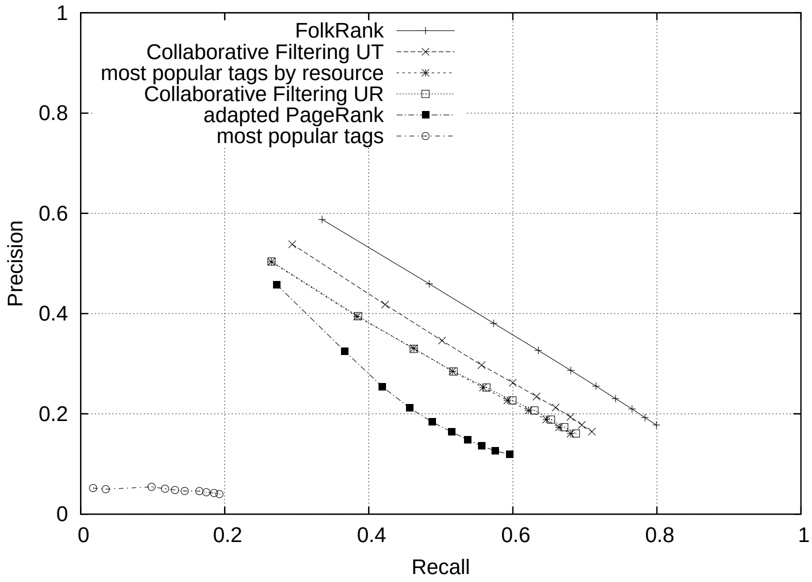


Figure 2: Recall and Precision for del.icio.us p-core at level 10

[Deshpande and Karypis, 2004] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[Dubinko *et al.*, 2006] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. In *Proc. of the 15th International WWW Conference*, Edinburgh, Scotland, 2006.

[Halpin *et al.*, 2006] H. Halpin, V. Robu, and H. Shepard. The dynamics and semantics of collaborative tagging. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW'06)*, 2006.

[Hammond *et al.*, 2005] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine*, 11(4), April 2005.

[Herlocker *et al.*, 2004] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[Hotho *et al.*, 2006a] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426, Heidelberg, June 2006. Springer.

[Hotho *et al.*, 2006b] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Trend detection in folksonomies. In Yannis S. Avrithis, Yiannis Kompatsiaris, Steffen Staab, and Noel E. O'Connor, editors, *Proc. First International Conference on Semantics And Digital Media Technology (SAMT)*, volume 4306 of *LNCS*, pages 56–70, Heidelberg, Dec 2006. Springer.

[Kleinberg, 1999] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[Lund *et al.*, 2005] Ben Lund, Tony Hammond, Martin Flack, and Timo Hannay. Social Bookmarking Tools (II): A Case Study - Connotea. *D-Lib Magazine*, 11(4), April 2005.

[Mathes, 2004] Adam Mathes. Folksonomies – Cooperative Classification and Communication Through Shared Metadata, December 2004. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.

[Mika, 2005] Peter Mika. Ontologies Are Us: A Unified Model of Social Networks and Semantics. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *ISWC 2005*, volume 3729 of *LNCS*, pages 522–536, Berlin Heidelberg, November 2005. Springer-Verlag.

[Mishne, 2006] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press.

[Sarwar *et al.*, 2001] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *World Wide Web*, pages 285–295, 2001.

[Schmitz *et al.*, 2006] Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Žiberna, editors, *Data Science and Classification: Proc. of the 10th IFCS Conf.*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Berlin, Heidelberg, 2006. Springer.

[Xi *et al.*, 2004] W. Xi, B. Zhang, Y. Lu, Z. Chen, S. Yan, H. Zeng, W. Ma, and E. Fox. Link fusion: A unified link analysis framework for multi-type interrelated data objects. In *Proc. 13th International World Wide Web Conference*, New York, 2004.

[Xu *et al.*, 2006] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the Collaborative*

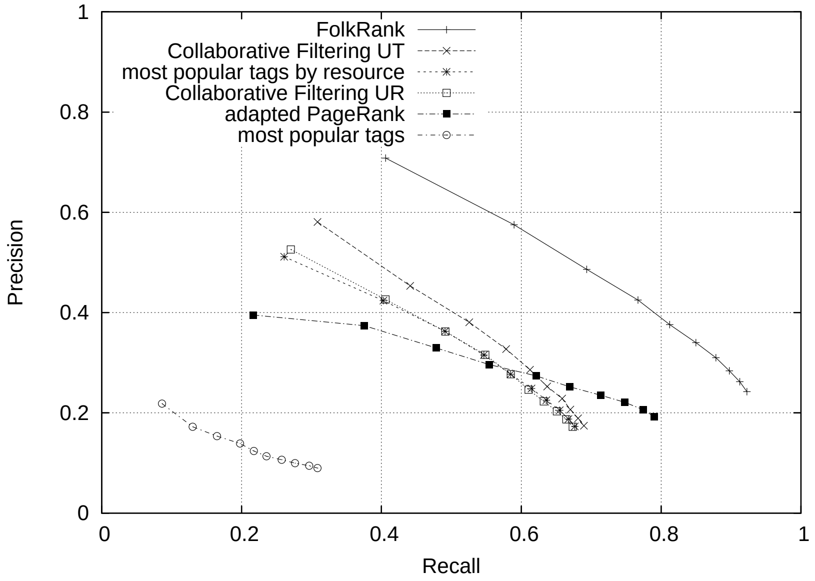


Figure 3: Recall and Precision for Last.fm p-core at level 10

Web Tagging Workshop at the WWW 2006, Edinburgh, Scotland, 2006.

Enhancing a Web Recommender System Based on Q-Learning

Nima Taghipour, Ahmad Kardan
Department of Computer Engineering
Amirkabir University of Technology
15875-4413, Tehran, Iran
{n-taghipour,aakardan}@aut.ac.ir

Abstract

The problem of information overload on the Internet has received much attention in the recent years. In this regard Recommender Systems were introduced that aim at directing users toward the items that best meet their needs and interests. Web Recommendation has been an active application area in Web Mining and Machine Learning research. In this paper we utilize various approaches to enhance a reinforcement learning framework, previously devised by us for web recommendations based on web usage data. We specifically aim at addressing the problems of sparsity and the low coverage achieved in specific situations as the main weaknesses of the system. Our efforts to improve the system fall into two general categories: improvements made from a machine learning perspective to the problem and the efforts made to improve the quality of recommendations. Regarding the first category, we make use of an alternative reward function based on our definition of states and actions. In the second category, we exploit the information available in web content to enhance our model of the problem and devise two methods for enriching it with semantic knowledge. Each solution imposes its specific advantages and limitations. A detailed analysis of these factors is presented at the end. We evaluate our method under different settings and show how these methods can improve the quality of web recommendations.

1 Introduction

The amount of information available on-line is increasing rapidly with the explosive growth of the World Wide Web and the advent of e-Commerce. Although this surely provides users with more options, at the same time makes it more difficult to find the “right” or “interesting” information from this great pool of information, the problem commonly known as information overload. To address these problems, recommender systems have been introduced [Resnick and Varian, 1997]. They can be defined as the personalized information technology used to predict a user evaluation of a particular item [Deshpande and Karypis, 2004] or more generally as any system that guides users toward interesting or useful objects in a large space of possible options [Burke, 2002].

Recommender systems have been used in various applications ranging from predicting the products a customer is likely to buy [Shany et al. 2005], movies, music or news that might interest the user [Konstan et al. 1998; Zhang and Seo, 2001] and web pages that the user is likely to seek[Cooley *et al.*, 1999; Fu et al., 2000; Joachims et al. 1997; Mobasher et al. 2000a], which is also the focus of this paper. Web page recommendation is considered a user modeling or web personalization task. One research area that has recently contributed greatly to this problem is web mining. Most of the systems developed in this field are based on web usage mining [Srivastava et al, 2000] which is the process of applying data mining techniques to the discovery of usage patterns form web data. These systems are mainly concerned with analyzing web usage logs, discovering patterns from this data and making recommendations based on the extracted knowledge [Fu et al., 2000; Mobasher et al. 2000a; Shahabi et al., 1997; Zhang and Seo, 2001]. One important characteristic of these systems is that unlike traditional recommender systems, which mainly base their decisions on user ratings on different items or other explicit feedbacks provided by the user [Deshpande and Karypis, 2004; Herlocker et al., 2000] these techniques discover user preferences from their implicit feedbacks, namely the web pages they have visited. More recently, systems that take advantage of a combination of content, usage and even structure information of the web have been introduced [Li and Zaiane, 2004; Mobasher et al. 2000b; Nakagawa and Mobasher , 2003] and shown superior results in the web page recommendation problem.

We have previously devised a machine learning perspective toward the problem based on Reinforcement Learning (RL) [Sutton, 1998] which we showed was suitable to the nature of web page recommendation problem and had some intrinsic advantages over previous methods [Taghipour et. al, 2007]. Our previous system made recommendations primarily based on web usage logs. We have modeled the recommendation process as a Q-Learning problem. For this purpose we devised state and action definitions and rewarding policies, considering common concepts and techniques used in the web usage mining domain. The choice of reinforcement learning was due to several reasons: It seems appropriate for the nature of web page recommendation problem. Due to the characteristics of this type of learning and the fact that we are not making decisions explicitly from the patterns discovered from the data, it provides us with a system which is constantly in the learning process. Does not need periodic updates; can easily adapt itself to changes in website structure and content and new trends in user behavior. Our evaluation showed that the system performance was superior to methods based on association rules and collaborative filtering.

In this paper we extend and enhance our previous work. We conduct an analysis of the system's behavior and aim at addressing the problems of sparsity of the state space and the low coverage achieved in specific situations as the main weaknesses of the system. Our efforts to improve the system fall into two general categories: improvements made from a machine learning perspective to the problem and the efforts made to improve the quality of recommendations. Regarding the first category, we make use of an alternative reward function based on our definition of states and actions. We show how this alternative method can reduce the sparsity of the states and what trade offs it imposes on the overall system performance. In the second category, we exploit the information available in web content to enhance our model of the problem and devise two methods for enriching it with semantic knowledge. Each solution imposes its specific advantages and limitations. A detailed analysis of these factors is presented at the end. We evaluate our method under different settings and show how these methods can improve the quality of web recommendations

The organization of the paper is as follows: in section 2 we overview the related work done in recommender systems, focusing more on recent systems and on the application of reinforcement learning in these systems. We overview our previous work in section 3. We represent the issues we tend to improve in section 4 and present our solution in sections 5 and 6. The discussions and conclusion of the paper comes in section 7 along with some recommendations for future work.

2 Related Work

Recommender systems have been developed using various approaches and can be categorized in various ways [Burke, 2002]. Collaborative techniques [Herlocker et al., 2000] are the most successful and the most widely used techniques employed in these systems [Deshpande and Karypis, 2004; Konstanet al. 1998; Wasfi, 1999]. Recently, Web mining and especially web usage mining techniques have been used widely in web recommender systems [Cooley *et al.*, 1999; Fu et al., 2000; Mobasher et al. 2000a; Mobasher et al. 2000b]. The common approach in these systems is to extract navigational patterns from usage data by data mining techniques such as association rules and clustering, and making recommendations based on them. These approaches differ fundamentally from our method in which no static pattern is extracted from data.

RL has been previously used for recommendations in several applications. WebWatcher [Joachims et al. 1997], exploits Q-Learning to guide users to their desired pages. Pages correspond to states and hyperlinks to actions, rewards are computed based on the similarity of the page content and user profile keywords. In most other systems reinforcement learning is used to reflect user feedback and update current state of recommendations. A general framework is presented in [Golovin and Rahm, 2004], which consists of a database of recommendations generated by various models and a learning module that updates the weight of each recommendation by user feedback. In [Srivihok and Sukonmanee, 2005] a travel recommendation agent is introduced which considers various attributes for trips and customers, computes each trip's value with a linear function and updates function coefficients after receiving each user feedback. RL is used for information filtering in [Zhang and Seo, 2001] which maintains a profile for each user containing keywords of interests and updates each word's weight according to the implicit and explicit feedbacks received from the user. In [Shany et al. 2005] the recommendation problem is modeled as a Markov Decision Process (MDP). The system's states correspond to user's previous purchases, rewards are based on the profit achieved by selling the items and the recommendations are made using the theory of MDP and their novel state-transition function. In a more recent work [Mahmood and Ricci 2007] RL is used in the context of a conversational travel recommender system in order to learn optimal interaction strategies. They model the problem with a finite state-space based on variables like the interaction stage, user action and the result size of a query. The set of actions represent what the system chooses to perform in each state e.g. executing a query, suggesting modification. Finally RL is used to learn an optimal strategy, based on a user behavior model.

3 Web Page Recommendation and Reinforcement Learning

3.1 Problem Definition

The specific problem which our system is supposed to solve can be summarized as follows: the system has, as input data, the log file of users' past visits to the website. A user enters our website and begins requesting web pages. Considering the pages this user has requested so far the system has to predict in what other pages the user is probably interested and recommend them to him/her. Table 1 illustrates a sample scenario. Predictions are considered successful if the user chooses to visit those pages in the remaining of that session.

Table 1: A sample session and system recommendations

Visited Page	a	b	c	d	e	f
Navigation Trail	*a*	*ab*	*abc*	*abcd*	*abcde*	*abcdef*
System Prediction	{c,g}	{d,m}	{e,d}	{s,r}	{f,b}	{h}

3.2 Recommendations as a Q-Learning problem

Reinforcement learning [Sutton, 1998] is primarily known in machine learning research as a framework in which agents learn to choose the optimal action in each situation or *state* they are in. The goal of the agent is to learn which actions to perform in each state to receive the greatest accumulative reward, in its path to the goal state.

Using the Analogy of a Game

In order to better represent our approach toward the problem we try to use the notion of a game. In a typical sce-

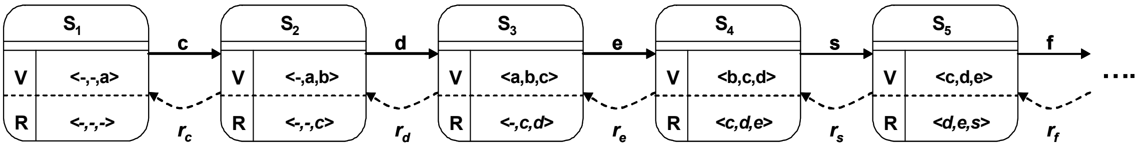


Figure 1. States and actions in the recommendation problem

nario a web user visits pages sequentially from a web site, let's say the sequence a user *u* requested is composed of pages a, b, c and d. Each page the user requests can be considered a step or move in our game. The system's purpose is to predict user's next move(s) with the knowledge of his previous moves. Whenever the user makes a move (requests a page), if the system has previously predicted the move, it will receive positive points and otherwise it will receive none or negative points. For example predicting a visit of page *d* after viewing pages *a* and *b* by the user in the above example yields in positive points for the system.

Important issues can be inferred from this simple analogy. We can see the problem certainly has a stochastic nature; Actions will consist of prediction or recommendation of web pages by the system in each state; Each state should at least show the history of pages visited by the user so far and that an action should be rewarded positively if it recommends a page that will be visited in one of the consequent states.

Modeling States and Actions

We tend to keep our states as simple as possible, at least in order to keep their number manageable. Regarding the states, we can see keeping only the user trail can be insufficient. With that definition it won't be possible to reflect the effect of an action *a* performed in state s_i, in any consequent state S_{i+n} where $i > 1$. Another issue we should take into account is the number of possible states: if we allow the states to contain any given sequence of page visits clearly we'll be potentially faced by an infinite number of states. What we chose to do was to limit the page visit sequences to a constant number. For this purpose we adopted the notion of N-Grams which is commonly applied in similar personalization systems based on web usage mining [Mobasher et al. 2000a; Mobasher et al. 2000b]. In this model we put a sliding window of size *w* on user's page visits, resulting in states containing only the last *w* pages requested by the user.

Regarding the actions, we chose simplicity. Our action consists of a single page recommendation in each state. Considering multiple page recommendations might have shown us the effect of the combination of recommended pages on the user, in the expense of making our state space and rewarding policy much more complicated. The corresponding states and actions of the user session of Table 1is presented in Figure 1.

Choosing a Reward Function

If we consider each state s consists of two sequences V, R indicating the sequence of visited and previously recommended pages respectively:

$$V_s = < p_1^v, p_2^v, \ldots, p_w^v > \qquad (1)$$
$$R_s = < p_1^R, p_2^R, \ldots, p_n^R >$$

Where p_i^v and p_i^R indicates the *i*th visited and recommended page in the state. Reward for each action would be a function of $V_{s'}$ and $R_{s'}$ where S' is our next state. One tricky issue worth considering is that though tempting, we should not base on rewards on $| V_{s'} \cap R_{s'} |$ since it will cause extra credit for a single correct move. We simply consider only the occurrence of the last page visited in the recommended pages list in state S' to reward the action performed in the previous sate *s.* To complete our rewarding procedure we take into account common metrics used in web page recommender systems. One issue is considering when the page was predicted by the system and when the user actually visited the page. Another factor commonly considered in theses systems [Mobasher et al. 2000a; Liu et al., 2004; Fu et al., 2000] is the time the user spends on a page, assuming the more time the user spends on a page the more interested he probably has been in that page. The rewarding can be summarized as follows:

```
- Assume $\delta(s,a) = s'$
- $P_R = V_{s',w} \cap R_{s'}$
- If p≠Ø
      For each page p in $P_R$
      r(s,a) += reward(Dist(R_{s'},p),Time(p_w^v))
```

Where $Dist(R_{s'}, p)$ is the distance of page p from the end of the recommended pages list and the time user has spent on the last page of the state is $Time(p^v{}_w)$.

Having put all the pieces of the model together, we can see why reinforcement learning might be a good candidate for the recommendation problem: it does not rely on any previous assumptions regarding the probability distribution of visiting a page after having visited a sequence of pages, which makes it general enough for diverse usage patterns. The nature of the problem matches perfectly with the notion of temporal difference, as performing an action/recommendation might not seem valuable to us in the immediate next state and sequence of actions might have led to a successful recommendation for which we must credit rewards. Also with reinforcement learning the system is intrinsically learning even when performing in real world, as the recommendations are the actions the system performs.

Training the system

The training process can be summarized as the following:

```
- Initial values of Q(s,a) for each pair s,a are set to zero
- Repeat until convergence
      o A random episode is chosen from the set of training episodes.
      o s is set to the first step/state of the episode.
      o For each step of the episode do
            ▪ Chose an action a of this state using the $\varepsilon$ – greedy policy.
            ▪ Perform action a, observe the next state and compute r(s,a) as described before.
            ▪ Update value of Q(s,a)
            ▪ $s \leftarrow s'$.
```

The last modification we experimented was changing our reward function. We noticed as we put a sliding window on our sequence of previously recommended pages, practically we had limited the effect of each action to w' next states. After training the system using this definition, the system was mostly successful in recommending pages visited around w' steps ahead. Although this might be quite acceptable while choosing an appropriate value for w', it tends to limit system's prediction ability as large numbers of w' make our state space enormous. To overcome this problem we devised a rather simple modification in our reward function: what we needed was to reward recommendation of a page if it is likely to be visited an unknown number of states ahead. Fortunately our definition of states and actions gives us just the information we need and ironically this information is stored in Q values of each state. The basic idea is that when an action/recommendation is appropriate in state S_i, indicating the recommended page is likely to occur in the following states, it should also be considered appropriate in state S_{i-1} and the actions in that state that frequently lead to S_i. Following this recursive procedure we can propagate the value of performing a specific action beyond the limits imposed by w'. This change is easily reflected in our learning system by considering value of $Q(s',a)$ in computation of *r(s,a)* with a coefficient like γ. It should be taken into account that the effect of this modification in our reward function must certainly be limited as in its most extreme case where we only take this next Q value into account we're practically encouraging recommendation of pages that tend to occur mostly in the end of user sessions.

Experimental Evaluation Setting

We evaluated the system performance in the different settings described above. We used simulated log files generated by a web traffic simulator [Liu et al., 2004] to tune our rewarding functions. The log files were simulated for a website containing 700 web pages. We pruned user sessions with a length smaller than 5 and were provided with 16000 user sessions with average length of eight. As our evaluation data set we used the web logs of the Depaul University website, made available by the author of [Mobasher et al. 2000b]. This dataset contains 13745 sessions and 687 pages. 70% of the data set was used as the training set and the remaining was used to test the system. For our evaluation we presented each user session to the system, and recorded the recommendations it made after seeing each page the user had visited.

4 Observations on System Performance

In our initial evaluation of the system, we identified two main deficiencies in the system's performance:

The first problem was the rather large number of states, in the case where we used larger window sizes for the visited and recommended pages in the states. This large number of states not only had a negative effect on system learning efficiency, but also caused a sparse state space which in turn contributed negatively to our second problem.

The second problem was the problem of *unknown states*. There were cases where we noticed that the state resulted from the sequence of pages visited by the user had actually never occurred in the training phase. In situations like these the system was unable to make any decisions regarding the pages to recommend to the users. There are two main causes behind this problem: The first and the more obvious one is the sequence of recommended pages included in the states; and the other one is the fact that our system relies solely on usage data and thus is unable to make any generalization.

When we analyzed the system's final states after the training phase, we came across the fact that the large number of states was mostly due to the recommended page sequence. This was anticipated from the beginning, due to the system's tendency to explore during the training phase. The visited page sequence in the user's sessions is restricted by the website structure, the links available on each web page, while there is no restriction on the system's recommended page sequence. In our training data of about 600 pages we actually had 988 different visited page sequences, when using a window of size *k=3*, while the actual number of states was larger by a magnitude of 2. Of course actions with high Q-values where mostly common among the states with the same visited sequence and different recommended sequence.

Regarding the problem of unknown states, it should be mentioned that the problem is only partially due to the sparse state space. Even when ignoring the recommended page sequence in the states, we still came across previously unseen states in the test phase. This is actually a problem common in recommender systems that have usage data as their only source of information. One common solution to this problem is to incorporate some semantic knowledge about the items being recommended into the system. We also decided to do the same and devised two different approaches for this purpose which we describe in detail in section 6.

5 Reducing the Recommended Page Window Size

In order to address the sparsity of the states resulted by the recommended sequence in the states, a simple solution would be to train the system with the method mentioned earlier while keeping a parallel model of states and actions which could be considered an abstraction of the previous model. This second model simply ignored the sequence of recommended pages in the states (had a recommended page window of size zero). The Q values of this abstract model were updated each time a Q value of one of its corresponding states (the ones with the same *V* regardless of the recommended page sequence) in the original model was updated. The update rule was the same as the one used in the original model. Although relatively effective, this solution had its drawbacks, the most obvious of which was the need to keep two parallel models for recommendations. The abstract model would actually be a structure to hold inaccurate Q-values updated from the still sparse and complex model. Besides this solution would lose or weaken the features which motivated using the reinforcement learning framework for the problem.

In order to enhance our method, we took advantage of one of our proposed reward functions. As we mentioned earlier, we could exploit the Q value of performing an action *a* in state S_i to reward the action a in S_{i-1} resulting to S_i. This was due to the fact that this Q value encapsulated the probability of a visit to page *a* later in the user session. The purpose for keeping a sequence of recommended pages was to be able to reward pages recommended some steps ahead and it can be seen that with the choice of reward function, we'll be able to do the same in the reverse manner: When keeping the recommended sequence we'll be able to reward a recommended page *after it has actually been visited*, while in the other case we'll reward a recommendation *in anticipation of its occurrence*, relying on the knowledge gathered in our *Q-Values*. This might seem inaccurate at first, but it actually is in great agreement with the fundamentals of *Q*-learning and the meaning behind *Q-Values* kept in each state. Using this method our reward function will be a simple combination of the rewards gathered by predicting the immediate next page and the reward gained form performing the same action in the next state. We evaluated system performance in these settings using the metrics introduced in [Li and Zaiane, 2004].

Table 2: System performance with the alternative Reward Function (AC=Accuracy, SG=Shortcut Gain, k=Window Size)

Coverage	**Performance**					
	K=0		K=2		K=3	
	AC	SG	AC	SG	AC	SG
.10	.75	.16	.74	.18	.72	.20
.15	.63	.35	.65	.32	.60	.24
.20	.57	.40	.60	.43	.49	.30
.25	.52	.47	.53	.48	.41	.33
.30	.48	.55	.40	.63	.38	.37
.40	.30	.58	.33	.65	.31	.42
.50	.2$	.63	.26	.65	.26	.42

As can be seen in Table 2 this choice of reward function enables the system to reduce its recommended window size to one or zero without losing accuracy except for a 0.11 in the worst case, while reducing the number of states dramatically and eliminating the need to use a parallel model of the recommendations along with the original model. This approach also imposes a trade off on the learning process. In one hand it reduces the state space and on the other hand it requires a longer time to train the system because of the time needed to propagate the Q values. It should be mentioned that this cost is ignorable as it is merely a 3% increase in the training repetitions with much faster training period overall.

6 Incorporating Content Information to Enhance a Usage-Based Model

The second problem we were facing was rather different in nature. This time we were faced with sequences of user page visits which had not happened before in the user sessions. In our setting, this problem was due to the fact that a specific visit sequence had not been present in the training data. Although not the case here, this problem can be also due to the infamous "new item" problem commonly faced in collaborative filtering [Burke, 2002; Mobasher et al. 2000b]. Another issue worth considering is the fact that the mere presence of a state in our state space cannot guarantee a high quality recommendation, to be more accurate it can be said that even a high Q-value cannot guarantee a high quality recommendation by itself. Simply put, when a pattern has few occurrences in the training data it cannot be a strong basis for decision making, a problem addressed in other systems by introducing metrics like support threshold in association rules [Mobasher et al. 2000a]. Similarly in our case a high Q-value, like a high confidence for an association rule, cannot be trusted unless it has strong supporting evidence in the data. We conclude that there are cases when historical usage data provides no evidence or evidence that's not strong enough to make a rational decision about user's need or behavior, in these cases there is the need to use an additional source of information. In the following sections we show how we incorporate web content information into our model to enhance the system.

When faced with an unseen state or a state with few occurrences in the training data, we needed to find the most similar previously seen state(s) and recommend the pages with the highest scores in those states. The problem will be how to find the most similar state(s) to a given state. As a simplified example, consider the case when the user visited page sequence is *<a,b,x>* and this sequence is not present in our states. If we were able to find a page y which could be considered similar to x by some measure and the state *<a,b,y>* was among our states, intuitively we could use the Q values of actions present in the latter state to recommend pages for the former state. To measure similarity between pages we exploited the content information of them. We devised two different approaches for this matter, each with its own motivation.

6.1 Employing Hierarchical Clustering for a New State-Action Definition

The task of finding similarity between the states based on their content information is not a trivial one. Any state contains a number of sequential page visits by a user. The content information of this sequence can be modeled in various ways e.g. k bag of words, a vector aggregating all page contents in a state, etc. The same is true regarding the problem of computing similarity between the states. One successful approach used to enhance web usage mining systems is exploiting content information to transform the raw log files into more meaningful semantic logs [Eirikani et al., 2003] and then applying data mining techniques on them. In a typical scenario pages are mapped to higher level concepts e.g. catalogue page, product page, etc and a user session consisting of sequential pages will be transformed to a sequence of concepts followed by the user.

We decided to exploit the same techniques in our system to improve our state and action model. In order to make our solution both general and applicable, we avoided using an ad-hoc concept hierarchy or a general ontology for this purpose. Instead we chose to exploit hierarchical document clustering which can provide us with semantic relationships between pages without the need of a specifically devised ontology or concept hierarchy and manual assignment of concepts to pages. It should be noted that the output of other more sophisticated approaches like the one proposed in [Eirikani et al., 2004] for generating C-Logs could also be used for this purpose without affecting our general RL model. In order to map pages to higher level concepts, we applied the DC-Tree [Wong and Fu, 2000] clustering algorithm on the web pages. After clustering the web pages, our state and action definition change as follows: each state would consist of a sequence of *k* page clusters visited by the user, the actions will be recommendation of a specific page cluster to the user. In order to do so we just need a module to find the cluster each page belongs to and transform each usage log to a page cluster sequence in the training phase. All the other aspects of the system like the reward function and the learning process would remain the same. This will result in a smaller state-action space as now the state space size is dependant on the number of distinct page clusters instead of the number of distinct pages in the website. The learning process will become more efficient and the system has a more general model of users' browsing behavior on the site. In this setting the chance of an unseen state will be much less and actually minimized as our evaluation results show, since now we're moving in a much denser state space.

In the test phase, the user's raw session will be converted to a semantic session, the corresponding state will be found and the page cluster with the highest value is identified. The next step would be to recommend pages from the chosen cluster(s). We chose to recommend the pages with a probability corresponding to their similarity to the cluster mean vector. This method also has another advantage which is its ability to cover a wider range of pages to be recommended as our evaluations show and also the potential ability of avoiding the "new item" problem as any new page will be categorized in the appropriate cluster and have a fair chance of being recommended.

6.2 Semantic Clustering of the States as an Alternative Method for integrating Content and Usage

In the previous approach we applied generalization on our states and actions by replacing occurrences of web pages by their semantic groupings. Besides its advantages, like any generalization it also has the effect of information loss at a detail level. There's certainly a trade off between the two, but sometimes it might be more useful to know exactly that a sequence *a,b,c* is probably going to be followed by d rather than knowing the general rule that users tend to visit a page related to *Topic(d)* after visiting pages on a sequence of topics. The former can be considered a smaller and more accurate piece of knowledge. As we said there's always a trade off between the two and having information on both levels might come handy in different situations. In an attempt to move a little deeper into the abstract levels of semantics, we devised the following approach based on semantic clustering of states.

In this method, we use the content information residing in the visited page sequence of our states to derive an aggregate content profile from each state and then group our states into semantic state clusters. The first issue we need to consider is how to represent the content information of each state. Regarding the content information of each page, we used the keyword vector extracted from each during the DC-tree clustering method in the previous stage. In order to model the content information of each state we experimented with two different methods: the first method was based on considering each state as k (ordered) bag of words and the other one based on deriving a keyword vector from the weighted sum of all the k pages in the state. Our empirical evaluations, which we do not present here due to space limitations, showed the first method was more appropriate for our purposes. Having chosen a content model for the states, the next step would be to group them into clusters. Similarity between pages was computed by the cosine-based similarity function commonly used in information retrieval literature. The similarity between states was computed based on a weighted average of the similarity of corresponding pages, with regard to their position in the sequence. More weight was assigned to the similarity score of the pages visited later in the user session. In the case of k=3, a ratio of 5:3:2 was used for the similarity scores, ordered reversely by their position in the visit sequence. The recommendation procedure now changes as follows. Whenever the user session results in a state s, if s is an unknown state, the corresponding content model of the state is computed. This model containing k bag of words is then compared to the mean vectors of the state clusters (using the same weighted similarity measure used for the clustering) and the nearest cluster is found. In the next step, the states within the cluster are considered and the actions with higher *Q-Values* are selected as the recommendations. Note that in this last step we tend to choose the states from the cluster which have occurred more frequently in the usage data. In this manner we'll be able to find portions of sessions which are both semantically similar to our session and have a rather stronger usage support and then recommend the actions appropriate for those states. The same can be done in the case where a state resides in our state space but it has been visited less than a given threshold. The only difference is that in this case we know the cluster the state belongs to in advance and there's no need to compute the corresponding content model and find nearest clusters.

This method performs better than the previous one especially with respect to the accuracy or precision of the recommendations. This is due to the more detailed information that the system still keeps in individual states. Of course the number of the states is also larger than the previous one and the process of finding similar states is added, but it's not really affecting the system's efficiency since after the initial clustering of the pages which is also present in the previous method, only one additional clustering is needed which is done offline. The only major drawback is the loss of coverage on the web site's web pages. This is due to the fact that in this model we still model the actions as recommendation of web pages. This is where our tradeoff comes from; we lose coverage on the web pages and in return gain accuracy. There are some solutions to achieve better accuracy which we plan to explore as we discuss in our conclusion.

7 Conclusion and Discussions

In this paper we introduced various ways to enhance a web recommender system based on the reinforcement learning paradigm. The first set of solutions showed how we can utilize the underlying definition of a problem model to simplify a state model and achieve better results both in the sense of the system's efficiency and less directly in its performance from a goal oriented perspective. The second set of improvements was resulted from incorporating another source of information into our model.

Table 3- Comparison of different methods for integrating content into the model

Method	Metrics			
	Coverage	**Hit Ratio**	**Click Reduction**	**Average Hit Rank**
Usage Based	56	**56.06**	10.21	**3.25**
Content Clusters	**97.22**	49.12	9.10	4.31
State Clusters	89.30	53.11	**10.51**	3.39

We integrated our usage based model with content data in two different manners, resulting in different outcomes. The achieved results confirm the flexibility of our proposed RL model, in incorporating different sources of information into the web recommendation problem. In the first method, we changed our state and action definition to include web page categories instead of web pages themselves. This way we achieved a model containing the semantics of usage logs, which also contains a smaller number of states and actions. This setting provided us with the advantage of reducing the sparsity of our state space, which was one of our motivations in the first place. It should be noted that this alternative state definition also introduces the problem of information loss, as now we have a more abstract view of the user behavior.

The second method was based on semantic clustering of the state space. This resulted in groups of page visit sequences which are semantically similar, along with the corresponding actions. Using this alternative representation, we were able to generalize user behavior when needed while keeping the detailed information encapsulated in the state model. This model performs better with regard to precision, for the price of a larger number of states and a weaker coverage. This seems inevitable due to the inherent trade off between utilizing a generalized model and making decisions based on detailed information.

Our second model is a step toward reaching a balance between generalized and detail models of user behavior. We intend to work further on this model and explore different methods for its improvements in our future works. One initial solution is to expand the action definition by methods like utilizing semantically similar actions in the states or devising an alternative reward function which takes into account the content similarity of a recommended page to the content of the sequence of visited pages.

References

[Burke, 2000] Burke, R. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 2002.

[Cooley et al., 1999] Cooley, R., Tan, P. N., Srivastava, J. WebSIFT: The Web Site Information Filter System. In Proceedings of the Web Usage Analysis and User Profiling Workshop (WEBKDD'99), 1999.

[Deshpande and Karypis, 2004] Deshpande, M., Karypis, G. Item-based top-N recommendation algorithms. ACM Transactions on Information Systems (TOIS), 2004.

[Eirikani et al., 2003] Eirinaki, M., Vazirgiannis, M., Varlamis, I. SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process, in Proc. of the 9th SIGKDD Conf. 2003.

[Eirikani et al., 2004] Eirinaki, M., Lampos, C., Paulakis, S., Vazirgiannis, M. Web Personalization Integrating Content Semantics and Navigational Patterns. In Proceedings of the sixth ACM workshop on Web Information and Data Management WIDM 2004.

[Fu et al., 2000] Fu, X., Budzik, J., Hammond, K. J. Mining navigation history for recommendation. Intelligent User Interfaces, 2000.

[Golovin and Rahm, 2004] Golovin, N., Rahm, E. Reinforcement Learning Architecture for Web Recommendations. Proceedings of the ITCC2004. IEEE, 2004.

[Herlocker et al., 2000] Herlocker, J., Konstan, J., Brochers, A., Riedel, J. An Algorithmic Framework for Performing Collaborative Filtering. Proceedings of 2000 Conference on Research and development in Information Retrieval, 2000.

[Joachims et al. 1997] Joachims, T., Freitag, D., Mitchell, T. M. WebWatcher: A tour guide for the world wide web. Proceedings of International Joint Conference on Artificial Intelligence, 1997.

[Konstanet al. 1998] Konstan, J., Riedl, J., Borchers, A., Herlocker, J. Recommender Systems: A GroupLens Perspective. In: Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08), 1998.

[Li and Zaiane, 2004] Li, J., Zaiane, O. R. Combining Usage, Content and Structure Data to Improve Web Site Recommendation, 5th International Conference on Electronic Commerce and Web, 2004

[Liu et al., 2004]Liu, J., Zhang, S., Yang, J. Characterizing Web usage regularities with information foraging agents. IEEE Transactions on Knowledge and Data Engineering, 16(5), 566-584. 2004.

[Mobasher et al. 2000a] Mobasher, B., Cooley, R., Srivastava, J. Automatic Personalization based on Web Usage Mining. Communications of the ACM. 43 (8), pp. 142-151, 2000.

[Mobasher et al. 2000b] Mobasher, B., Dai, H., Luo, T., Sun, Y., Zhu, J. Integrating web usage and content mining for more effective personalization. In EC-Web, pages 165–176, 2000.

[Nakagawa and Mobasher , 2003]Nakagawa M., Mobasher, B. A Hybrid Web Personalization Model Based on Site Connectivity. Proc. 5th WEBKDD workshop, 2003.

[Resnick and Varian, 1997] Resnick, P., Varian, H.R. Recommender Systems. Communications of the ACM, 40 (3), 56-58, 1997.

[Ricci and Mahmood, 2007] Ricci F., Mahmood, T. Learning and adaptivity in interactive recommender systems. In Proceedings of the ICEC'07 Conference, August 2007.

[Shahabi et al., 1997] Shahabi, C., M. Zarkesh, A., Abidi, J., Shah, V. Knowledge discovery from user's Web-page navigation. In Proceedings of the 7th IEEE Intl. Workshop on Research Issues in Data Engineering, 1997.

[Shany et al. 2005] Shany, G., Heckerman, D., Barfman, R. An MDP-Based Recommender System. Journal of Machine Learning Research, 2005.

[Srivastava et al, 2000] Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. SIGKDD Explorations, 1(2):12–23, 2000.

[Srivihok and Sukonmanee, 2005] Srivihok, A., Sukonmanee, V. E-commerce intelligent agent: personalization travel support agent using Q Learning. ACM International Conference Proceeding Series; Proceedings of the 7th international conference on Electronic commerce, 2005

[Su et al. 2000] Su, Z., Yang, Q., Lu, Y., Zhang, H. What next: A prediction System for Web Requests Using N-gram Sequence Models. In Proceedings of the First International Conference on Web Information Systems and Engineering Conference.2000.

[Sutton, 1998] Sutton, R.S., Barto, A.G. Reinforcement Learning: An Introduction, MIT Press, Cambridge, 1998

[Taghipour et. al, 2007] Taghipour, N., Kardan, A., Shiry Ghidary, S. Usage-Based Web Recommendations: A Reinforcement Learning Approach. To Appear in the Proceedings of the ACM Recommender Systems 2007 Conference. 2007.

[Wasfi, 1999] Wasfi, A. M. Collecting User Access Patterns for Building User Profiles and Collaborative Filtering. In: IUI '99: Proceedings of the 1999 International Conference on Intelligent User Interfaces. 1999.

[Wong and Fu, 2000] W. Wong and A. Fu. Incremental document clustering for web page classification, 2000. IEEE 2000 Internatial Confernece on Information Society in the 21st century: emerging technologies and new challenges (IS2000), Nov 5-8, 2000, Japan.

[Zhang and Seo, 2001] Zhang, B., Seo, Y. Personalized web-document filtering using reinforcement learning. Applied Artificial Intelligence, 15(7):665-685, 2001.

Constraint Based Hierarchical Clustering for Text Documents

Korinna Bade and **Andreas Nürnberger**
Otto-von-Guericke-University Magdeburg,
D-39106 Magdeburg, Germany,
{kbade,nuernb}@iws.cs.uni-magdeburg.de

Abstract

This paper deals with clustering under constraints in a hierarchical clustering scenario on text documents. The scenario considers a partially given hierarchy that shall be usefully extended. It is shown how constraints can be utilized in such a scenario. Related work on the topic is discussed showing major issues, especially concerning our setting. We present three approaches to solve the given task and evaluate them under different aspects, most importantly the case of unevenly distributed constraints.

1 Introduction

There has been published lately a lot of work on constraint based clustering, e.g. [Bilenko *et al.*, 2004], [Davidson and Ravi, 2005], [Wagstaff *et al.*, 2001], [Xing *et al.*, 2003]. All this work aims at deriving a single flat cluster partition, even though they might use a hierarchical cluster algorithm. In contrast to them, we are interested in obtaining a hierarchical structure of nested clusters, which poses different requirements on the cluster algorithm. There are many applications, in which a hierarchical cluster structure is more useful than a single flat partition. One such example is the clustering of text documents into a (personal) topic hierarchy. Such topics are naturally structured hierarchically. Furthermore, hierarchies can improve the access to the data for a user, if a large number of specific clusters is present, as the user can locate interesting topics step by step by several specializations.

After defining our hierarchical setting, we analyze how constraints can be used in the scope of hierarchical clustering (Sec. 2). In Sec. 3, we review related work on constraint based clustering in more detail, trying to show different predominant concepts and their problems. We then present two different approaches for hierarchical constraint based clustering of text documents in Sec. 4 as well as their combination. These approaches are then evaluated in Sec. 5 with different hierarchical datasets.

2 Hierarchical constraint based clustering

To avoid confusion with other approaches of constraint based clustering as well as different opinions about the concept of hierarchical clustering, we use this section to define the current problem at hand from our perspective. Furthermore, we clarify the use of constraints in this setting.

2.1 Problem Definition

First, we want to clarify the use of the terms *class* and *cluster*. In our opinion, both terms have very similar meaning, describing a set of objects belonging together. However, *class* is usually used in supervised learning and *cluster* in unsupervised learning. Here, we use the term *class* to refer to the structure in the data that shall be uncovered, while *clusters* are the actual groupings of items derived by the algorithm. However, as a direct mapping from the derived *clusters* to the actual *classes* is sought, both terms might be used interchangeable.

We define our task at hand as a semi-supervised hierarchical learning problem. The goal of the clustering is to uncover a hierarchical class structure that consist of a set of classes C, between which hierarchical relations $R_H = \{(c_1, c_2) \in C \times C | c_1 \geq_H c_2\}$ hold ($c_1 \geq_H c_2$ means that c_1 contains c_2 as a subclass). Thus, the combination between C and R_H represents a hierarchical tree structure of classes. The data objects in O uniquely belong to one class in C. It is important to note that we specifically allow for the assignment of objects to intermediate levels of the hierarchy. This makes the whole problem a true hierarchical problem. Therefore, a recursive application of flat partitioning algorithms is not sufficient.

The cluster algorithm will be constrained in deriving any cluster hierarchy by making a part of the class hierarchy known to it, i.e. $C_k \subseteq C$ and $R_{Hk} \subseteq R_H$. Furthermore, some objects belonging to these classes are given, i.e. $O_k \subseteq O$. Please note that we assume for real world applications that C_k is smaller than C and O_k is smaller than O. Furthermore, we assume that we have at least one given object $o \in O_k$ for each given class $c \in C_k$. By C_k and R_{Hk}, the cluster algorithm is constrained to produce a hierarchical clustering that preserves the existing structure R_{Hk}, while recovering further clusters and extracting their relations to each other and to the classes in C_k, i.e. the constrained algorithm is supposed to refine the given structure by further data (see Fig. 1).

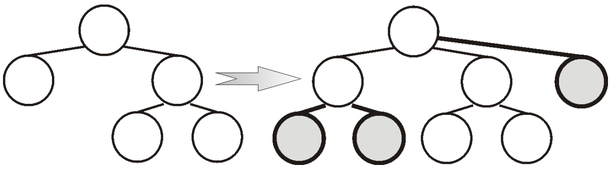


Figure 1: Hierarchy refinement/extension

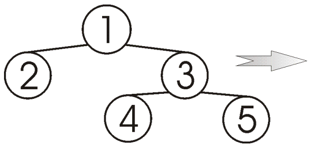
2.2 Constraints

In constraint based clustering, the supervised knowledge is usually expressed by two sets of object pairs, the must-link constraints and the cannot-link constraints [Wagstaff *et al.*, 2001]. The pairs in the must-link-set describe objects that should be clustered in the same cluster, and the cannot-link set contains object pairs that should be separated. However, in hierarchical clustering, each object is actually linked with every other object, although over different hierarchy levels. Therefore, such an absolute constraint definition is not appropriate. The drawbacks of this constraint definition for hierarchical clustering were also mentioned in [Kestler *et al.*, 2006]. The authors work with hierarchical agglomerative clustering and solve the problem by fixing the constraints to a certain dendrogram level. However, in most cases, it is not clear, on which level the constraints should be applied.

We therefore suggest another definition of constraints for hierarchical clustering tasks. What matters here is the order, in which objects are linked. We formalize this as must-link-before (MLB) constraints (see also [Bade and Nürnberger, 2006]). An element of this set is a pair of an object $o \in O_k$ with an ordered list $\mathcal{S}$ of m object sets. Each of these sets contains objects from O_k, to which o should be linked before all objects in the following sets. The order between the sets is important, while inside the sets it does not matter, which object is linked first. Each MLB constraint covers all objects in O_k. Therefore, application of the MLB constraints requires knowledge on all relations between the elements of O_k, which is, e.g., true, in our case of knowing C_k and R_{Hk}. The definition of the MLB constraint is formalized as

$$MLB = \{(o, \mathcal{S})\} = \{(o, (S_1, ..., S_m))\} \tag{1}$$

with $O_k = \bigcup_i S_i \cup o$ and $\forall i, j (i \neq j) : S_i \cap S_j = \emptyset$ and $\forall i : o \notin S_i$. For each $o \in O_k$ such a constraint can be extracted. An example is given in Fig. 2. Here, any object from class 4 should first be linked with all objects in 4. Then, it should be linked to objects from class 3 and 5, whereas the order does not matter. Finally, the object should be linked to all other objects, i.e objects from class 1 and 2. This leads to the MLB constraint shown on the right of Fig. 2.



$$\begin{array}{l}(o^* \in C_4,\\ (\bigcup_{o \in C_4 \setminus \{o^*\}} o,\\ \bigcup_{o \in C_3 \cup C_5} o,\\ \bigcup_{o \in C_1 \cup C_2} o))\end{array}$$

Figure 2: Example of a MLB Constraint Extraction

Furthermore, the MLB constraints can be seen as expressing several relative comparisons between objects. Similar to the formalization in [Schultz and Joachims, 2004], we can extract several comparisons between three objects (o_x, o_y, o_z) in the form: o_x should be linked to o_y before it is linked to o_z. The above formalization of the MLB constraint is merely a compressed view on several such relative comparisons. However, considering the single three object pairs instead also allows for constraining the hierarchical clustering algorithm without a complete knowledge of hierarchical relationship between elements O_k, but rather with a pairwise view as typical for constraint based clustering.

3 Discussion of Related Work

Existing methods in constraint based clustering can be divided in two different types of approaches, i.e. *instance based* and *metric based* approaches. In the *instance based* approaches, the constraints are used to influence the cluster algorithm directly based on the constraint pairs. This is done in several ways, e.g. by using the constraints for initialization (e.g. in [Kim and Lee, 2002]), by enforcing them during the clustering (e.g. in [Wagstaff *et al.*, 2001]), or by integrating them in the cluster objective function (e.g. in [Basu *et al.*, 2004]). The *metric based* approaches map the given constraints to a distance metric, which is then used during the clustering process (e.g. [Bar-Hillel *et al.*, 2003]). The basic idea of most of these approaches is to weight features differently, depending on their importance for the distance computation. While the metric is usually learned in advance using only the given constraints, the approach in [Bilenko *et al.*, 2004] adapts the distance metric during clustering. Such an approach allows for the integration of knowledge from unlabeled objects.

Instance based approaches usually have less impact on the clustering as the constraints influence the algorithm rather locally. In the worst case (and depending on the clustering method used), directly enforcing the constraints might lead to no overall benefit or even decrease the clustering performance, if their merge highly violates the underlying cluster similarity space. On the other hand, learning a distance metric aims on generalizing knowledge about the final clustering from the given constraints. However, this global influence can be problematic under two different aspects. First, different classes might contradict in terms of import features to stress. Features that improve the cluster formation for a certain class might tear apart a different cluster. And second, if the constraints do not cover all interesting clusters, the metric learning might be strongly biased by the known classes. Unfortunately, this is not analyzed in current literature, although it is an important aspect in our opinion. In particular, this means that constraints are not evenly distributed in the filled instance space, but occur locally focused. Current literature evaluations are usually based on using about evenly distributed constraints by drawing them at random from the complete instance space. In this paper, we want to fill this gap by evaluating our approaches with different scenarios of supervision yielding different local distributions of constraints.

From the used clustering algorithms, K-Means is the most predominant, e.g. in [Wagstaff *et al.*, 2001]. Hierarchical agglomerative clustering (HAC) can also be found several times, e.g. in [Davidson and Ravi, 2005]. For metric learning, optimization problems are often formulated and solved, e.g. in [Xing *et al.*, 2003]. In our work presented here, we compare an instance based approach with a metric based approach as well as their combination. Gradient descent is used for metric learning and a HAC algorithm for clustering. This choice was particularly motivated by the fact that the number of clusters is unknown in advance. Furthermore, we are interested in hierarchical cluster structures, which also makes it difficult to compare our results to other approaches, which only search for a flat partitioning of data.

Another important factor is the data to be clustered. In general, all approaches in the literature only require a feature vector as representation of the data. However, if features are weighted for metric learning, special attention should be paid to how these vectors were created in the first place. Here, we focus on text data. Feature vectors of text documents are usually created by assigning each term to a single feature, weighting each feature with the $tf \times idf$ weighting scheme. As the created vectors highly depend on the length of the document, all vectors are usually normalized. However, this has the effect that dimensions are no longer independent from each other as the final value for a feature also depends on the values of the other features. Especially, if "noise features" have largely different values for two documents assumed to be equal (or at least very similar), the values in the important features can vary a lot just due to normalization. This cannot be corrected by any weighting scheme that works on the normalized vectors limiting the possible performance gain. Therefore, our metric based approach presented in the following is especially designed to handle these normalization effects.

4 Our Approaches

As mentioned earlier, we implemented an instance based approach as well as a metric based approach. The instanced based approach is rather simple. Here, the hierarchical agglomerative clustering algorithm is modified in a way that merges occur only in accordance with the given hierarchy. I.e. the step of merging the two closest clusters is replaced by merging the two closest clusters from all possible merges that do not violate the given constraints. For all constraints (o_x, o_y, o_z), o_x is only merged with o_z, if it was already merged with o_y. We decided not to initialize clusters with groups of items known to belong together. As we use centroid linkage for cluster similarity computation, this could have a negative impact on similarity computation to the other elements. Furthermore, such groups could only be formed for leaf node classes in the given hierarchy. And if constraints are only given by triplets (o_x, o_y, o_z), such groups are even not specifically known in advance.

Our metric based approach uses the cosine similarity as basis, because this is often used in the domain of text mining. It is defined as

$$\mathrm{sim}(o_1, o_2) = \frac{o_1^T o_2}{|o_1| \cdot |o_2|} = \frac{\sum_i o_{1,i} o_{2,i}}{\sqrt{\sum_i o_{1,i}^2}\sqrt{\sum_i o_{2,i}^2}} \quad (2)$$

and is therefore by definition independent from vector length.

A weight vector w is defined that contains a weight for each dimension (i.e. each feature/term). Its goal is to express that certain terms/features are more important for the similarity as others. We require that each weight is greater than or equal to zero and that their sum is bound to the number of features to avoid extreme case solutions (see (3)). Setting all weights to 1 is a special included weighting scheme that expresses the standard case of no feature weighting.

$$\forall w_i : w_i \geq 0 \qquad \sum_i w_i = n \quad (3)$$

The weights are directly integrated in the vector representation. We use unnormalized $tf \times idf$ vectors for the representation of text documents. This ensures that we do not carry over effects from normalization as described in Sect. 3. The document vectors therefore look as follows:

$$d = \begin{pmatrix} \sqrt{w_1} \cdot d_1 \\ \cdots \\ \sqrt{w_n} \cdot d_n \end{pmatrix} = \begin{pmatrix} \sqrt{w_1} \cdot tf_{1,d} \cdot \log idf_1 \\ \cdots \\ \sqrt{w_n} \cdot tf_{n,d} \cdot \log idf_n \end{pmatrix} \quad (4)$$

Instead of integrating each weight linearly, we use its square root. That has mainly computational reasons. First, in all necessary computations, i.e. similarity or vector length computation, vector elements are always multiplied. Therefore, only w_i^2 can be found, which can be substituted by another weight $w_i' = w_i^2$ as a consequence. For the computation of the weight update rule, this substitution is especially beneficiary, as it leads to an update rule that can also modify weights that are currently 0 (see later on). To sum up, similarity between documents is therefore computed by the cosine similarity in (2) between our weighted document vectors as defined in (4).

For learning the weights, all constraint triplets (o_x, o_y, o_z) are presented to the algorithm several times. For each violated constraint, w is updated using a gradient descent approach. Each constraint is thereby understood in providing a relation between object similarities (see (5)). This means that o_x should be more similar to o_y then to o_z. This relation is also used to guide the gradient descent, trying to maximize (7) for each constraint. This leads to the weight update rule in (6), where η is the learning rate defining the step width of each adaptation step.

$$\mathrm{sim}(o_x, o_y) - \mathrm{sim}(o_x, o_z) > 0. \quad (5)$$

$$w_i \leftarrow w_i + \eta \Delta w_i = w_i + \eta \frac{\partial obj_{xyz}}{\partial w_i} \quad (6)$$

$$obj_{xyz} = \mathrm{sim}(o_x, o_y) - \mathrm{sim}(o_x, o_z) \quad (7)$$

The final computation of Δw_i after differentiation is shown in (8), with similarity and vector length computed on the weighted vectors. As you can see, this formula can also modify weights that are currently zero as it is not of the form $w_i \cdot term$. However, that would have been the case, if vectors were weighted by w_i instead of $\sqrt{w_i}$. After all weights are updated by (6), all negative weights are set to 0. Finally, the weights are normalized to sum up to n to ensure (3).

$$\begin{aligned} \Delta w_i \quad = \quad & \overline{o}_{x,i}(\overline{o}_{y,i} - \overline{o}_{z,i}) - \frac{1}{2}\mathrm{sim}(o_x, o_y)(\overline{o}_{x,i}^2 + \overline{o}_{y,i}^2) \\ & + \frac{1}{2}\mathrm{sim}(o_x, o_z)(\overline{o}_{x,i}^2 + \overline{o}_{z,i}^2) \end{aligned} \quad (8)$$

$$\overline{o}_{x,i} = o_{x,i}/|o_x| \quad (9)$$

Our metric based approach can be summarized as follows. In a first step, a weight vector is learned with a gradient descent approach (see Fig.3). Unfortunately, we cannot assume that weight learning will succeed in producing a weighting scheme that violates no constraints. Therefore, another stopping criterion is needed. As gradient descent is performed, we stop, if the weight change is below a certain

```
learnWeights($MLB$)
    Initialize $w$: $\forall w_i : w_i := 1$
    Do:
        For all $(o_x, o_y, o_z) \in MLB$:
            If $\mathrm{sim}(o_x, o_y) \leq \mathrm{sim}(o_x, o_z)$:
                $\forall i : w_i := w_i + \eta \Delta w_i$
                $\forall i :$ if $w_i < 0$ then $w_i := 0$
                $\forall i : w_i := w_i \frac{n}{\sum_{j=1}^{n} w_j}$
    Until weight change is too small
    Return $w$
```

Figure 3: Weight Learning

threshold. After that, hierarchical agglomerative clustering is performed on our weighted document vectors.

We also evaluated a combined approach. Here, we first learned weights as in our metric based approach. However, after that, the documents were clustered with the instance based clustering approach, whereas the documents were represented by our weighted vectors.

5 Evaluation

We evaluated the approaches from the previous section for its suitability in our learning task. As this task differs from research proposed in the literature, comparisons to other work were not possible. In the following, we first describe the used datasets and evaluation measures. Then we show and discuss our gained results.

5.1 Datasets

As we want to evaluate hierarchical clustering, we need a hierarchical dataset. Unfortunately, such datasets are very rare, as usually only a flat class structure is used. Furthermore, we are interested in text documents here. Fitting our needs, we used the banksearch dataset [Sinka and Corne, 2002] as shown in Fig. 4. As a second dataset, we created one by downloading parts of the open directory (www.dmoz.org). It is shown in Fig. 5. All documents were represented with $tf \times idf$ document vectors. We performed a feature selection, throwing out all terms that occurred less than 5 times, were less than 3 characters long, or contained numbers. From the rest, we selected 5000 terms in an unsupervised manner as described in [Borgelt and Nürnberger, 2004]. This number showed in preliminary tests to just have a small impact on initial clustering performance.

- **Finance** (0)
  - ○ Commercial Banks (100)
  - ○ Building Societies (100)
  - ○ Insurance Agencies (100)
- **Science** (0)
  - ○ Astronomy (100)
  - ○ Biology (100)
- **Programming** (0)
  - ○ C/C++ (100)
  - ○ Java (100)
  - ○ Visual Basic (100)
- **Sport** (100)
  - ○ Soccer (100)
  - ○ Motor Racing (100)

Figure 4: Class structure of the banksearch dataset

- **Fitness** (50)
  - ○ Gyms (50)
  - ○ Personal Training (50)
  - ○ Pilates Method (50)
- **Society** (0)
  - ○ Activism (50)
    - ○ Anti-Corporation (50)
    - ○ In Daily Life (49)
    - ○ Media (50)
    - ○ Nonviolence (50)
- **Society cont.**
  - ○ Paranormal (50)
    - ○ Crop Circles (50)
    - ○ Ghosts (50)
    - ○ Prophecies (50)
    - ○ Psychic (50)
      - ○ Animals (50)
      - ○ Healers (28)
      - ○ Ouija (50)
    - ○ UFOs (50)

Figure 5: Class structure of the open directory dataset

We evaluated different settings to simulate different user data. For both datasets, we evaluated a classification scenario, i.e. a setting, where all classes are known in advance. For the banksearch data, we also evaluated two settings with unknown classes: (a) *Motor Racing*, (b) *Science* (including subclasses). Furthermore, we used different numbers of given data per class with a maximum of 20% of all documents in a class. We did not use higher numbers as we assume for an application point of view that it is much more likely that labeled data is rare.

5.2 Evaluation Measures

We used two measures to evaluate and compare the performance of our algorithms. First, we used the f-score gained in accordance to the given dataset, which is supposed to be the true user defined class structure that shall be recovered. For its computation in an unlabeled cluster tree, we followed a common approach that selects for each class in the dataset the cluster gaining the highest f-score on it. This is done for all classes in the hierarchy. For a higher level class, all documents contained in sub-classes are also counted as belonging to this class. Please remark, that it might occur that clusters are selected hierarchy inconsistently or multiple times in the case of noisy clusters. As f-score is a class specific value, we computed two mean values: one over all leaf node classes (ll) and one over all non-leaf node classes (hl).

As the f-score measure only evaluates a part of the clustering by only considering the best cluster per class, we introduce a second measure that aims at evaluating the total clustering. We call it the cluster error CE. It measures how many constraints were violated by the current clustering in the resulting dendrogram. For each document d with a given constraint $(d,(S_1,...,S_m)) = (d,\mathcal{S})$, we can derive from the dendrogram of the current clustering the order, in which documents are merged to d, producing another ordered list of document sets $\mathcal{R} = (R_1,...,R_r)$. To compute CE, we count how often the order in $\mathcal{R}$ violates the order in $\mathcal{S}$, i.e. the order of two documents d_i and d_j is reversed. In addition, we assume violations between classes that are further apart in the hierarchy as more severe. Therefore, constraint violations are counted weighted. The distance between two classes is reflected in the distance between two constraint sets in the ordered list, which is therefore used for weighting. The overall cluster error CE is then computed by:

$$CE = \sum_{(d,\mathcal{S},\mathcal{R})} \sum_{\substack{(d_{i_{k,x}},d_{j_{l,y}})\\ k<l}} \left\{ \begin{array}{ll} x-y & \text{if } x>y \\ 0 & \text{else} \end{array} \right\}, \quad (10)$$

where $d_{i_{k,x}} \in R_k$, $d_{i_{k,x}} \in S_x$, $d_{j_{l,y}} \in R_l$ and $d_{j_{l,y}} \in S_y$.

5.3 Results

The presented results were gained with the algorithms described in Sec. 4. For the metric learning approach, we fixed the number of iterations to 30 for the banksearch data and to 10 for the open directory data. This was mainly done due to time limitations. However, it impacts the results as an optimum might not have been reached.

Figures 6-17 show the results. In all figures, results are shown for all three algorithms, depending on different numbers of available labeled data. Figures 6, 8, 11, and 16 show the cluster error on the complete dataset. Figures 7, 9, 12, and 17 show the f-score gained on the known classes of the respective setting. Mean values are computed for the leaf nodes in the hierarchy tree (ll) and for the intermediate nodes (hl). However, the unknown classes were not used in computing these values. They are shown separately in figures 10, 13, 14, and 15. More detailed, Fig. 10 shows the f-score for the Motor Racing class, Fig. 13 the f-score of the Science class, Fig. 14 the f-score of the Astronomy

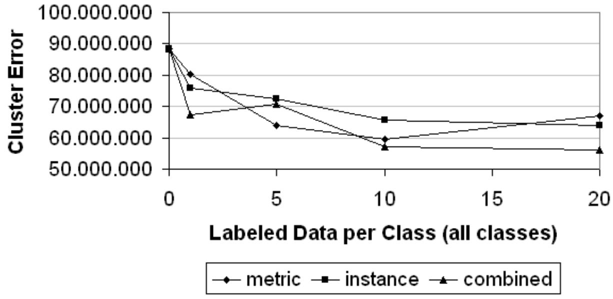


Figure 6: Banksearch; No unknown classes

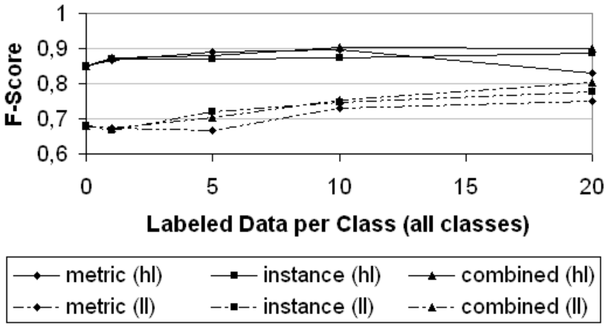


Figure 7: Banksearch; No unknown classes

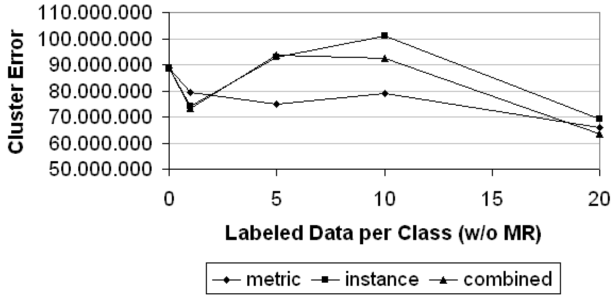


Figure 8: Banksearch; Unknown class Motor Racing

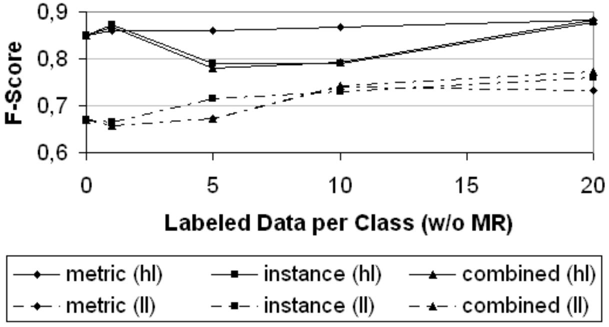


Figure 9: Banksearch; Unknown class Motor Racing

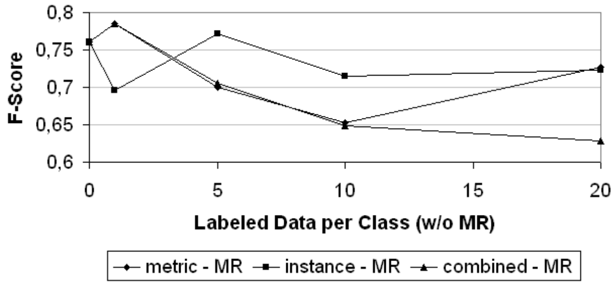


Figure 10: Banksearch; Unknown class Motor Racing

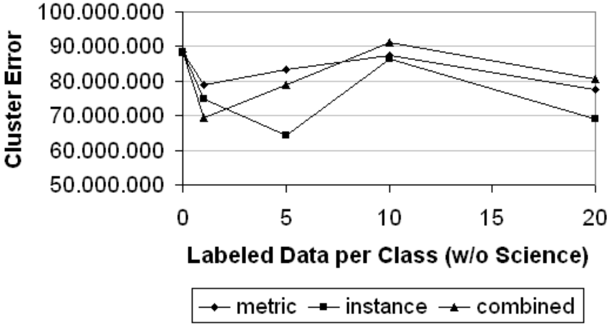


Figure 11: Banksearch; Unknown class Science

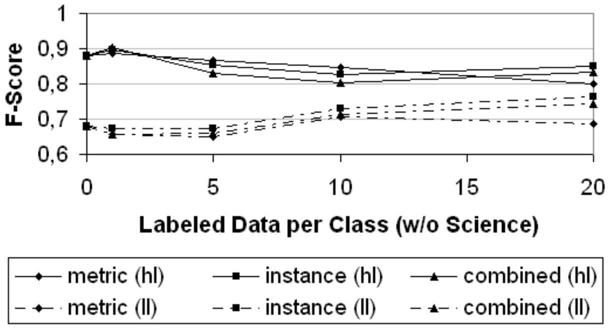


Figure 12: Banksearch; Unknown class Science

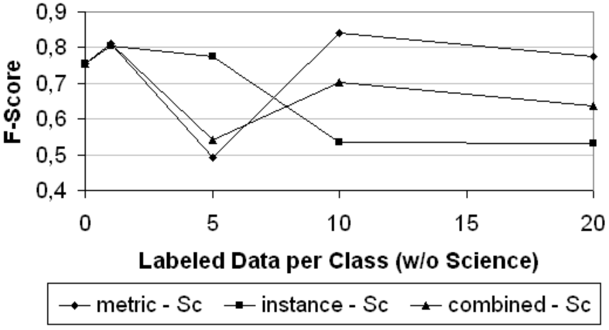


Figure 13: Banksearch; Unknown class Science

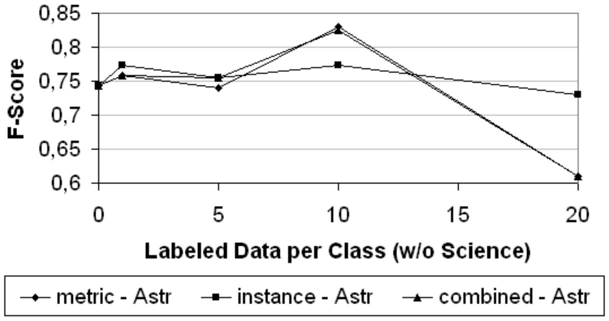


Figure 14: Banksearch; Unknown class Science

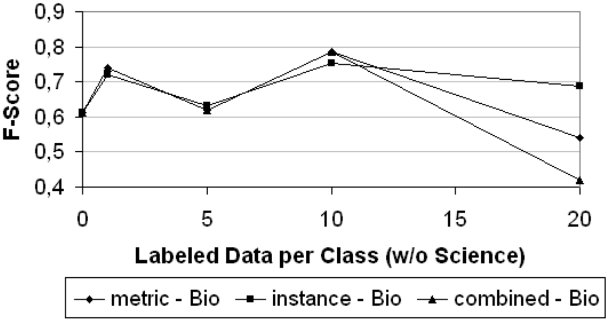


Figure 15: Banksearch; Unknown class Science

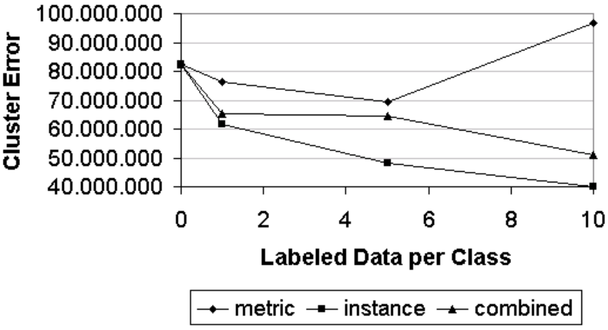


Figure 16: Open Directory; No unknown classes

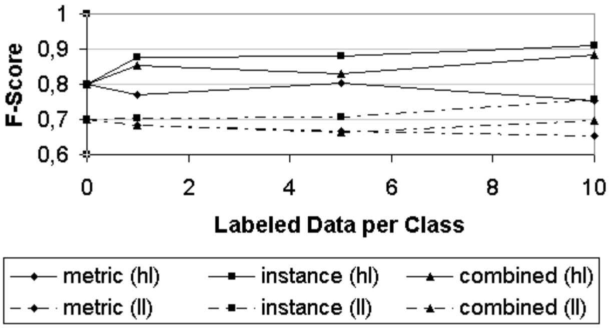


Figure 17: Open Directory; No unknown classes

class, and Fig. 15 the f-score of the Biology class belonging to the settings, where these classes were unknown.

In general, it can be noted that all three approaches are capable of improving cluster quality. However, the specific results differ a lot over the different settings analyzed. Starting with the banksearch dataset and all classes known (Fig. 6, 7), the combined approach seems to be the best choice, combining the best of both worlds. Nevertheless, all approaches behave quite similar, increasing the performance with increasing number of given labeled items. For the open directory data (Fig. 16, 17), the constraint based approach outperforms the metric based approach. However, we think that this might be due to the few number of iterations that we used in the weight learning algorithm. This requires further evaluation in the future.

With increasing number of unknown classes, the results get less stable. Omitting a single class (the Motor Racing class, Fig. 8-10) has only marginally effects on the performance for the metric based approach, while the instance based approach (and with it the combined approach) gets worse on the higher class level. The performance of the Motor Racing class itself is fluctuating, varying between improvement and deterioration. Omitting an even larger part of the hierarchy (the Science class, Fig. 11-15) has even stronger effects. On the higher hierarchy level, performance generally decreases. However, on the specific hierarchy level, performance still increases, although to a smaller amount. As in the previous scenario, the performance of the unknown classes is unstable.

From these experiments, we draw the following conclusions. First, it has been shown that an uneven distribution of constraints over the instance space has a negative influence on the performance of both metric based and instance based approaches. Future work needs to investigate how the performance of such unknown classes can be stabilized. Second, a thorough analysis of the class specific f-score values for the metric based approach showed that not all classes improve equally. While some are improving, others deteriorate. That indicates that feature weighting of different classes is conflicting. This gets especially true, the more classes exist. Hierarchies usually tend to include more classes as considered in flat scenarios, as a high number of classes is still manageable by the user within hierarchies. Therefore, in hierarchical settings, we assume the problem to be more dominant. To avoid or at least minimize the problem, the development of local weighting schemes rather than global ones is a research goal of future work.

6 Conclusion

In this paper, we dealt with semi-supervised hierarchical structuring of collections. The specific requirements of such a hierarchical setting were specified. We then analyzed how constraints can be exploited in such a setting. We proposed and evaluated an instance based and a metric based approach as well as their combination. Special attention was drawn to the problem of unevenly distributed constraints, which have a great impact on the final cluster performance. Here, further future work is needed.

The result of our presented methods is always a dendrogram, i.e. a complete hierarchical representation of the data. However, for user interaction, the extraction of a more coarse grained structure is required to allow efficient access to the data. Furthermore, cluster labels are necessary, so that the user is capable of identifying interesting clusters quickly. Both tasks can also be viewed under semi-supervised aspects. The interested reader shall therefor be referred to [Bade *et al.*, 2007].

References

[Bade and Nürnberger, 2006] K. Bade and A. Nürnberger. Personalized hierarchical clustering. In *Proceedings of the 2006 IEEE/WIC/ACM Int. Conference on Web Intelligence*, 2006.

[Bade *et al.*, 2007] K. Bade, M. Hermkes, and A. Nürnberger. User oriented hierarchical information organization and retrieval. In *Proceedings of the 2007 European Conference on Machine Learning (ECML)*, 2007.

[Bar-Hillel *et al.*, 2003] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *Proc. of the 20th International Conference on Machine Learning (ICML 2003)*, pages 11–18, 2003.

[Basu *et al.*, 2004] S. Basu, A. Banerjee, and R. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proc. of the 4th SIAM Int. Conf. on Data Mining*, 2004.

[Bilenko *et al.*, 2004] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of the 21st International Conference on Machine Learning (ICML04)*, pages 81–88, 2004.

[Borgelt and Nürnberger, 2004] Christian Borgelt and Andreas Nürnberger. Fast fuzzy clustering of web page collections. In *Proc. of PKDD Workshop on Statistical Approaches for Web Mining (SAWM)*, 2004.

[Davidson and Ravi, 2005] Ian Davidson and S. S. Ravi. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In *Proc. of the 9th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD05)*, pages 59–70, 2005.

[Kestler *et al.*, 2006] Hans A. Kestler, Johann M. Kraus, Günther Palm, and Friedhelm Schwenker. On the effects of constraints in semi-supervised hierarchical clustering. In F. Schwenker and S. Marinai, editors, *Artificial Neural Networks in Pattern Recognition*, volume 4087 of *LNAI*, pages 57–66, 2006.

[Kim and Lee, 2002] H. Kim and S. Lee. An effective document clustering method using user-adaptable distance metrics. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 16–20, New York, NY, USA, 2002. ACM Press.

[Schultz and Joachims, 2004] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Proceedings of Neural Information Processing Systems*, 2004.

[Sinka and Corne, 2002] M. Sinka and D. Corne. A large benchmark dataset for web document clustering. In *Soft Computing Systems: Design, Management and Applications, Vol. 87 of Frontiers in Artificial Intelligence and Applications*, pages 881–890, 2002.

[Wagstaff *et al.*, 2001] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of 18th International Conference on Machine Learning*, pages 577–584, 2001.

[Xing *et al.*, 2003] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512. 2003.

Parameterfreies Clustering durch Mehrstufigen Ansatz

Dirk Habich, Martin Hahmann, Wolfgang Lehner
Technische Universität Dresden
Lehrstuhl für Datenbanken
dbinfo@mail.inf.tu-dresden.de

Abstract

Das Ziel des Clusterings besteht darin, eine (semi-)automatische Partitionierung einer Datenmenge in Gruppen durchzuführen, so dass Objekte in einer Gruppe möglichst ähnlich und Objekte verschiedener Gruppen möglichst unähnlich zueinander sind. In allen Anwendungsgebieten muss nach der Festlegung eines Ähnlichkeitsmaßes ein geeignetes Verfahren aus der Menge der vorhandenen ausgewählt werden. Diese Auswahl des am besten geeigneten Verfahrens ist sehr komplex und muss heutzutage durch einen Experten erledigt werden. Des Weiteren ist die Bestimmung der optimalen Parameterwerte für dieses Verfahren bezüglich der betrachteten Daten oftmals nicht trivial. Um ein gutes Partitionierungsergebnis zu erhalten, wird dieses Verfahren iterativ mit angepassten Parameterwerten auf die Daten angewandt. Um diese Komplexität zu reduzieren, präsentieren wir unsere Lösung zum parameterfreien Clustering durch einen mehrstufigen Ansatz, wobei wir uns das Clustering-Aggregation-Konzept effizient zu Nutze machen.

1 Motivation

Die Menge der Daten, die in relationalen Datenbanksystemen gespeichert werden, nimmt ständig zu. Eine manuelle Analyse zur Wissensextraktion übersteigt aber die menschlichen Kapazitäten und somit gewinnt das Gebiet des *Knowledge Discovery in Databases (KDD)* immer mehr an Bedeutung. KDD ist der Prozess der (semi-)automatischen Extraktion von Wissen in Datenbanken, das gültig, bisher unbekannt und potenziell nützlich ist [3; 4].

Eine wichtige Phase innerhalb des KDD ist die Data-Mining-Phase, in der effiziente Algorithmen Anwendung finden, um die in den Daten enthaltenen gültigen Muster zu extrahieren. Zu den relevanten Data-Mining-Aufgaben zählt das Clustering, wo es um die Partitionierung einer Datenmenge in Gruppen (Cluster) von Objekten geht, so dass Objekte eines Clusters möglichst ähnlich und Objekte verschiedener Cluster möglichst unähnlich sind. Üblicherweise wird die Ähnlichkeit durch eine Distanzfunktion modelliert, d.h. die Distanz zwischen Objekten wird als Ähnlichkeitsmaß herangezogen. Eine geringe Distanz zwischen zwei Objekten bedeutet, dass es sich um ähnliche Objekte handelt, während eine große Distanz anzeigt, dass es sich um unähnliche Objekte handelt. Je besser also die Distanzfunktion die Ähnlichkeit in den Daten widerspiegelt, um so bessere Ergebnisse kann das Clustering liefern.

Die Hauptschwierigkeit im Bereich des Clustering liegt in der Vielzahl der vorhandenen Algorithmen und deren Anwendung auf unbekannte Datenmengen. Die existierenden Algorithmen lassen sich grob in vier Klassen einteilen: *partitionierende*, *hierarchische*, *dichtebasierte* und *hierarchisch dichtebasierte Verfahren*. Diese Verfahren unterscheiden sich deutlich in ihrer Mächtigkeit hinsichtlich der Extraktion von Clustern. So können beispielsweise mit *dichtebasierten Verfahren* Cluster mit unterschiedlicher Dichte und beliebiger Form bestimmt werden, was mit *partitionierenden Verfahren* nur schwer möglich ist. Aus diesem Grund ist die Bestimmung des besten Verfahrens für eine unbekannte Datenmenge sehr schwierig und wird oftmals dadurch gelöst, dass verschiedene Algorithmen aus den Klassen auf die Datenmenge angewandt und die verschiedenen Ergebnisse analysiert werden.

Neben der Auswahl der optimalen Verfahrensklasse für eine unbekannte Datenmenge, ist die spezifische Anwendung von Algorithmen aus den unterschiedlichen Klassen keine triviale Angelegenheit. Jede Verfahrensklasse zeichnet sich dadurch aus, dass sie eine Vielzahl von Algorithmen enthält und jeder Algorithmus über Parameter verfügt, die einen starken Einfluss auf das Ergebnis haben. Die Anzahl der Parameter schwankt sehr stark bei den Algorithmen. Darüber hinaus existieren wenige Heuristiken, um die optimalen Parameterwerte für unbekannte Datenmengen bestimmen zu können. Um trotzdem eine zufriedenstellende Partitionierung zu erhalten, sind Hintergrundinformationen über die Daten notwendig oder die Parameter werden iterativ angepasst, was in der Mehrfachausführung der Algorithmen resultiert.

Im Allgemeinen kann festgehalten werden, dass das Clustering von Datenmengen spezifisches Fachwissen über die Algorithmen und die Daten voraussetzt, um eine optimale Partitionierung der Daten zu extrahieren. Dieser Gegenstand ist Motivation für uns, das Clustering von Datenmengen auf eine abstraktere Ebene zu heben, wo diese Probleme nicht mehr auftreten. Diese Arbeit kann dabei als ein erster Schritt in diese Richtung angesehen werden. Hier geht es uns speziell darum, verschiedene Verfahren so zu kombinieren, dass ein parameterfreies Clustering möglich ist. Durch die Kombination verschiedener Algorithmen reduziert sich der Einfluss eines einzelnen Algorithmus mit spezifischer Parameterbelegung. Ein wichtiger Aspekt dabei ist die Zusammenführung von einzelnen Clustering-Ergebnissen, wie sie bereits in [5; 7] betrachtet wurden.

Die Arbeit ist folgendermaßen strukturiert: In Abschnitt 2 gehen wir präziser auf vorhandene Konzepte zum Clustering ein. Aufbauend darauf präsentieren wir unseren *mehrstufigen Ansatz* in Abschnitt 3. Bevor wir diesen Forschungsbericht mit einer Zusammenfassung schliessen, präsentieren wir in Abschnitt 4 eine Evaluierung.

2 Verwandte Arbeiten

In diesem Abschnitt geben wir einen Überblick über verwandte Arbeiten. Des Weiteren wollen wir verwendete Konzepte unseres Ansatzes im Detail beschreiben.

2.1 Clustering

Wie bereits in Abschnitt 1 erwähnt worden ist, lassen sich die Algorithmen für das Clustering in vier Klassen einteilen: *partitionierende*, *hierarchische*, *dichtebasierte* und *hierarchisch dichtebasierte Verfahren*. In diesem Abschnitt wollen wir insbesondere auf *k-means* als Vertreter für partitionierende Verfahren und *DBSCAN* als dichtebasierten Vertreter eingehen. Weitere Beschreibungen relevanter Algorithmen finden sich in [8; 10; 11].

Clustering mit *k-means*

Partitionierende Verfahren zeichnen sich dadurch aus, dass sie die Datenmenge in eine Zahl von Clustern zerlegen, wobei jedes Objekt maximal einem Cluster zugeordnet ist und jeder Cluster mindestens ein Objekt enthält. Ausgehend von einer initialen zufälligen Zerlegung in Cluster, wird mit der in [1; 3] beschriebenen Methode eine Lösung durch Umgruppierung erzielt. In einem ersten Schritt wird für die initiale Zerlegung für jeden Cluster sein Zentroid berechnet. Dann werden neue Cluster gebildet, indem jedes Objekt dem Cluster zugeordnet wird, zu dessen Zentroid es den geringsten Abstand aufweist. Dieser Schritt wird wiederholt, bis sich die Zusammensetzung der Cluster nicht mehr verändert. Es ist zu beachten, dass beim Clustering mit *k-means* unter Umständen nur ein lokales Minimum geliefert wird.

Durch den verhältnismäßig geringen Rechenaufwand, kommt das Clustering mit *k-means* oft zum Einsatz und ist einer der verbreitetsten Algorithmen zum Clustering. Als Input erwartet der *k-means*-Algorithmus die Datenmenge und die Anzahl der Cluster k. Bedingt durch den Ansatz werden Cluster als kreisförmig angesehen und stark gestreckte Cluster in den Datenmengen mit hoher Wahrscheinlichkeit nicht korrekt klassifiziert.

Clustering mit DBSCAN

Eine weitere Möglichkeit zum Clustering bieten dichtebasierte Verfahren. Ein Cluster wird hier nicht mehr über die durchschnittliche Distanz seiner Objekte zu einem zentralen Punkt, sondern über einen durch die Objektdichte definierten Bereich beschrieben. Dabei werden Bereiche hoher Objektdichte von Bereichen niedriger Objektdichte separiert. Ein Cluster beinhaltet demnach nur Objekte, die eine minimale Anzahl an benachbarten Objekte in einer um sie festgelegten Umgebung nicht unterschreiten. Eine genaue Spezifikation ist in [2] nachzulesen.

Jeder durch dichtebasiertes Clustering erzeugte Cluster beinhaltet eine Menge von Objekten, die miteinander dichteverbunden sind, zuzüglich aller von den Kernobjekten des Clusters dichteerreichbaren Objekte. Ist ein Kernobjekt gegeben, kann, durch Suche aller von diesem Objekt aus dichteerreichbaren Objekte, die Gesamtobjektmenge des Clusters bzgl. der Parameter ϵ und $MinPts$ bestimmt werden. Mit dem Algorithmus DBSCAN (Density-Based Clustering of Applications with Noise), beschrieben in [2], kann eine nicht-klassifizierte Objektmenge mit einem Aufwand von O(n * Aufwand zum Finden der Nachbarn innerhalb ϵ) in dichtebasierte Cluster partitioniert werden.

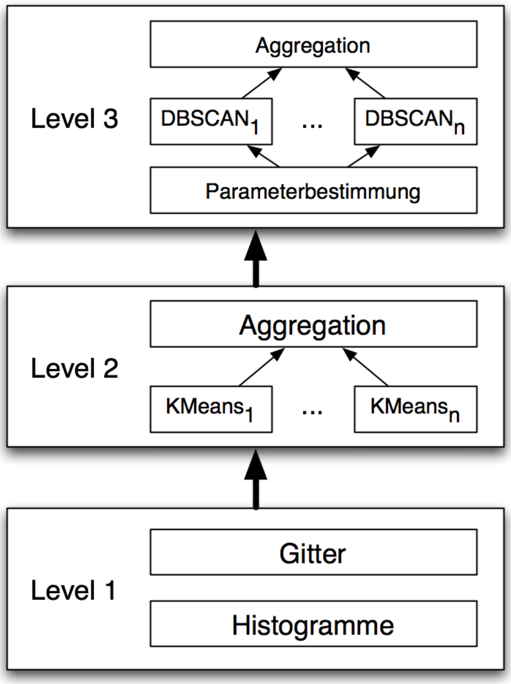


Abbildung 1: Parameterfreies Clustering mit dem mehrstufigen Ansatz

Die Qualität des Clusterings hängt bei dichtebasierten Verfahren in hohem Maße von den Parametern ϵ und $MinPts$ ab. Diese Parameter gehören, wie auch die Anzahl an zu erwartenden Clustern k, nicht zum vorher bekannten Wissen und müssen heuristisch bestimmt werden. Auf geeignete Heuristiken soll hier nicht weiter eingegangen werden; Vorschläge werden in [2; 3] gegeben.

Im Gegensatz zu *k-means* ist es mit dem hier beschriebenen Ansatz zum dichtebasierten Clustering auch möglich, Cluster beliebiger Form zu finden. Allerdings stößt man bei der Bestimmung geeigneter Werte für ϵ und $MinPts$ auf Probleme, wenn die Daten durch hierarchisch geschachtelte Cluster beschrieben werden, Cluster und Rauschen nicht gut unterscheidbar sind oder die Dichteverteilung im Datenraum sehr stark variiert. In eben diesen Fällen ist es gegebenenfalls nicht möglich, Parameter zu finden, welche die Menge korrekt in Cluster teilen.

2.2 *Clustering Aggregation*

In [5] setzen sich Gionis et al. mit dem Problem der Aggregation einer Menge von Clustering-Ergebnissen auseinander. Allgemein lässt sich dieses wie folgt beschreiben: zu einer Menge von Clusterings soll ein Clustering berechnet werden, das so stark wie möglich mit den gegebenen Clusterings übereinstimmt. Der Forschungsbericht gibt eine formale Definition des Problems der *Clustering Aggregation* und stellt Definitionen und Algorithmen zur Lösung vor.

Auf dem relativ jungen Gebiet der *Clustering Aggregation* gibt es bereits einige Veröffentlichungen, wie zum Beispiel [7]. Das globale Ziel dieser Arbeiten besteht darin, ein robustes Clustering durch die Zusammenführung von Ergebnissen unterschiedlicher Algorithmen zu bestimmen. Die präsentierten Ergebnisse untermauern dieses Ziel.

3 Mehrstufiger Clustering-Ansatz

Im Rahmen dieser Arbeit sind bereits einige gebräuchliche Clustering-Algorithmen sowie das Verfahren der *Clustering Aggregation* erläutert worden. In diesem Abschnitt präsentieren wir unseren mehrstufigen Ansatz zum parameterfreien Clustering, wobei innerhalb der unterschiedlichen Stufen verschiedene Clustering-Algorithmen angewandt werden. Durch mehrfache Aggregation der entstandenen Ergebnisse soll eine Partitionierung der betrachteten Datenmenge erzeugt werden, die eine bessere Qualität aufweist. Weiterhin sollen alle notwendigen Parameter für die Algorithmen innerhalb der Stufen automatisch abgeleitet werden. Dadurch wird ein aus der Sicht des Anwenders parameterfreies Clustering möglich.

In Abbildung 1 ist der Aufbau unseres mehrstufigen Clusterings schematisch dargestellt. Wie zu erkennen ist, beinhaltet unser Ansatz drei Stufen (Level), wobei auf jedem Level unterschiedliche Verfahren zum Einsatz kommen. Im Folgenden werden die einzelnen Level detailliert beschrieben. Dabei werden zum besseren Verständnis grafische Beispiele eingesetzt. Die Grundlage für diese Beispiele ist die in Abbildung 2 dargestellte 2-dimensionale Datenmenge mit vier Clustern, von denen zwei hierarchisch sind.

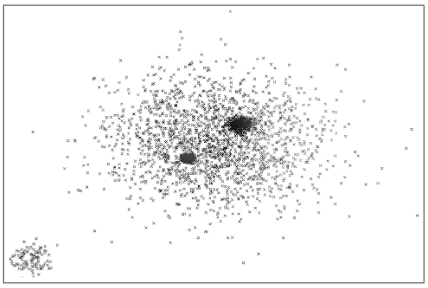


Abbildung 2: Datenmenge mit 4 Clustern (2 davon hierarchisch)

3.1 Level 1

Das erste Level muss als Initialisierungsstufe unseres Ansatzes angesehen werden. In ihm werden die zu partitionierenden Daten erstmals vermessen und strukturiert, um eine Weiterverarbeitung durch die nachfolgenden Stufen zu ermöglichen. Um auch für diese Stufe des Ansatzes das Ziel der Parameterfreiheit zu erfüllen, müssen die verwendeten Operationen selbst parameterfrei sein.

Die Initialoperation dieses Levels erstellt ein Histogramm für jede Dimension des in Cluster zu zerlegenden Datensatzes. Histogramme sind klassische Mittel der Statistik, die Auskunft über die Häufigkeitsverteilung einer Datenmenge geben. Im Rahmen dieser Arbeit wird das vorgeschlagene Verfahren von David W. Scott [9] verwendet.

Durch Histogramme stehen nun Informationen über die Häufigkeitsverteilung innerhalb der einzelnen Dimensionen zur Verfügung. Aus diesen lassen sich kaum Aussagen über Häufigkeits- bzw. Dichteverteilung der Datenobjekte über den gesamten Raum des Datensatzes ableiten. Um mehr Informationen über die Objektverteilung und Strukturen innerhalb des Datenraumes zu erhalten, benutzen wir ein Gitter (Grid), das den Datenraum in gleichgroße Teilräume zerlegt. Die Wahl der Gitterzellgröße bzw. -anzahl hat einen großen Einfluss auf die Qualität der Ergebnisse, die mit dem Gitter gewonnen werden. Aufgrund der starken Schwankungen in Datenverteilung und Größe, die zwischen verschiedenen Datenmengen auftreten können, ist die Verwendung von festen Werten für die Gitterzellen im Hinblick auf die Ergebnisqualität ebenfalls nicht zu empfehlen. Aus diesem Grund generieren wir die Gitterzellgrößen aus den Ergebnissen der Histogramme. Zu jeder Dimension des Datenraums existiert bereits ein Histogramm, welches diese in k gleich große Intervalle teilt. Diese Unterteilung wird zur Berechnung der Zellanzahl und -größe genutzt.

Nach der Bestimmung der Gitterzellgrößen werden die Datenobjekte den Gitterzellen zugewiesen. Da wir nur an den Teilen des Datenraumes interessiert sind, die Datenobjekte enthalten, werden alle *besetzten* Zellen des Grids markiert. Eine Zelle gilt genau dann als *besetzt*, wenn sie mindestens ein Datenobjekt enthält. Aufgrund dieser Definition werden auch solche Gitterzellen als besetzt markiert, die eigentlich nur Rauschen enthalten; dieses Rauschen gilt es zu filtern. Weiterhin sollen die besetzten Zellen auf die relevanten, relativ dichten Bereiche des Datensatzes beschränkt werden. Um dies zu erreichen, wird jede besetzte Zelle z wieder als leer markiert, wenn Sie folgende Bedingung erfüllt:

$$elem(z) \leq round(\frac{|O|}{\prod_{i=1}^{n} k_i}). \quad (1)$$

Der Ausdruck $elem(z)$ steht dabei für die Anzahl der Datenobjekte, die innerhalb einer Zelle z liegen, $|O|$ ist die Anzahl der Datenobjekte insgesamt und $\prod_{i=1}^{n} k_i$ ist die Anzahl der Gitterzellen. Durch $round(W)$ wird der in Klammern angegebene Wert W kaufmännisch auf eine ganze Zahl gerundet. Kurz gesagt, es wird die durchschnittliche Anzahl von Datenobjekten pro Gitterzelle berechnet und auf eine ganze Zahl gerundet. Die verbliebenen markierten Zellen repräsentieren den Bereich der Datenmenge, der mit hoher Wahrscheinlichkeit Cluster enthält.

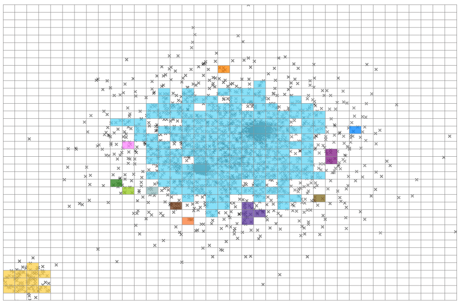


Abbildung 3: Zusammenfassung von Zellen zu Regionen

Da wir in diesem Bereich der besetzten Zellen Cluster vermuten, fassen wir benachbarte besetzte Zellen zu Regionen zusammen. Die Nachbarn einer Zelle z sind alle Zellen, die in einer Dimension direkt vor oder nach z im Gitter liegen. In Abbildung 3 sind nach dieser Definition für unser Beispiel die erstellten Regionen farblich dargestellt. Nach der Identifikation der Regionen erfolgt noch einmal eine Kategorisierung der Gitterzellen. Diese hat das Ziel, die Zellen einer Region zu finden, die eine sehr viel höhere Elementanzahl aufweisen als die restlichen Zellen der Region. Sollten solche *hochbesetzten* Zellen innerhalb einer Region existieren, so kann man daraus schließen, dass in dieser Region sogenannte hierarchische Cluster existieren. Damit solche Hierarchien im späteren Verlauf gefunden werden können, werden sie im Gitter separat markiert.

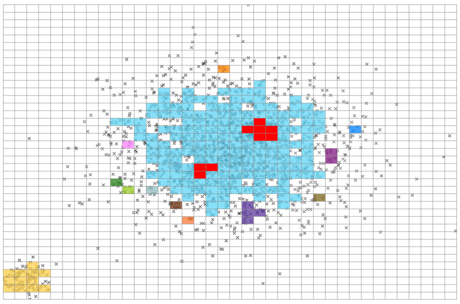


Abbildung 4: Regionen mit hochbesetzten Zellen in rot

Eine Zelle z gilt dabei als hochbesetzt innerhalb einer Region r, wenn folgende Regel gilt:

$$elem(z) > 2 \cdot round(\frac{elem(r)}{|Z_r|}). \qquad (2)$$

Dabei steht $elem(r)$ für die Anzahl der Datenobjekte, die innerhalb der Zellen der Region r liegen und Z_r für die Menge der Zellen, die zu r gehören. In Abbildung 4 werden die hochbesetzten Zellen unseres Beispiels in rot dargestellt.

3.2 Level 2

Dieses Level besteht aus zwei Schritten, wie in Abbildung 1 bereits dargestellt ist. Im ersten Schritt werden mit Hilfe eines auf *k-means* basierenden, partitionierenden Algorithmus (*k-means**) mehrere Clusterings des Datensatzes erzeugt. Der zweite Schritt besteht aus der Aggregation der einzelnen *lokalen* Ergebnisse der Clusterings.

In diesem Level unseres mehrstufigen Ansatzes werden die Daten erst einmal mit einem schnellen Clusteringverfahren, in unserem Fall mit *k-means**, bearbeitet. Als Input dient dabei das erstellte Gitter aus Level 1. Dieses Grid G enthält eine Menge R, bestehend aus mehreren Regionen r und r^{high}, von denen bekannt ist, dass sie jeweils eine zusammenhängende Menge von Datenobjekten enthalten. Wir nehmen nun an, dass jede dieser Regionen einen Cluster innerhalb der Datenmenge repräsentiert bzw. enthält. Aufgrund dieser Annahme ergibt sich die Anzahl der Cluster für *k-means* wie folgt:

$$k = |R|. \qquad (3)$$

Das Ergebnis des Clusterings mit *k-means** ist jedoch nicht nur abhängig von der Anzahl k, sondern auch von der initialen Positionierung der Zentroide und der Reihenfolge der Datenpunktverarbeitung. Um das zu berücksichtigen, erzeugen wir wie folgt mehrere Clusterings.

Clustering-Modus 1

In diesem Modus wird ein Zentroid c aus dem Mittelwert aller Datenobjekte einer Region r erzeugt. Es gilt also:

$$c = (\hat{x_1}(r), \ldots, \hat{x_n}(r)), \hat{x_j} = \frac{1}{elem(r)} \cdot \sum_{x \in r} x. \qquad (4)$$

Dabei steht n für die Anzahl der Dimensionen und $elem(r)$ für die Anzahl der Datenobjekte innerhalb der Region r. Somit stellt $\hat{x}_j$ den Mittelwert der j-ten Dimension aller Objekte von r dar. Aufgrund dieser Berechnung befinden sich die Zentroide immer im Schwerpunkt ihrer Region.

Der Modus erzeugt nun zwei Zerlegungen, wobei die Bedingungen für *k-means** variiert werden. In der ersten wird für jede Region r ein Centroid bestimmt, während das zweite Clustering eventuelle Hierarchien innerhalb der Cluster berücksichtigen soll. Zu diesem Zweck wird neben jeder Region r auch für jede hochbesetzte Region r^{high} ein Zentroid berechnet. Für unseren Beispieldatensatz ergeben sich die in Abbildung 5 dargestellten Clustering-Ergebnisse.

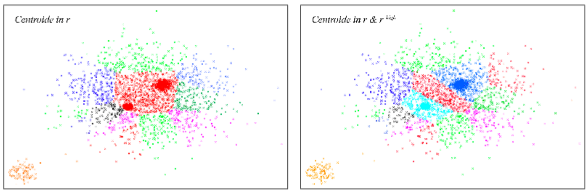


Abbildung 5: Ergebnisse von *k-means** - Modus 1

Betrachten wir diese Ergebnisse und vergleichen sie mit der Gitterapproximation aus den Abbildungen 3 und 4, so werden signifikante Unterschiede deutlich.

Clustering-Modus 2 - Gewichtete Distanzen

In diesem Modus wird versucht, die Probleme, die beim einfachen Einfügen bei dem Verfahren entstehen, zu dämpfen. Es werden in Analogie zu Modus 1 wieder zwei Clusterings erzeugt, wobei die Zentroide identisch bestimmt werden.

Beim einfachen Clustering mit *k-means* treten Probleme bei der Zuordnung der Datenobjekte auf, da jeder Zentroid den gleichen Einfluss besitzt. Um dieses Zuordnungsproblem zu beheben, wird für jeden Zentroiden c ein Gewichtungsfaktor berechnet. Dieser Gewichtungsfaktor orientiert sich in unserem Ansatz an der Größe der jeweiligen Region des Zentroiden. Diese Größe wird durch die Anzahl der Gitterzellen, die zur Region gehören, repräsentiert. Zuerst wird die Region mit den meisten Zellen gesucht. Sie erhält den Gewichtungsfaktor 1 und wird mit r_{max} bezeichnet. Für jede übrige Region r wird der Gewichtungsfaktor wie folgt berechnet:

$$w = \frac{cell(r)}{cell(r_{max}} + 2. \qquad (5)$$

Dabei steht $cell(r)$ für die Anzahl der Zellen einer Region r. Mit Hilfe dieser Formel erhält jedes r dessen Zell-anzahl kleiner ist als die von r_{max} einen Gewichtungsfaktor, welcher größer ist als 1 und den Größenunterschied darstellt. Die berechneten Gewichtungsfaktoren werden bei der Abstandsberechnung zwischen Datenobjekten und Zentroiden beachtet.

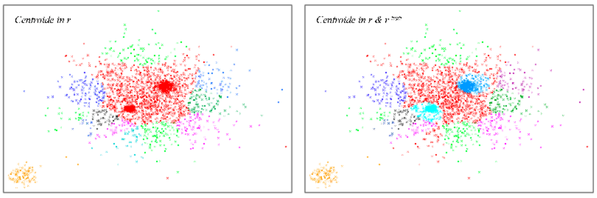


Abbildung 6: Ergebnisse von *k-means** - Modus 2

In Abbildung 6 werden die Clustering-Ergebnisse für unsere Beispieldaten im Modus 2 dargestellt. Man erkennt in den Ergebnissen einen deutlichen Unterschied zu den Ergebnissen aus Modus 1 (siehe Abbildung 5).

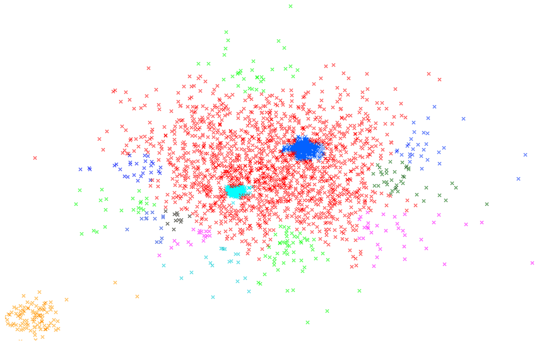


Abbildung 7: Ergebnis von *k-means** - Modus 3

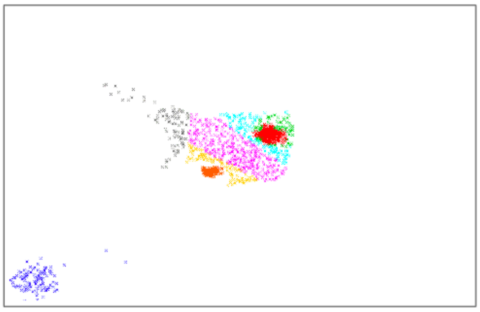


Abbildung 8: Ergebnis der Aggregation in Level 2.

Clustering-Modus 3 - Multiple Zentroide

Der *k-means*-Algorithmus hat Probleme, Cluster zu identifizieren, die sich nicht in allen Dimensionen des Datenraumes gleichmäßig ausdehnen. Sollte ein Cluster gegenüber den anderen eine stark gedehnte oder wenig kompakte Form haben, so wird er oft nicht richtig erkannt. Um dieses Problem abzuschwächen, nutzen wir die Informationen über Form und Ausdehnung von potenziellen Clustern, die aus dem erstellten Grid aus Level 1 extrahierbar sind.

Um die Form einer Region beschreiben zu können, müssen ihre Grenzen zu anderen Regionen bzw. zum leeren Raum bekannt sein. Zur Modellierung dieser Grenzen benutzen wir Grenzzellen mit einer wohldefinierten Semantik. Damit die erkannten Formen der Regionen auch beim Clustering erhalten bleiben, werden im Modus 3 mehrere Zentroide für jede erkannte Region erstellt. Während bis jetzt nur ein Zentroid im Schwerpunkt einer Region erzeugt wurde, wird jetzt für jede Grenzzelle ein Zentroid erzeugt. Das heißt, alle Datenobjekte, die den Zentroiden einer Region zugeordnet werden, gehören zu ein und demselben Cluster. In Abbildung 7 ist das Ergebnis des Clusterings mit multiplen Zentroiden für die Beispieldaten illustriert.

Aggregation

Nachdem insgesamt fünf Zerlegungen entsprechend der vorgestellten Modi erstellt wurden, werden diese im letzten Schritt von Level 2 zusammengefasst. Mit der Durchführung mehrerer Clusterings mit unterschiedlichen Modi wird gewissermaßen versucht, Redundanz zu schaffen.

Sollte eine Datenmenge so strukturiert sein, dass *k-means** in einem Clustering-Modus nur ungünstige oder sogar falsche Zuordnungen liefert, besteht immer noch die Chance, dass ein oder mehrere Clusterings in anderen Modi bessere Ergebnisse erzielen. Durch unseren Ansatz erhalten wir also eine Basis aus verschiedenen Zerlegungen einer Datenmenge. Wir müssen davon ausgehen, dass uns sowohl gute als auch weniger gute Ergebnisse vorliegen. Zur Weiterverarbeitung soll aber nur ein einziges, möglichst optimales Clustering genutzt werden. Da es nicht möglich ist, zu entscheiden, welche Zerlegung die beste ist, soll versucht werden, aus den vorliegenden Clusterings ein ideales Ergebnis zu extrahieren.

Jedes einzelne unserer Clusterings zerlegt die Datenmenge in verschiedene Cluster. Sollte einer dieser Cluster in jedem Clustering vorkommen, so kann mit hoher Wahrscheinlichkeit davon ausgegangen werden, dass er korrekt erkannt wurde und in der Datenmenge entweder in dieser Form oder als Teil eines Clusters vorkommt. Um das ideale Clustering für die Weiterverarbeitung zu erhalten, müssen die Teile der Datenmenge identifiziert werden, die in *jeder* der einzelnen Zerlegungen zusammen in einem Cluster vorkommen. Zur Bestimmung dieser häufig auftretenden Objektmengen haben wir einen neuen Algorithmus zur *Clustering Aggregation* auf Basis der Bestimmung von sogenannten (*Frequent Itemsets*) entwickelt. Eine detaillierte Beschreibung dieses Ansatzes würde den Rahmen dieses Forschungsberichtes sprengen.

Das finale Ergebnis der Aggregation von Level 2 ist in Abbildung 8 für die Beispieldatenmenge dargestellt. Mit dieser Aggregation endet die zweite Stufe unseres Ansatzes. Diese aggregierten Cluster bilden nun die Grundlage für das dritte und letzte Level.

3.3 Level 3

Innerhalb dieses Levels wird die zu untersuchende Datenmenge wieder mehrmals in Cluster zerlegt und wiederum zu einem globalen Ergebnis zusammengefasst. In diesem Level wird für die Segmentierung der aufwendigere, aber mächtigere DBSCAN-Algorithmus genutzt. Vorteil dieses Algorithmus ist, dass unterschiedlich dichte Bereiche und Cluster mit beliebiger Form erkannt werden. Dadurch ist es möglich, Cluster zu finden, die in der zweiten Stufe nicht entdeckt wurden.

Bevor mit der Herleitung und Erklärung des Verfahrens zur Parameterbestimmung für DBSCAN begonnen wird, soll das Ziel verdeutlicht werden, dass wir mittels DBSCAN erreichen wollen. Den Ausgangspunkt für die Operationen von Level 3 stellen die durch Aggregation erstellten Cluster aus Level 2 dar. Wir gehen von der Annahme aus, dass diese aggregierten Cluster mit hoher Wahrscheinlichkeit entweder einen kompletten, tatsächlich in der Datenmenge vorkommenden Cluster oder zumindest einen Teil eines solchen Clusters darstellen. Für jeden aggregierten Level-2-Cluster C_{agg} soll durch DBSCAN ein Clustering erstellt werden. Dieses Clustering soll den durch C_{agg} teilweise oder vollständig repräsentierten Cluster C innerhalb der Datenmenge identifizieren. Da das dichtebasierte DBSCAN-Verfahren mächtiger ist als die bisher angewandten Clusteralgorithmen, kann mit einer verbesserten Ergebnisqualität gerechnet werden. Um unser Ziel zu erreichen und alle Cluster zu identifizieren, müssen wir für jeden einzelnen Cluster C_{agg} aus Level 2 einen Satz Parameter ϵ und $MinPts$ bestimmen und mit den Werten ein Clustering durchführen.

Verfahren zur Parameterbestimmung

Um die Zielstellung der Parameterbestimmung zu erfüllen, benötigen wir also einen Wert für ϵ und einen Wert für $MinPts$, der jeweils charakteristisch für einen gegebenen C_{agg} ist. Konzentrieren wir uns zuerst auf den Parameter ϵ.

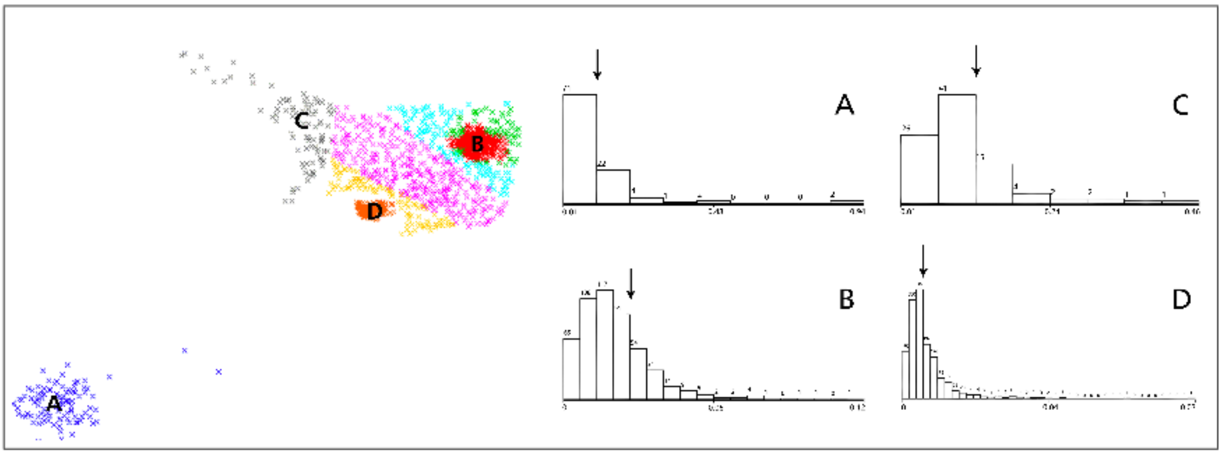


Abbildung 9: Einige C_{agg} der Stufe 2 mit zugehörigen 1-Distanz-Verteilungen

Dieser definiert die Größe des Bereichs um ein Objekt o, in dem sich eine bestimmte Anzahl weiterer Objekte befinden muss, damit o durch DBSCAN als Kernobjekt klassifiziert wird. Wir berechnen zunächst für jeden aggregierten Stufe-2-Cluster C_{agg} die Nächste-Nachbarn-Distanzen, genauer gesagt, die Distanzen zum direkten nächsten Nachbarn (1-Distanz). Innerhalb eines C_{agg} kennen wir nun für jedes Datenobjekt o die Distanz zum nächstgelegenen Objekt (1-Distanz). Diese 1-Distanzen erlauben es uns bereits, im beschränkten Maß Aussagen über die Größe von ϵ zu tätigen. Für jedes Objekt $o \in C_{agg}$ gilt Folgendes:

1. Für o existiert eine 1-Distanz $dist^1(o)$, die den Abstand zwischen o und seinem nächsten Nachbarn darstellt.
2. Sei $dist^1(o)$ der Radius einer n-dimensionalen Hyperkugel, dann liegen genau 2 Objekte innerhalb dieser Kugel: zum einen o und zum anderen der nächste Nachbar von o.
3. Setzt man $\epsilon = dist^1(o)$ und $MinPts = 2$, dann wird o durch DBSCAN als Kernobjekt klassifiziert.

Diese Definition ermöglicht es uns, die Parameter für DBSCAN so zu wählen, dass ein bestimmtes Objekt als Kernobjekt erkannt wird. Betrachten wir alle 1-Distanzen eines Clusters C_{agg}, so können wir weitere Aussagen in Bezug auf ϵ treffen. Der Wert ϵ muss also zum einen groß genug sein, um einen repräsentativen Anteil der Elemente von C_{agg} zu Kernobjekten zu erklren, und zum anderen klein genug sein, um vorhandenes Rauschen zu filtern. Ein günstiges ϵ wäre eine Distanz, die zu den 1-Distanzen einer möglichst großen Gruppe von Objekten $o \in C_{agg}$ möglichst ähnlich ist. Um einen solchen Wert bestimmen zu können, benötigen wir Informationen über den Wertebereich der 1-Distanzen von C_{agg} und deren Häufigkeitsverteilung. Hierzu bietet sich die Nutzung von Histogrammen an. Wir berechnen also für die Menge der 1-Distanzen eines jeden Clusters ein Histogramm. Zur Berechnung nutzen wir erneut das bereits in Stufe 1 vorgestellte Verfahren.

Abbildung 9 zeigt einige aggregierte Level-2-Cluster unseres Beispiels und die Histogramme ihrer 1-Distanzen. Anhand der errechneten Histogramme muss ein Wert für ϵ bestimmt werden. Zur Bestimmung dieses Wertes werden die Anstiege zwischen den einzelnen Intervallbalken betrachtet. Der Parameter soll so gewählt werden, dass durch ihn ein repräsentativer, also relativ großer Teil des Clusters erkannt werden kann. Dazu wird innerhalb des Histogramms nach einer Grenze zwischen hoher und geringer Objektanzahl gesucht. Innerhalb der Histogramme sind die Intervalle aufsteigend geordnet. Das Intervall, bei dem wir unsere Suche beginnen, repräsentiert die Objekte mit den kleinsten 1-Distanzen. Alle Objekte der folgenden Intervalle haben dementsprechend größere 1-Distanzen. Um unter diesen Bedingungen den repräsentativen dichten Bereich des Clusters vom Rauschen zu trennen, suchen wir nach dem größten Abfall in der Objektanzahl zwischen zwei Intervallen (dargestellt in Abbildung 9 durch Pfeile an den Histogrammen). Wir gehen davon aus, dass dieser maximale Abfall der Objektanzahl die Grenze zum Rauschen darstellt und dass die Menge der Objekte, die sich vor dieser Grenze befindet, repräsentativ für den gesamten C_{agg} ist.

Das Intervall, das zum größten Unterschied beigetragen hat, wird zur Berechnung von ϵ folgendermaßen herangezogen:

$$\epsilon = g \cdot \frac{dist^1_{max} - dist^1_{min}}{j} + dist^1_{min}. \qquad (6)$$

Dabei entspricht g der Nummer des gewählten Intervalls, j entspricht der Anzahl an Intervallen des Histograms, $dist^1_{min}$ und $dist^1_{max}$ stehen für die minimale und maximale 1-Distanz. Mit dem auf diese Weise berechneten ϵ und $MinPts = 2$ könnte nun ein DBSCAN-Clustering durchgeführt werden. Die Qualität dieses Clusterings ist nicht ausreichend, da zwar der aktuelle C_{agg} in summa gefunden wird, aber nicht unbedingt zusammenhängend ist.

Zur Qualitätssteigerung konnten bessere Heuristiken für $MinPts$ und ϵ während der Entwicklung des Ansatzes abgeleitet werden:

$$MinPts' = \sum_{x=1}^{g} elem(I_x). \qquad (7)$$

Dabei steht $elem(I_x)$ im Histogramm der 1-Distanzen von C_{agg} für die Objektanzahl des x-ten Intervalls. Durch g wird die Nummer des gewählten Histogramms dargestellt. Nachdem ein neues $MinPts'$ bestimmt worden ist, muß ein angepasster ϵ'-Wert bestimmt werden, so dass der durch diesen Parameter definierte Raum groß genug ist, um die

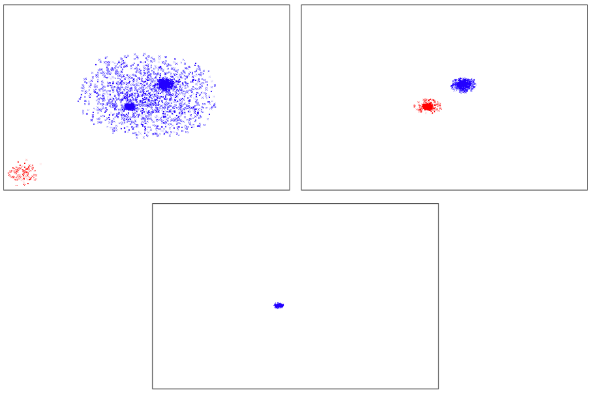


Abbildung 10: DBSCAN-Ergebnisse

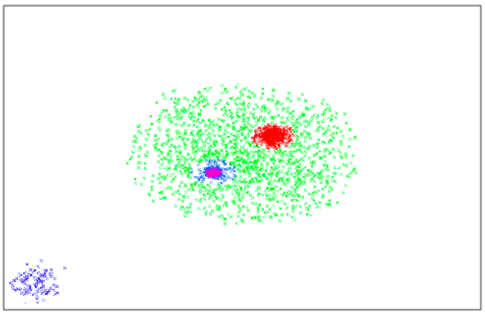
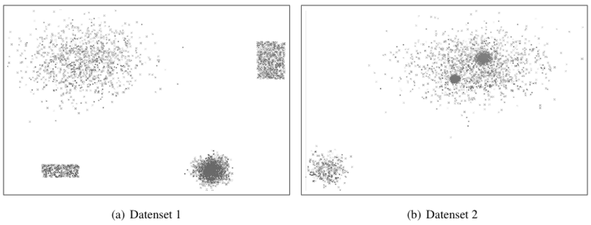


Abbildung 11: Endergebnis des mehrstufigen Ansatzes



(a) Datenset 1

(b) Datenset 2

Abbildung 12: Synthetische Datenmenge der Evaluierung

erforderliche Anzahl von Objekten aufzunehmen.

$$\epsilon' = \epsilon \cdot \sqrt[n]{MinPts'} \tag{8}$$

Die Variable n entspricht dabei der Anzahl der Dimensionen des Datenraums. Diese angepassten Werte lieferten in unseren Experimenten immer gute Ergebnisse. Diese Art der Parameterbestimmung muss aber noch genauer untersucht werden.

DBSCAN-Clustering
Für jeden erkannten Level-2-Cluster wird ein Parameterpaar ϵ und $MinPts$ berechnet und ein DBSCAN-Clustering durchgeführt, d.h. die Anzahl der Clusterings in Level 3 hängt direkt von der Anzahl der erkannten Level-2-Cluster ab. Für unser Beispiel würde das in 8 DBSCAN-Ausführungen münden. Um die Anzahl der Ausführungen zu reduzieren, haben wir ein Verfahren zur Parameterzusammenfassung entwickelt, um redundante DBSCAN-Ausführungen mit ähnlichen Werten zu vermeiden. Die durch diese Parameterzusammenfassung entstandenen Clusterings werden in Abbildung 10 dargestellt.

3.4 Finale Aggregation

Zu Beginn dieser Operation liegen uns die mit Hilfe der zusammengefassten Parameter erstellten DBSCAN-Clusterings vor. Betrachten wir unser Beispiel in Abbildung 10 etwas genauer, so stellen wir fest, dass jedes dieser Clusterings einer bestimmten Dichte-Ebene innerhalb der Datenmenge entspricht. Die Existenz von Bereichen verschiedener Dichte innerhalb einer Datenmenge deutet auf hierarchische Strukturen hin. Das Ziel der finalen Aggregation ist es, alle Informationen, die uns bisher über die Cluster und ihre Hierarchie vorliegen, zu einem einzelnen Clustering zusammenzufassen. Diese Aggregation erledigen wir ebenfalls mit unserem Algorithmus zur Zusammenführung von verschiedenen Clusterings. Das Endergebnis für unser Beispiel ist in Abbildung 11 illustriert.

4 Evaluierung

In diesem Abschnitt bewerten wir unseren mehrstufigen Ansatz, wobei wir die in Abbildung 12 dargestellten synthetischen Datenmengen benutzen. Beide Datenmengen sind gleich groß und beinhalten jeweils vier Cluster. Während im Datensatz 1 keine Hierarchien vorkommen, ist das im Datensatz 2 durchaus der Fall.

Die beschriebenen Datenmengen sind mit dem erläuterten mehrstufigen Ansatz bearbeitet worden. Dabei wurden neben der Gesamtlaufzeit auch die Laufzeiten der einzelnen Komponenten gemessen. Für jede Datenmenge wurden vier Messreihen durchgeführt. Um den Einfluss der Anzahl der zu bearbeitenden Datenobjekte auf die Laufzeit beobachten zu können, wurde in jeder Messreihe die Größe des jeweiligen Datensatzes verdoppelt. Die Verdopplung wurde gleichmäßig durchgeführt, so dass die Strukturen innerhalb der Daten trotz der Vergrößerung der Datenmengen erhalten blieben. Für jeden Datensatz existiert jeweils eine Version mit 4.000, 8.000, 16.000 und 32.000 Objekten.

In Abbildung 13 wird dargestellt, wie sich die Gesamtlaufzeit des mehrstufigen Ansatzes mit Zunahme der Objektanzahl verändert, wobei von einer sequenziellen Verarbeitung ausgegangen wird. Die Messwerte liegen dabei in der Form Stunden:Minuten:Sekunden,Millisekunden (hh:mm:ss,ms) vor. Wie leicht zu erkennen ist, nimmt die Laufzeit mit der Größe der Datenmenge zu.

Wie in Abschnitt 3 beschrieben, werden innerhalb des mehrstufigen Ansatzes bestimmte Funktionen wie das Clustering mehrmals ausgeführt. Da teilweise keine Abhängigkeiten zwischen ihnen bestehen, können be-

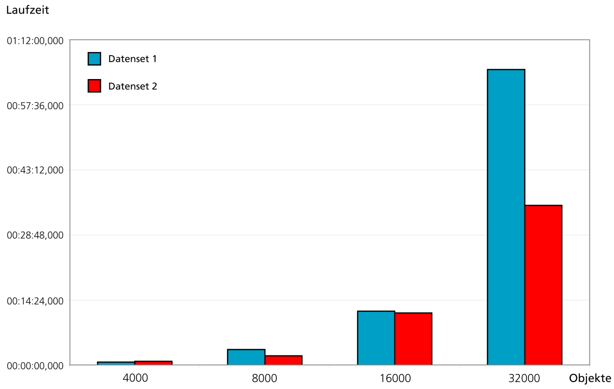


Abbildung 13: Sequenzielle Gesamtlaufzeiten für die synthetischen Datenmengen

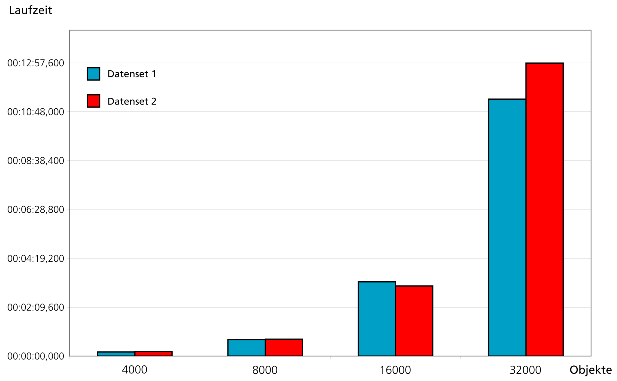
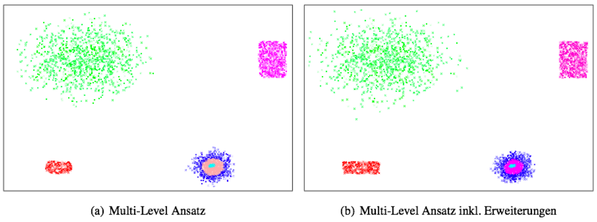


Abbildung 14: Parallele Gesamtlaufzeiten für die synthetischen Datenmengen


(a) Multi-Level Ansatz

(b) Multi-Level Ansatz inkl. Erweiterungen

Abbildung 15: Ergebnisse des mehrstufigen Ansatzes für die Datenmenge 1

stimmte Operationen parallel auf verschiedenen Systemen ausgeführt werden. Daraus würde sich gegenüber der sequenziellen Verarbeitung ein Vorteil im Hinblick auf die Laufzeit ergeben. Nachstehende Elemente des mehrstufigen Ansatzes sind parallelisierbar: *k-means-Ausführungen*, *Parameterbestimmung für jeden aggregierten Cluster in Stufe 2* und *DBSCAN-Ausführungen*. Alle anderen Operationen wie Gittererstellung, Parameterzusammenfassung und die Aggregationen können dagegen nicht verteilt ausgeführt werden. In Abbildung 14 sind die Gesamtlaufzeiten unseres mehrstufigen Ansatzes dargestellt. Als Ausführungsumgebung diente dabei unsere erweiterte SOA-Umgebung [6]. An den Rahmenbedingungen ist nichts geändert worden.

In den Abbildungen 11 und 15 sind die Ergebnisse unseres mehrstufigen Ansatzes dargestellt. Während der Berechnung des Ergebnisses mussten wir keine Parameter setzen.

5 Zusammenfassung und Ausblick

Das Ziel dieser Arbeit war es, einen mehrstufigen Algorithmus zu entwickeln, der parameterfreies Clustering ermöglicht. Um dieses Ziel zu erreichen, wurden bereits bekannte Clustering-Algorithmen und andere Verfahren aus dem Data-Mining-Bereich untersucht. Nach dieser Untersuchung wurden bestimmte Algorithmen, wie z.B. DBSCAN, ausgewählt und ganz oder teilweise in die einzelnen Stufen des entstehenden mehrstufigen Ansatzes übernommen. Weiterhin dienten bereits vorhandene Methoden und Verfahren als Ausgangsbasis für die Herleitung und Entwicklung neuer Funktionen. So wurden beispielsweise unterschiedliche Verfahren zur (semi-)automatischen Bestimmung diverser Parameter entwickelt. Durch die effiziente Integration dieser vorhandenen und neu entwickelten Algorithmen in einen mehrstufigen Prozess entstand unser Ansatz. Dieser vereint die Vorteile unterschiedlicher Clustering-Verfahren und erlaubt es, eine Datenmenge ohne Eingriff durch einen Nutzer zu partitionieren. Anhand der prototypischen Umsetzung konnten Eigenschaften wie Laufzeit und Ergebnisqualität bewertet werden. Für diese Evaluierung wurden im Moment nur synthetische Datenmengen herangezogen. Insgesamt kann festgehalten werden, dass unser mehrstufiger Ansatz ein Schritt in die Richtung eines parameterfreien Clusterings ist. Nichtsdestotrotz ist noch eine Menge Forschungsarbeit vonnöten.

Literaturverzeichnis

[1] E. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifcations. *Biometrics*, 21:768, 1965.

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96,Portland, Oregon, USA)*, pages 226–231, 1996.

[3] Martin Ester and Jörg Sander. *Knowledge Discovery in Databases- Techniken und Anwendungen*. Springer Verlag Berlin Heidelberg New York, 2000.

[4] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Knowledge discovery and data mining: Towards a unifying framework. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 96, Portland, Oregon, USA)*, pages 82–88, 1996.

[5] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. In *Proceedings of fifth Internation Conference on Data Mining (ICDM 2005, 27-30 November, Houstan, Texas, USA)*, 2005.

[6] Dirk Habich, Sebastian Richly, Wolfgang Lehner, Uwe Aßmann, Mike Grasselt, Albert Maier, and Christian Pilarsky. Data-aware soa for gene expression analysis processes. In *Proceedings of 2007 IEEE International Conference on Services Computing - Workshops (SCW 2007, 9-13 July 2007, Salt Lake City, Utah, USA)*, pages 138–145, 2007.

[7] Dirk Habich, Thomas Wächter, Wolfgang Lehner, and Christian Pilarsky. Two-phase clustering strategy for gene expression data sets. In *Proceedings of the 2006 ACM Symposium on Applied Computing - Bioinformatics Track (SAC 2006, Dijon, France, April 23-27)*, pages 145–150, 2006.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.

[9] David W. Scoot. On optimal and data-based histograms. *Biometrika 66*, 66 Nr.3:605–610, 1979.

[10] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Pearson Addison Wesley, 2006.

[11] Ian H. Witten and Eibe Frank. *Data Mining - Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, second edition edition, 2005.

Clustering using EM-based Hill-Climbing on Kernel Density Estimation

Alexander Hinneburg
Institute of Computer Science
Martin-Luther-University Halle-Wittenberg,
Germany
hinneburg@informatik.uni-halle.de

Hans-Henning Gabriel
Otto-von-Guericke-University Magdeburg,
and 101Tec Media Style GmbH,
Germany
Hans-Henning.Gabriel@web.de

Abstract

The DENCLUE algorithm employs a cluster model based on kernel density estimation. A cluster is defined by a local maximum of the estimated density function. Data points are assigned to clusters by hill climbing, i.e. points going to the same local maximum are put into the same cluster. A disadvantage of DENCLUE 1.0 is, that the used hill climbing may make unnecessary small steps in the beginning and never converges exactly to the maximum, it just comes close.

We introduce a new hill climbing procedure for Gaussian kernels, which adjusts the step size automatically at no extra costs. We prove that the procedure converges exactly towards a local maximum by reducing it to a special case of the expectation maximization algorithm. We show experimentally that the new procedure needs much less iterations and can be accelerated by sampling based methods with sacrificing only a small amount of accuracy.

1 Introduction

Clustering can be formulated in many different ways. Nonparametric methods are well suited for exploring clusters, because no generative model of the data is assumed. Instead, the probability density in the data space is directly estimated from data instances. Kernel density estimation [Silverman, 1986; Scott, 1992] is a principled way of doing that task. There are several clustering algorithms, which exploit the adaptive nature of a kernel density estimate. Examples are the algorithms by Schnell [Schnell, 1964] and Fukunaga [Fukunaga and Hostler, 1975] which use the gradient of the estimated density function. The algorithms are also described in the books by Bock [Bock, 1974] and Fukunaga [Fukunaga, 1990] respectively. The DENCLUE framework for clustering [Hinneburg and Keim, 1998; 2003] builds upon Schnells algorithm. There, clusters are defined by local maxima of the density estimate. Data points are assigned to local maxima by hill climbing. Those points which are assigned to the same local maximum are put into a single cluster.

However, the algorithms use directional information of the gradient only. The step size remains fixed throughout the hill climbing. This implies certain disadvantages, namely the hill climbing does not converges towards the local maximum, it just comes close, and the number of iteration steps may be large due to many unnecessary small steps in the beginning. The step size could be heuristically adjusted by probing the density function at several positions in the direction of the gradient. As the computation of the density function is relatively costly, such a method involves extra costs for step size adjustment, which are not guaranteed to be compensated by less iterations.

The contribution of this article is a new hill climbing method for kernel density estimates with Gaussian kernels. The new method adjusts the step size automatically at no additional costs and converges towards a local maximum. We prove this by casting the hill climbing as a special case of the expectation maximization algorithm. Depending on the convergence criterium, the new method needs less iterations as fixed step size methods. Since the new hill climbing can be seen as an EM algorithm, general acceleration methods for EM, like sparse EM [Neal and Hinton, 1999] can be used as well. We also explore acceleration by sampling. Fast Density estimation [Zhang *et al.*, 1999] can be combined with our method as well but is not tested in this first study.

Other density based clustering methods beside DENCLUE, which would benefit from the new hill climbing, have been proposed by Herbin et al [Herbin *et al.*, 2001]. Variants of density based clustering are DBSCAN [Sander *et al.*, 1997], OPTICS [Ankerst *et al.*, 1999], and followup versions, which, however, do not use a probabilistic framework. This lack of foundation prevents the application of our new method there.

Related approaches include fuzzy c-means [Bezdek, 1999], which optimized the location of cluster centers and uses membership functions in a similar way as kernel functions are used by DENCLUE. A subtle difference between fuzzy c-means and DENCLUE is, that in c-means the membership grades of a point belonging to a cluster are normalized, s.t. the weights of a single data point for all clusters sum to one. This additional restriction makes the clusters competing for data points. DENCLUE does not have such restriction. The mountain method [Yager and Filev, 1994] also uses similar membership grades as c-means. It finds clusters by first discretizing the data space into a grid, calculates for all grid vertices the mountain function (which is comparable to the density up to normalization) and determines the grid vertex with the maximal mountain function as the center of the dominant cluster. After effects of the dominant cluster on the mountain function are removed, the second dominant cluster is found. The method iterates until the heights of the clusters drop below a predefined percentage of the dominant cluster. As the number of grid vertices grow exponentially in high dimensional data spaces, the method is limited to low dimensional data. Niche clustering [Nasraoui and Krishnapuram, 2001] uses a non-normalized density function as fitness function for prototype-based clustering in a genetic algorithm. Data points with high density (larger than a threshold) are seen as core points, which are used to estimate scale parameters similar to the smoothing parameter h introduced in the next section.

The rest of the paper is structured as follows. In section 2, we briefly introduce the old DENCLUE framework and in section 3 we propose our new improvements for that framework. In section 4, we compare the old and the new hill climbing experimentally.

2 DENCLUE 1.0 framework for clustering

The DENCLUE framework [Hinneburg and Keim, 2003] builds on non-parametric methods, namely kernel density estimation. Non-parametric methods are not looking for optimal parameters of some model, but estimate desired quantities like the probability density of the data directly from the data instances. This allows a more direct definition of a clustering in contrast to ‘parametric methods, where a clustering corresponds to an optimal parameter setting of some high-dimensional function. In the DENCLUE framework, the probability density in the data space is estimated as a function of all data instances $\vec{x}_t \in X \subset \mathbb{R}^d, d \in \mathbb{N},\ t = 1, \ldots, N$. The influences of the data instances in the data space are modeled via a simple kernel function, e.g. the Gaussian kernel $K(\vec{u}) = (2\pi)^{-\frac{d}{2}} \cdot \exp\left[-\frac{\vec{u}^2}{2}\right]$. The sum of all kernels (with suitable normalization) gives an estimate of the probability at any point $\vec{x}$ in the data space $\hat{p}(\vec{x}) = 1/(Nh^d) \sum_{t=1}^{N} K\big(\vec{x}-\vec{x}_t/h\big)$. The estimate $\hat{p}(\vec{x})$ enjoys all properties like differentiability like the original kernel function. The quantity $h > 0$ specifies to what degree a data instance is smoothed over data space. When h is large, an instance stretches its influence up to more distant regions. When h is small, an instance effects only the local neighborhood. We illustrate the idea of kernel density estimation on one-dimensional data as shown in figure 1.

A clustering in the DENCLUE framework is defined by the local maxima of the estimated density function. A hill-climbing procedure is started for each data instance, which assigns the instance to a local maxima. In case of Gaussian kernels, the hill climbing is guided by the gradient of $\hat{p}(\vec{x})$, which takes the form

$$\nabla\hat{p}(\vec{x}) = \frac{1}{h^{d+2}N}\sum_{t=1}^{N} K\left(\frac{\vec{x}-\vec{x}_t}{h}\right)\cdot(\vec{x}_t-\vec{x}). \qquad (1)$$

The hill climbing procedure starts at a data point and iterates until the density does not grow anymore. The update formula of the iteration to proceed from $\vec{x}^{(l)}$ to $\vec{x}^{(l+1)}$ is

$$\vec{x}^{(l+1)} = \vec{x}^{(l)} + \delta\frac{\nabla\hat{p}(\vec{x}^{(l)})}{\|\nabla\hat{p}(\vec{x}^{(l)})\|_2}. \qquad (2)$$

The step size δ is a small positive number. In the end, those end points of the hill climbing iteration, which are closer than 2δ are considered, to belong to the same local maximum. Instances, which are assigned to the same local maximum, are put into the same cluster.

A practical problem of gradient based hill climbing in general is the adaptation of the step size. In other words, how far to follow the direction of the gradient? There are several general heuristics for this problem, which all need to calculate $\hat{p}(\vec{x})$ several times to decide a suitable step size.

In the presence of random noise in the data, the DENCLUE framework provides an extra parameter $\xi > 0$, which treats all points assigned to local maxima $\hat{\vec{x}}$ with $\hat{p}(\hat{\vec{x}}) < \xi$ as outliers. Figure 2 sketches the idea of a DENCLUE clustering.

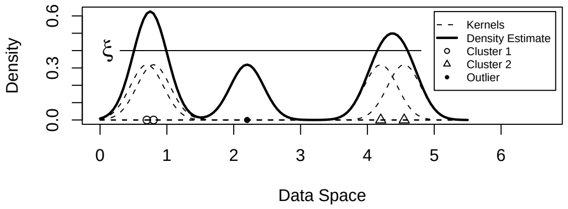


Figure 2: Example of a DENCLUE clustering based on a kernel density estimate and a noise threshold ξ.

3 DENCLUE 2.0

In this section, we propose significant improvements of the DENCLUE 1.0 framework for Gaussian kernels. Since the choice of the kernel type does not have large effects on the results in the typical case, the restriction on Gaussian kernels is not very serious. First, we introduce a new hill climbing procedure for Gaussian kernels, which adjust the step size automatically at no extra costs. The new method does really converge towards a local maximum. We prove this property by casting the hill climbing procedure as an instance of the expectation maximization algorithm. Last, we propose sampling based methods to accelerate the computation of the kernel density estimate.

3.1 Fast Hill Climbing

The goal of a hill climbing procedure is to maximize the density $\hat{p}(\vec{x})$. An alternative approach to gradient based hill climbing is to set the first derivative of $\hat{p}(\vec{x})$ to zero and solve for $\vec{x}$. Setting (1) to zero and rearranging we get

$$\vec{x} = \frac{\sum_{t=1}^{N} K\big(\frac{\vec{x}-\vec{x}_t}{h}\big)\vec{x}_t}{\sum_{t=1}^{N} K\big(\frac{\vec{x}-\vec{x}_t}{h}\big)} \qquad (3)$$

Obviously, this is not a solution for $\vec{x}$, since the vector is still involved into the righthand side. Since $\vec{x}$ influences the righthand side only through the kernel, the idea is to compute the kernel for some fixed $\vec{x}$ and update the vector on the lefthand side according to formula (3). This give a new iterative procedure with the update formula

$$\vec{x}^{(l+1)} = \frac{\sum_{t=1}^{N} K\big(\frac{\vec{x}^{(l)}-\vec{x}_t}{h}\big)\vec{x}_t}{\sum_{t=1}^{N} K\big(\frac{\vec{x}^{(l)}-\vec{x}_t}{h}\big)} \qquad (4)$$

The update formula can be interpreted as a normalized and weighted average of the data points and the weights of the data points depend on the influence of their kernels on the current $\vec{x}^{(l)}$. In order to see that the new update formula makes sense it is interesting to look at the special case $N = 1$. In that case, the estimated density function consists just of a single kernel and the iteration jumps after one step to $\vec{x}^1$, which is the maximum.

The behavior of DENCLUEs 1.0 hill climbing and the new hill climbing procedure is illustrated in figure 3. The figure shows that the step size of the new procedure is adjusted to the shape of the density function. On the other

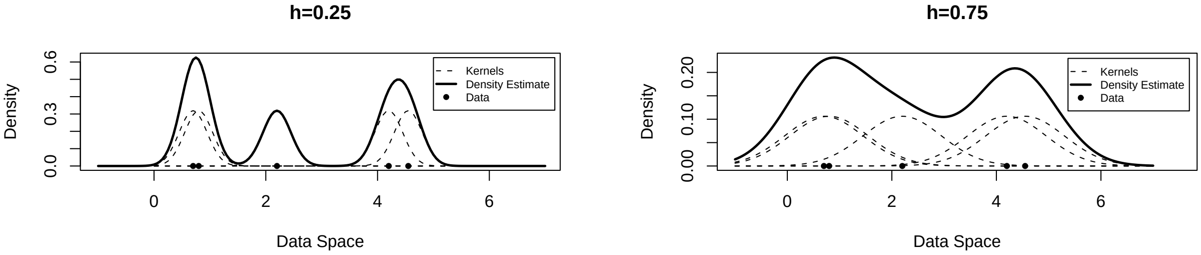


Figure 1: Kernel density estimate for one-dimensional data and different values for the smoothing parameter h.

hand, an iteration of the new procedure has the same computational costs as one of the old gradient based hill climbing. So, adjusting the step size comes at no additional costs. Another difference is, that the hill climbing of the new method really converges towards a local maximum, while the old method just comes close.

Since the new method does not need the step size parameter δ, the assignment of the instances to clusters is done in a new way. The problem is to define a heuristic, which automatically adjusts to the scale of distance between the converged points.

A hill climbing is started at each data point $\vec{x}_t \in X$ and iterates until the density does not change much, i.e. $[\hat{f}(\vec{x}_t^{(l)})-\hat{f}(\vec{x}_t^{(l-1)})]/\hat{f}(\vec{x}_t^{(l)}) \leq \epsilon$. An end point reached by the hill climbing is denoted by $\vec{x}_t^* = \vec{x}_t^{(l)}$ and the sum of the k last step sizes is $s_t = \sum_{i=1}^{k} \|\vec{x}_t^{(l-i+1)} - \vec{x}_t^{(l-i)}\|_2$. The integer k is parameter of the heuristic. We found that $k = 2$ worked well for all experiments. Note, that the number of iterations may vary between the data points, however, we restricted the number of iterations to be larger than k. For appropriate $\epsilon > 0$, it is safe to assume that the end points $\vec{x}_t^*$ are close to the respective local maxima. Typically, the step sizes are strongly shrinking before the convergence criterium is met. Therefore, we assume that the true local maximum is within a ball around $\vec{x}_t^*$ of radius s_t. Thus, the points belonging to the same local maximum have end points $\vec{x}_t^*$ and $\vec{x}_{t'}^*$, which are closer than $s_t + s_{t'}$. Figure 4 left illustrates that case.

However, there might exists rare cases, when such an assignment is not unique. This happens when for three end points $\vec{x}_t^*$, $\vec{x}_{t'}^*$ and $\vec{x}_{t''}^*$ hold the following conditions $\|\vec{x}_t^* - \vec{x}_{t'}^*\| \leq s_t + s_{t'}$ and $\|\vec{x}_t^* - \vec{x}_{t''}^*\| \leq s_t + s_{t''}$ but not $\|\vec{x}_{t'}^* - \vec{x}_{t''}^*\| \leq s_{t'} + s_{t''}$. In order to solve the problem, the hill climbing is continued for all points, which are involved in such situations, until the convergence criterium is met for some smaller ϵ (a simple way to reduce ϵ is multiply it with a constant between zero and one). After convergence is reached again, the ambiguous cases are rechecked. The hill climbing is continued until all such cases are solved. Since further iterations causes the step sizes to shrink the procedure will stop at some point. The idea is illustrated in figure 4 right.

However, until now it is not clear why the new hill climbing procedure converges towards a local maximum. In the next section, we prove this claim.

3.2 Reduction to Expectation Maximization

We prove the convergence of the new hill climbing method by casting the maximization of the density function as a special case of the expectation maximization framework [McLachlan and Krishnan, 1997]. When using the Gaussian kernel we can rewrite the kernel density estimate $\hat{p}(\vec{x})$ in the form of a constrained mixture model with Gaussian components

$$p(\vec{x}|\vec{\mu},\sigma) = \sum_{t=1}^{N} \pi_t \mathcal{N}(\vec{x}|\vec{\mu}_t,\sigma) \tag{5}$$

and the constraints $\pi_t = 1/N$, $\vec{\mu}_t = \vec{x}_t$ ($\vec{\mu}$ denotes a vector consisting of all concatenated $\vec{\mu}_t$), and $\sigma = h$. We can think of $p(\vec{x}|\vec{\mu},\sigma)$ as a likelihood of $\vec{x}$ given the model determined by $\vec{\mu}$ and σ. Maximizing $\log p(\vec{x}|\vec{\mu},\sigma)$ wrt. $\vec{x}$ is not possible in a direct way. Therefore, we resort to the EM framework by introducing a hidden bit variable $\vec{z} \in \{0,1\}^N$ with $\sum_{t=1}^{N} z_t = 1$ and

$$z_t = \begin{cases} 1 & \text{if the density at } \vec{x} \text{ is explained by } \mathcal{N}(\vec{x}|\vec{\mu}_t,\sigma) \text{ only} \\ 0 & \text{else} \end{cases} . \tag{6}$$

The complete log-likelihood is $\log p(\vec{x},\vec{z}|\vec{\mu},\sigma) = \log p(\vec{x}|\vec{z},\vec{\mu},\sigma)p(\vec{z})$ with $p(\vec{z}) = \prod_{t=1}^{N} \pi_t^{z_t}$ and $p(\vec{x}|\vec{z},\vec{\mu},\sigma) = \prod_{t=1}^{N} \mathcal{N}(\vec{x}|\vec{\mu}_t,\sigma)^{z_t}$.

In contrast to generative models, which use EM to determine parameters of the model, we maximize the complete likelihood wrt. $\vec{x}$. The EM-framework ensures that maximizing the complete log-likelihood maximizes the original log-likelihood as well. Therefore, we define the quantity

$$\mathcal{Q}(\vec{x}|\vec{x}^{(l)}) = E[\log p(\vec{x},\vec{z}|\vec{\mu},\sigma)|\vec{\mu},\sigma,\vec{x}^{(l)}] \tag{7}$$

In the E-step the expectation $\mathcal{Q}(\vec{x}|\vec{x}^{(l)})$ is computed wrt. to $\vec{z}$ and $\vec{x}^{(l)}$ is put for $\vec{x}$, while in the M-step $\mathcal{Q}(\vec{x}|\vec{x}^{(l)})$ is taken as a function of $\vec{x}$ and maximized. The E-step boils down to compute the posterior probability for the z_t:

$$E[z_t|\vec{\mu},\sigma,\vec{x}^{(l)}] = p(z_t = 1|\vec{x}^{(l)},\vec{\mu},\sigma) \tag{8}$$

$$= \frac{p(\vec{x}^{(l)}|z_t = 1,\vec{\mu},\sigma)p(z_t = 1|\vec{\mu},\sigma)}{\sum_{t=1}^{N} p(\vec{x}^{(l)}|z_t = 1,\vec{\mu},\sigma)p(z_t = 1|\vec{\mu},\sigma)} \tag{9}$$

$$= \frac{1/N \cdot \mathcal{N}(\vec{x}^{(l)}|\vec{\mu}_t,\sigma)}{\sum_{t'=1}^{N} 1/N \cdot \mathcal{N}(\vec{x}^{(l)}|\vec{\mu}_{t'},\sigma)} \tag{10}$$

$$= \frac{1/N \cdot K(\frac{\vec{x}^{(l)}-\vec{x}_t}{h})}{\hat{p}(\vec{x}^{(l)})} = \theta_t \tag{11}$$

In the M-step, z_t is replaced by the fixed posterior θ_t, which yields $\mathcal{Q}(\vec{x}|\vec{x}^{(l)}) = \sum_{t=1}^{N} \theta_t[\log 1/N + \log \mathcal{N}(\vec{x}|\vec{\mu}_t,\sigma)]$. Computing the derivative wrt. $\vec{x}$ and setting it to zero yields $\sum_{t=1}^{N} \theta_t \sigma^{-2}(\vec{x} - \vec{\mu}_t) = 0$ and thus

$$\vec{x}^{(l+1)} = \frac{\sum_{t=1}^{N} \theta_t \mu_t}{\sum_{t=1}^{N} \theta_t} = \frac{\sum_{t=1}^{N} K(\frac{\vec{x}^{(l)}-\vec{x}_t}{h})\vec{x}_t}{\sum_{t=1}^{N} K(\frac{\vec{x}^{(l)}-\vec{x}_t}{h})} \tag{12}$$

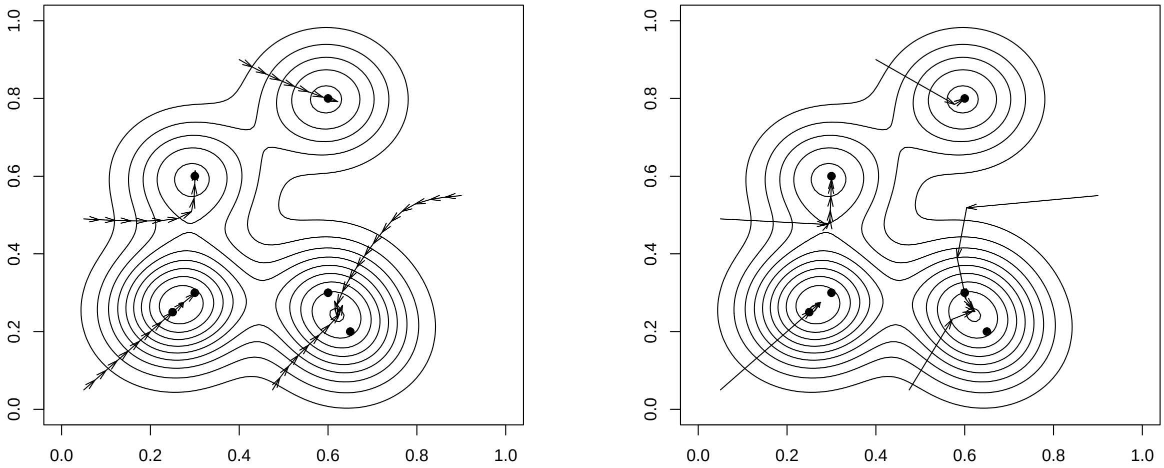


Figure 3: (left) Gradient hill climbing as used by DENCLUE 1.0, (right) Step size adjusting hill climbing used by DENCLUE 2.0.

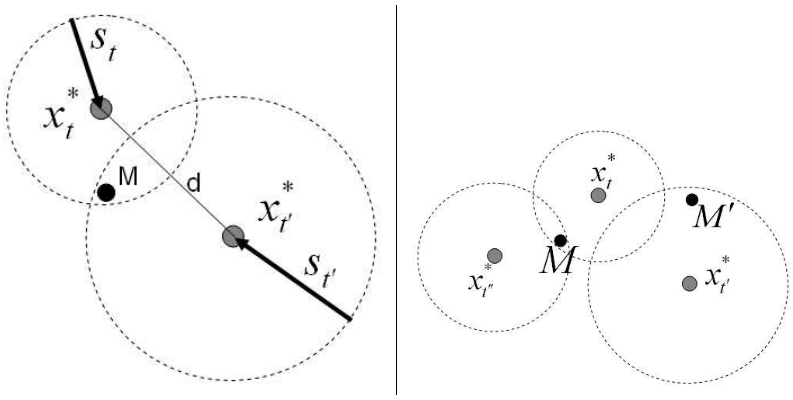


Figure 4: (left) Assignment to a local maximum, (right) Ambiguous assignment. The points M and M' denote the true but unknown local maxima.

By starting the EM with $\vec{x}^{(0)} = \vec{x}_t$ the method performs an iterative hill climbing starting at data point $\vec{x}_t$.

3.3 Sampling based Acceleration

As the hill climbing procedure is a special case of the expectation maximization algorithm, we can employ different general acceleration techniques known for EM to speed up the the DENCLUE clustering algorithm.

Most known methods for EM, try to reduce the number of iterations needed until convergence [McLachlan and Krishnan, 1997]. Since the number of iterations is typically quite low, that kind of techniques yield no significant reduction for the clustering algorithm.

In order to speed up the clustering algorithm, the costs for the iterations itself should be reduced. One option is sparse EM [Neal and Hinton, 1999], which still converges to the true local maxima. The idea is to freeze small posteriors for several iterations, so only the $p\%$ largest posteriors are updated in each iteration. As the hill climbing typically needs only a few iterations we modify the hill climbing starting at the single point $\vec{x}^{(0)}$ as follows. All kernels $K(\frac{\vec{x}^{(0)}-\vec{x}_t}{h})$ are determined in the initial iteration and $\vec{x}^{(1)}$ is determined as before. Let be U the index set of the $p\%$ largest kernels and L the complement. Then, in the next iterations the update formula is modified to

$$\vec{x}^{(l+1)} = \frac{\sum_{t\in U} K(\frac{\vec{x}^{(l)}-\vec{x}_t}{h})\vec{x}_t + \sum_{t\in L} K(\frac{\vec{x}^{(0)}-\vec{x}_t}{h})\vec{x}_t}{\sum_{t\in U} K(\frac{\vec{x}^{(l)}-\vec{x}_t}{h}) + \sum_{t\in L} K(\frac{\vec{x}^{(0)}-\vec{x}_t}{h})} \tag{13}$$

The index set U and L can be computed by sorting. The disadvantage of the method is, that the first iteration is still the same as in the original EM.

The original hill climbing converges towards a true local maximum of the density function. However, we does not need the exact position of such a maximum. It is sufficient for the clustering algorithm, that all points of a cluster converge to the same local maximum, regardless where that location might be. In that light, it makes sense to simplify the original density function by reducing the data set to a set of $p\%$ representative points. That reduction can be done in many ways. We consider here random sampling and k-means. So the number of points N is reduced to a much smaller number of representative points N', which are used to construct the density estimate.

Note that random sampling has much smaller costs as k-means. We investigate in the experimental section, whether

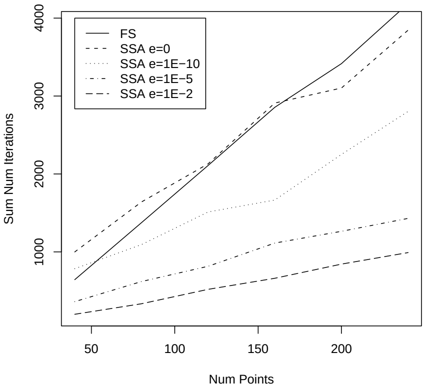


Figure 5: Number of data points versus the total sum of numbers of iterations.

the additional costs by k-means pay off by less needed iterations or by cluster quality.

4 Experimental Evaluation

We compared the new step size adjusting (SSA) hill climbing method with the old fixed step size hill climbing. We used synthetic data with normally distributed 16-dimensional clusters with uniformly distributed centers and approximately same size. Both methods are tuned to find the perfect clustering in the most efficient way. The total sum of numbers of iterations for the hill climbings of all data points is plotted versus the number of data points. SSA was run with different values for ϵ, which controls the convergence criterium of SSA. Figure 5 clearly shows that SSA ($\epsilon = 0.01$) needs only a fraction of the number of iterations of FS to achieve the same results. The costs per iterations are the same for both methods.

Next, we tested the influence of different sampling methods on the computational costs. Since the costs per iteration differ for sparse EM, we measure the costs in number of kernel computations versus sample size. Figure 6(left) shows that sparse EM is more expensive than random sampling and k-means based data reduction. The difference between the two latter methods is negligible, so the additional effort of k-means during the data reduction does not pay off in less computational costs during the hill climbing. For sample size 100% the methods converge to the original SSA hill climbing.

For random sampling, we tested sample size versus cluster quality measured by normalized mutual information (NMI is one if the perfect clustering is found). Figure 6(right) shows that the decrease of cluster quality is not linear in sample size. So, a sample of 20% is still sufficient for a good clustering when the dimensionality is $d = 16$. Larger dimensionality requires larger samples as well as more smoothing (larger h), but the clustering can still be found.

In the last experiment, we compared SSA, its sampling variants, and k-means with the optimal k on various real data sets from the machine learning repository wrt. cluster quality. Table 1 shows average values of NMI with standard deviation for k-means and sampling, but not for SSA which is a deterministic algorithm.

SSA has better or comparable cluster quality as k-means. The sampling variants degrade with smaller sample sizes $(0.8, 0.4, 0.2)$, but k-means based data reduction suffers much less from that effect. So, the additional effort of k-means based data reduction pays off in cluster quality.

In all experiments, the smoothing parameter h was tuned manually. Currently, we are working on methods to determine that parameter automatically.

5 Conclusion

In conclusion, we proposed a new hill climbing method for kernel density functions, which really converges towards a local maximum and adjusts the step size automatically. We believe, that our new technique has some potential for interesting combinations with parametric clustering methods.

References

[Ankerst *et al.*, 1999] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings SIGMOD'99*, pages 49–60. ACM Press, 1999.

[Bezdek, 1999] J.C. Bezdek. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Kluwer Academic Pub, 1999.

[Bock, 1974] H. H. Bock. *Automatic Classification*. Vandenhoeck and Ruprecht, 1974.

[Fukunaga and Hostler, 1975] K. Fukunaga and L.D. Hostler. The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Trans. Info. Thy.*, 21:32–40, 1975.

[Fukunaga, 1990] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[Herbin *et al.*, 2001] M. Herbin, N. Bonnet, and P. Vautrot. Estimation of the number of clusters and influence zones. *Pattern Recognition Letters*, 22:1557–1568, 2001.

[Hinneburg and Keim, 1998] A. Hinneburg and D.A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings KDD'98*, pages 58–65. AAAI Press, 1998.

[Hinneburg and Keim, 2003] Alexander Hinneburg and Daniel A. Keim. A general approach to clustering in large databases with noise. *Knowledge and Information Systems (KAIS)*, 5(4):387–415, 2003.

[McLachlan and Krishnan, 1997] Geoffrey J. McLachlan and Thiriyambakam Krishnan. *EM Algorithm and Extensions*. Wiley, 1997.

[Nasraoui and Krishnapuram, 2001] O. Nasraoui and R. Krishnapuram. The unsupervised niche clustering algorithm: extension tomultivariate clusters and application to color image segmentation. *IFSA World Congress and 20th NAFIPS International Conference,*, 3, 2001.

[Neal and Hinton, 1999] Radford M. Neal and Geoffrey E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. MIT Press, 1999.

[Sander *et al.*, 1997] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1997.

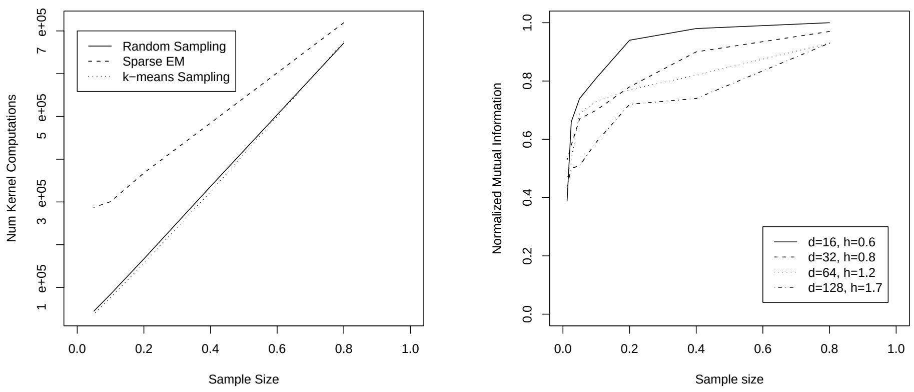


Figure 6: (left) Sample size versus number of kernel computations, (right) sample size versus cluster quality (normalized mutual information, NMI).

Table 1: NMI values for different data and methods, the first number in the three rightmost columns shows the sample size.

	k-means	**SSA**	**Random Sampling**	**Sparse EM**	**k-means Sampling**
iris	0.69±0.10	0.72	0.8: 0.66±0.05	0.8: 0.68±0.06	0.8: 0.67±0.06
			0.4: 0.63±0.05	0.4: 0.60±0.06	0.4: 0.65±0.07
			0.2: 0.63±0.06	0.2: 0.50±0.04	0.2: 0.64±0.07
ecoli	0.56±0.05	0.67	0.8: 0.65±0.02	0.8: 0.66±0.00	0.8: 0.65±0.02
			0.4: 0.62±0.06	0.4: 0.61±0.00	0.4: 0.65±0.04
			0.2: 0.59±0.06	0.2: 0.40±0.00	0.2: 0.65±0.03
wine	0.82±0.14	0.80	0.8: 0.71±0.06	0.8: 0.72±0.07	0.8: 0.70±0.11
			0.4: 0.63±0.10	0.4: 0.63±0.00	0.4: 0.70±0.05
			0.2: 0.55±0.15	0.2: 0.41±0.00	0.2: 0.58±0.21

[Schnell, 1964] P. Schnell. A method to find point-groups. *Biometrika*, 6:47–48, 1964.

[Scott, 1992] D.W. Scott. *Multivariate Density Estimation*. Wiley, 1992.

[Silverman, 1986] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[Yager and Filev, 1994] RR Yager and DP Filev. Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man and Cybernetics*, 24(8):1279–1284, 1994.

[Zhang *et al.*, 1999] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Fast density estimation using cf-kernel for very large databases. In *Proceedings KDD'99*, pages 312–316. ACM, 1999.

Conceptual Clustering of Social Bookmarking Sites

Miranda Grahl, Andreas Hotho, Gerd Stumme
Knowledge & Data Engineering Group, University of Kassel, Germany,
http://www.kde.cs.uni-kassel.de, {mgr, hotho, stumme}@cs.uni-kassel.de
Research Center L3S, Hannover, Germany, http://www.l3s.de

Abstract

Currently, social bookmarking systems provide intuitive support for browsing locally their content. A global view is usually presented by the tag cloud of the system, but it does not allow a conceptual drill-down, e. g., along a conceptual hierarchy. In this paper, we present a clustering approach for computing such a conceptual hierarchy for a given folksonomy. The hierarchy is complemented with ranked lists of users and resources most related to each cluster. The rankings are computed using our FolkRank algorithm. We have evaluated our approach on large scale data from the del.icio.us bookmarking system.

1 Introduction

This paper has been presented and published at I-Know 2007 [1] [Grahl *et al.*, 2007].

Social resource sharing systems are a way of collaboratively organising collections of resources, and are thus a promising alternative to classical knowledge management approaches -at least in domains where stronger structured approaches like ontologies could not take hold yet, or where their maintenance is too costly. This will hold especially in domains where people with no experience in data modelling have to deal with the tools.

The underlying structure of social resource sharing systems are so-called *folksonomies*, i. e., taxonomies created by the folk. A folksonomy consists of the *personomies* of its users. A personomy is the collection of all resources of a user, combined with a set of *tags*, which are catchwords that can be chosen arbitrarily by the users. Navigation in social resource sharing systems goes along hyperlinks which allow, for instance, to visit for a given tag, a web page listing all resources which have been tagged with this tag by at least one user. These systems allow thus for direct search of relevant entries. They also allow for serendipitous browsing, by following links to tags, users, and/or resources in a more or less random way. In a nutshell, folksonomy based systems are tuned for search and local navigation. With their tag clouds, social resource sharing systems also provide a simple mean to discover the overall content of their folksonomy. However, when the set of all tags becomes too large, one is looking for more structured ways of presenting the folksonomy's content.

In this paper, we present a conceptual, hierarchical clustering approach for folksonomies, and discuss its value for structuring of a folksonomy. First, we make iterative use of partitioning clustering (using two times the KMeans algorithm in our setting) on the set of tags. This step is followed by an application of our FolkRank algorithm [Hotho *et al.*, 2006] to discover resources and users that are related to the leaf clusters of the resulting cluster hierarchy, leading to the discovery of communities of interest. The generation of the cluster hierarchy is completely automatic, and may serve as input for a manual creation of a concept hierarchy (ontology) in a subsequent step. The result is a three-level hierarchy of sets of tags (see Fig. 1). The clusters on the lowest, most detailed level are complemented by lists of related resources and users.

In order to evaluate our approach, we have analyzed the large-scale popular social bookmarking system del.icio.us.[2] which is a simple-to-use interface that allows users to organize and share bookmarks on the internet.

State of the Art. The most similar feature to our approach that is implemented in del.icio.us is its tag cloud.[3]. It provides a first, global overview over the content of the system, but does not allow further conceptual drill-down into the data, since a click on one of the tags leads directly to an unstructured list of bookmarks.

Clustering of tags is one approach to support browsing and search within social bookmarking sites and is therefore of large interest. The scientific work that is most similar to ours is presented by [Begelman *et al.*, 2006]. A graph based clustering approach called Metis [Ayad and Kamel, 2003] is used on the weighted tag co-occurrence graph to find tag clusters in del.icio.us and RawSugar data.

Simpler clustering approaches are using only the weights of the tag co-occurrence network to split the graph into independent subgraphs. Every subgraph is then considered as a cluster [Mika, 2005]. The exploration of the network along the co-occurrence graph is discussed in the blog of Rashmi Sihna.[4] Simple probabilistic methods or association rule mining approaches are used to extract relation between tags in [Schmitz, 2006] and [Schmitz *et al.*, 2006]. A hierarchical clustering approach is applied on the weighted tag graph in [Heymann and Garcia-Molina, 2006] in order to compute a tag hierarchy. Similar clustering approaches are used to construct a hierarchy of tags for blogs in [Brooks and Montanez, 2006]. There are many more potential clustering approaches [Berkhin, 2002] (e. g., text clustering), but their applicability on folksonomy data still remains to be clarified.

[1] http://www.i-know.at

[2] http://del.icio.us

[3] http://del.icio.us/tag/

[4] http://www.rashmisinha.com/archives/05_02/tag-sorting.html

2 Dataset, Notations, and Algorithms

In the next section, we briefly present the used dataset for our experiment, and introduce some notations. Then we recall the basics of the clustering and the ranking algorithm that we used.

Dataset and Basic Notations. We have evaluated our approach on the social bookmarking system del.icio.us. Between July 27 and 30, 2005, we crawled del.icio.us and obtained a set U of 75,085 users, a set T of 456,666 tags, and a set R of 3,006,114 resources [Hotho *et al.*, 2006]. There were in total 7,281,940 posts, i. e., triples of the form (u, S, r), indicating that user $u \in U$ has assigned all tags contained in $S \subseteq T$ to resource $r \in R$. The set $Y \subseteq U \times T \times R$ of all tag assignments, i. e., of all (user, tag, resource) triples that show up in at least one post, consisted of 17,362,082 tag assignments

KMeans – a Clustering Algorithm. We used the well known cluster algorithm KMeans [Forgy, 1965] for our experiments as it provides in many cases good results. For KMeans, objects have to be represented in an n-dimensional vector space. As we will be working with a tag-tag-co-occurrence matrix [see Section 3], our objects will be tags, as well as each dimension (feature) of the vector space.

The principle of KMeans is as follows: Let k be the number of desired clusters. The algorithm starts by choosing randomly k data points of D as starting centroids and assigning each data point to the closest centroid (with respect to the given similarity measure; in our case the cosine measure). Then it (re-)calculates all cluster centroids and repeats the assignment to the closest centroid until no reassignment is performed. The result is a non-overlapping partitioning of the whole dataset into k clusters.

Each cluster is described by its centroid. Usually one considers only the top n features of each centroid, i.e. those n dimensions of the vector space which have the highest values in the vector. A large set of alternative clustering approaches exists [Berkhin, 2002]. To build a concept clustering hierarchy, an obvious solution would be to apply an hierarchical agglomerative clustering approach. But preliminary experiments showed that the distribution of the cluster size is strongly skewed, i.e. the majority of tags is assigned to one heterogenous cluster. KMeans, on the other hand, provided clusters with balanced sizes, which are more suitable to human perception.

Folkrank – a Ranking Algorithm. To compute the users and resources that are most related to clusters of tags, we use our Folkrank algorithm [Hotho *et al.*, 2006]. Given a set of preferred tags, users, and/or resources of a folksonomy, Folkrank computes a topic specific ranking which provides an ordering of the elements of the folksonomy in descending importance with respect to the preferred elements. Folkrank applies a two step approach to implement the weight-spreading ranking scheme on folksonomies. First, we transform the hypergraph between the sets of users, tags, and resources into an undirected, weighted, tripartite graph. On this graph A, we apply a version of PageRank [Brin and Page, 1998] that takes into account the edge weights. Our FolkRank algorithm computes a topic-specific ranking in a folksonomy by using a differential approach.

3 Constructing the Conceptual Hierarchy

For generating the conceptual hierarchy, we first removed, in a preprocessing step, some spam, and computed a vector space representation of the set of tags. Then we clustered the remaining set of tags, resulting in the lowest, most fine grained level of clusters. For each cluster, we extracted one tag as description. These descriptions were clustered again, yielding the middle layer of the conceptual hierarchy. Finally, we computed pairs of tags as descriptions of these 'meta-clusters', yielding the highest, most general level of the hierarchy. These steps are described in detail in the remainder of this section, together with the Folkrank based computation of sets of related users and resources for each cluster on the most fine grained level.

Data Preprocessing. In del.icio.us, posts with more than 50 tags are usually spam. As they strongly bias the co-occurrence network of tags, we removed all these posts in a first step. Then we computed, for each pair t_i, t_j of tags, their co-occurrence: $W(t_i, t_j) := |\{(u, r) \in U \times R \mid (u, t_i, r) \in Y \wedge (u, t_j, r) \in Y\}|$.

Each tag $t_i \in T$ is now represented in the $|T|$-dimensional vector space by the vector $\vec{t^i} := (\vec{t^i_j})_{j=1,\dots,|T|}$ with $\vec{t^i_j} := W(t_i, t_j)$ if $W(t_i, t_j) \geq 50$ and $i \neq j$, and $\vec{t^i_j} := 0$ else. The threshold of 50 turned out to be necessary to concentrate on the significant relationships between tags. Further experiments showed that lower thresholds resulted in clusters, contains a wide variation of unrelated tags.

We removed all tags that were represented by the $\vec{0}$ vector, as they are only peripheral to the folksonomy (see also [Cattuto *et al.*, 2007]). The set T was thus reduced to 6356 tags. The remaining 'core' tags were then clustered.

Iterated Clustering. The conceptual hierarchy is computed as follows from bottom to top.

(i) We cluster the set T of tags with k-Means with $k = 300$, resulting in a clustering $\mathbb{C} = \{C_1, \dots, C_{300}\}$ where each cluster contains 21.18 tags in average. (Five of these clusters are displayed completely at the lowest layer of Figure 1.) For each cluster C_i, we extracted from its centroid the tag $\hat{t}_i$ with the highest value as description of the cluster.[5] (The descriptors of 39 clusters are displayed in the middle layer of Figure 1.). The remaining tags of each centroid are excluded for the further computation. We denote the set of all descriptors by $\hat{T} := \{\hat{t}_i | i = 1, \dots, 300\}$. Note that $|\hat{T}| = 274$ instead of 300, as one descriptor can have the highest value in more than one centroid (e. g., the tag 'google').

(ii) While the set $\hat{T}$ provides already a good overview over the clusters computed above, it is still too large to be studied at a glance. Therefore, we clustered this set again with KMeans, this time with $k = 20$. We denote the result $\hat{\mathbb{C}} = \{\hat{C}_1, \dots, \hat{C}_{20}\}$. (Two of the resulting clusters are displayed at the middle layer of Figure 1.) Again, we extracted for each cluster a description from its centroid. This time, however, the most central tag in the centroid is not significant enough, as it contributes in average only 14.45% to the centroid. Therefore, we extracted the two most central tags from each centroid. (All 20 resulting tuples are shown in the top layer of Figure 1.) These tuples are a condensed summary of the current content of del.icio.us and enable

[5] Usually, one is taking more than one entry of the centroid as description, e. g., ten, but in our case, already the first tags turned out to be highly descriptive, contributing in average 67,41% to the centroid.

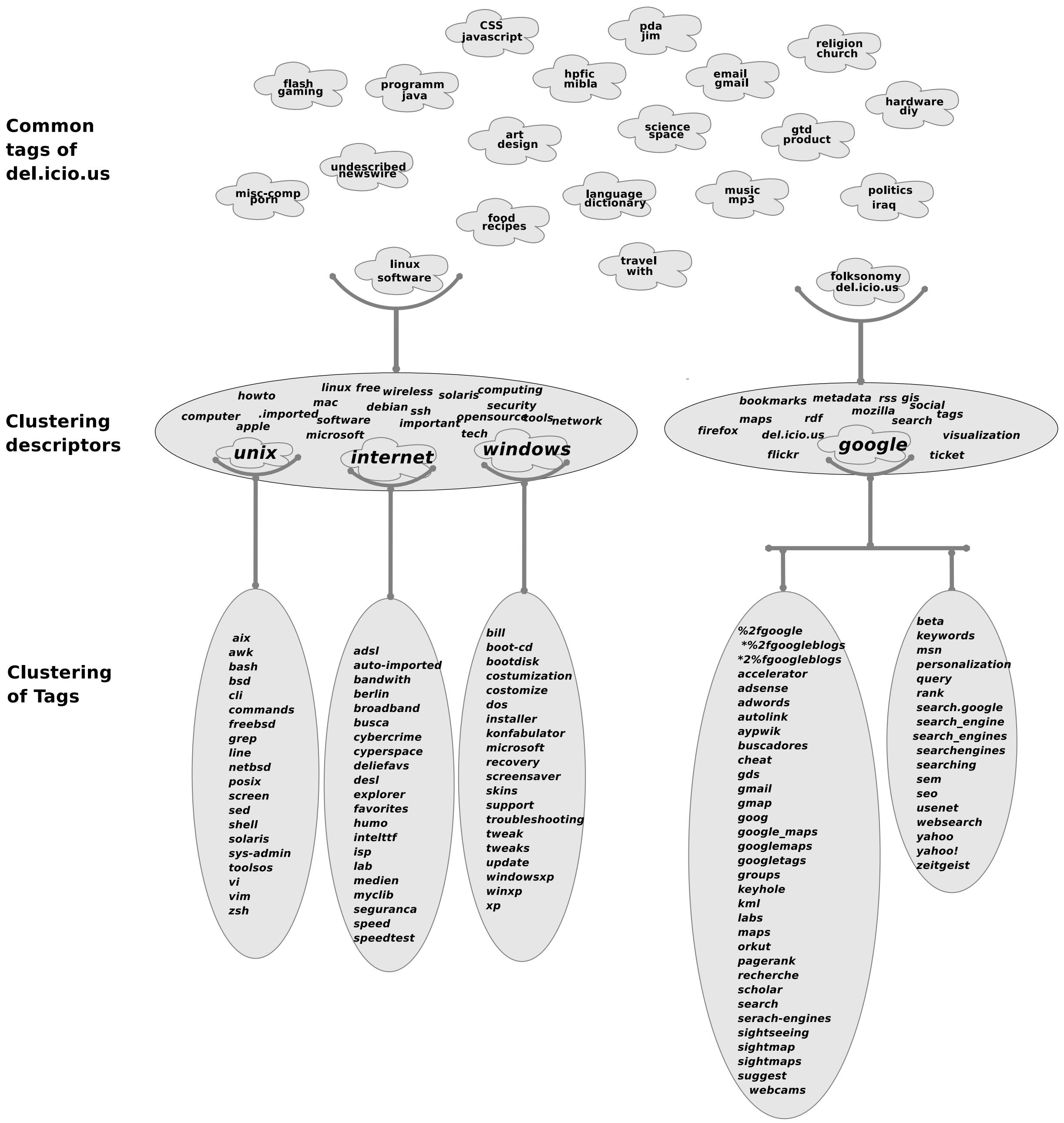


Figure 1: Conceptual hierarchy of the social bookmarking system del.icio.us.

the user to start a top-down navigation of the system.
The choice of $k = 300$ and $k = 20$, resp., results from the aim to obtain clusters with about 10-20 elements each. Slightly smaller/larger choices of k did not significantly affect the results; larger differences resulted in to small or too large clusters that were of no use to the user.

Computing Related Users and Resources. After having structured the set of tags in a conceptual hierarchy, we complement now the most fine grained level of tag clusters with those users and resources which are most related to each cluster. I. e., for each of the clusters $C_1, \ldots, C_{300}$, we compute a ranking of users and resources, resp., according to their relevance to the tags contained in the cluster. To this end, we have applied, for each cluster C_i, the FolkRank with $s = 0.85$ and with a preference vector $\vec{p}^{\,i}$ composed as follows: $\vec{p}^{\,i}_j := 1 + 10,000$ if tag $t_j \in C_i$ and 1 else.

4 Results

We have applied the process to the data of the del.icio.us system [6]. Part of the resulting hierarchy is shown in Fig. 1. From top to bottom, the hierarchy allows us to explore the folksonomy in more and more detail. The top layer of the hierarchy is displayed completely in Fig. 1. Its 20 pairs of tags provide a first overview over the content of del.icio.us, and are thus comparable to the tag cloud.[7] A main difference, however, is that the tag cloud of del.icio.us does not allow further drill-down along a conceptual hierarchy. When clicking on a tag in the tag cloud, del.icio.us directly presents all resources tagged with this tag. In our approach, we can navigate further down two more levels of the conceptual hierarchy.
For two selected entries, we have displayed the complete next level of the hierarchy. Let us first consider the subtree spanned by the tags ‘linux’ and ‘software’. The next level shows a cluster consisting of 23 tags, including the tags ‘linux’, ‘unix’ and ‘windows. The fact that only ‘linux’ made its way further up to the top level (because it had the largest contribution to the centroid of the cluster) indicates the preference of the del.icio.us users concerning operating systems.
For three of the 23 tags, we have also displayed the full next (and last) level of the hierarchy. We see for instance that the tag ‘unix’ on the intermediate level summarizes many unix commands and tools, such as ‘vi’ or ‘awk’. The tag ‘windows’ on the intermediate level represents a cluster which contains the microsoft operating systems ‘xp’ (in three variants) and ‘dos’, and issues like ‘custonmization’, ‘installer’ or ‘update’. The tag ‘internet’ on the intermediate level shows a wider variety of topics on the subsequent level, as might be expected.
The second branch of the hierarchy, spanned by del.icio.us, also provides some interesting insights. First, we observe that one tag in the subsequent layer is ‘google’, which, in contrast to the popularity of this search engine, did not make it to the top level of the hierarchy. This may again indicate a bias of the del.icio.us users. A second observation is more of a technical nature: We observe that ‘folksonomy’ is the most central component of the centroid of the rightmost cluster on the middle layer, even though the tag itself is not contained in the cluster itself. This effect results from the fact that we have set $\vec{t}^{\,i}_i = 0$ in the vector space representation. A third observation is that the tag ‘google’ in the intermediate layer leads to two different clusters in the fine grained layer. This results from the fact that both clusters displayed in the lower right of Fig. 1 have a centroid in which ‘google’ is the largest entry. In fact, both clusters are related to Google, but address different aspects. The rightmost cluster is about search engines in general, including ‘seo’ [= search engine optimization], and the web applications Yahoo! and Zeitgeist. The second cluster from the right consists mainly of Google services, like Google Maps or Google Blog Search. This cluster contains also, to a lesser extent, research related tags: ‘scholar’, ‘pagerank’. If one is interested in the users or resources that are related to a cluster, one can use the results of PageRank. For the two Google clusters, the rankings are shown in Tab. 1. We see for instance, that the del.icio.us user ‘ubi.quito.us’ is the most relevant contributor to the Google service cluster, and the second most relevant contributor (behind user ‘fritz’) to the search engine cluster. The top URL of the clusters are shown in Tab. 1. Our conceptual hierarchy leads us thus also to (implicit) communities of interest among the del.icio.us users, and to clusters of related web pages.

Table 1: Users and resources that are most related to the two Google clusters. The upper tables relates to the left and the lower to the right cluster.

rank	user
0.002	ubi.quito.us
0.002	kof2002
0.001	idealisms
6.4E-4	dajdump
3.1E-4	dymphna
2.6E-4	laugharne
2.5E-4	konno
2.5E-4	preoccupations
2.1E-4	josquin
2.1E-4	wxpbofh

rank	URL
2.6E-4	http://www.keyhole.com/kml/kml_tut.html
1.9E-4	http://www.googlesightseeing.com/
1.9E-4	http://scholar.google.com/
1.9E-4	http://webaccelerator.google.com/
1.8E-4	http://www.shreddies.org/gmaps/
1.8E-4	http://www.arnebrachhold.de/2005/06/05/google-sitemaps-generator-v2-final
1.7E-4	http://www.google.com/webhp?complete=1&amp;hl=en
1.6E-4	http://www.keyhole.com/kml/kml_doc.html
1.5E-4	http://serversideguy.blogspot.com/2004/12/google-suggest-dissected.html
1.3E-4	http://www.google.com/help/cheatsheet.html

rank	user
0.001	fritz
3.8E-4	ubi.quito.us
2.9E-4	kof2002
2.8E-4	triple_entendre
2.2E-4	cemper
1.7E-4	juanjoe
1.5E-4	konno
1.4E-4	tomohiromikami
1.2E-4	relephant
1.2E-4	masaka

rank	URL
1.5E-4	http://www.vaughns-1-pagers.com/internet/google-ranking-factors.htm
1.4E-4	http://www.google.com/press/zeitgeist.html
1.1E-4	http://www.philb.com/whichengine.htm
1.0E-4	http://inventory.overture.com/d/searchinventory/suggestion/
9.9E-5	http://www.google.com/
8.8E-5	http://www.buzzle.com/editorials/6-10-2005-71368.asp
7.8E-5	http://findory.com/
7.4E-5	http://www.betanews.com/
7.3E-5	http://clusty.com/
7.1E-5	http://cgi.cse.unsw.edu.au/ collabrank/del.icio.us/

[6] We also applied the approach to our BibSonomy system (http://www.bibsonomy.org). The results are not shown here due to space restrictions.

[7] http://del.icio.us/tag/

5 Summary and Conclusion

In this paper, we have shown a way of building a conceptual hierarchy on the set of tags of a folksonomy by using partitioning clustering algorithms. The leaves of the tag hierarchy have been extended with corresponding clusterings of the sets of resources and users, resp., to allow for accessing all dimensions of the folksonomy. Concerning the evaluation of the presented approach, it is difficult to find an objective measure of quality of clusters, as there does not exist any gold-standard for folksonomy clustering. This will be part of our future work.

Acknowledgement. This work has partially been supported by the European Commission within the research project 'TAGora – Emergent Semantics in Online Social Communities'.

References

[Ayad and Kamel, 2003] Hanan Ayad and Mohamed S. Kamel. Refined shared nearest neighbors graph for combining multiple data clusterings. In Michael R. Berthold, Hans-Joachim Lenz, Elizabeth Bradley, Rudolf Kruse, and Christian Borgelt, editors, *IDA*, volume 2810 of *Lecture Notes in Computer Science*, pages 307–318. Springer, 2003.

[Begelman *et al.*, 2006] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. *WWW2006, May*, pages 22–26, 2006.

[Berkhin, 2002] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[Brin and Page, 1998] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[Brooks and Montanez, 2006] Christopher H. Brooks and Nancy Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 625–632, New York, NY, USA, 2006. ACM Press.

[Cattuto *et al.*, 2007] Ciro Cattuto, Christoph Schmitz, Andre Baldassarri, Vito D. P. Servedio, Vittorio Loreto, Andreas Hotho, Miranda Grahl, and Gerd Stumme. Network properties of folksonomies. *AI Communications Special Issue on "Network Analysis in Natural Sciences and Engineering" (to appear)*, 2007.

[Forgy, 1965] E. Forgy. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 21(3):768–769, 1965.

[Grahl *et al.*, 2007] Miranda Grahl, Andreas Hotho, and Gerd Stumme. Conceptual clustering of social bookmarking sites. In *Proc. I-Know 2007 Conference (to appear)*, Graz, Austria, September 2007.

[Heymann and Garcia-Molina, 2006] P. Heymann and H. Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical report, Stanford InfoLab, 2006.

[Hotho *et al.*, 2006] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, June 2006. Springer.

[Mika, 2005] Peter Mika. Ontologies are us - a unified model of social networks and semantics. In *Proceedings of the 4th International Semantic Web Conference (ISWC)*, 2005.

[Schmitz *et al.*, 2006] Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. In *Proc. IFCS 2006 Conference*, Ljubljana, July 2006.

[Schmitz, 2006] Patrick Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, May 2006.

Integrating Genomic and Transcriptomic Data into Graph Based Approaches for Defining Essential Reactions in the Metabolic Network of *Escherichia Coli*

Kitiporn Plaimas^{1,2}, Marcus Oswald^{3}, Roland Eils^{1,2} and Rainer König^{1,2}

^{1}Theoretical Bioinformatics, German Cancer Research Center (DKFZ);
^{2}Department of Bioinformatics and Functional Genomics,
Institute of Pharmacy and Molecular Biotechnology;
^{3}Interdisciplinary Center for Scientific Computing,
University of Heidelberg, Germany;
k.plaimas@dkfz.de, marcus.oswald@informatik.uni-heidelberg.de,
r.eils@dkfz.de, r.koenig@dkfz.de

Abstract

The genomic data that presently is available enables an *in silico* representation of the metabolic network that can be enriched with a wide range of experimentally determined attributes. We integrated biochemical knowledge with genomic and transcriptomic data by establishing a simple machine learning tool to define the essentiality of single enzymes in the metabolic network of *Escherichia coli*. We collected characteristics of each single reaction based not only on network topology but also on genomic and transcriptomic characteristics. The metabolic network was investigated when blocking each single reaction, respectively. A reaction was marked to be essential when basically the mutated network could not deliver the products of the knocked out reaction from the upstream substrates of the knocked out reaction. Furthermore, we calculated characteristics of the network topology including choke points, clustering coefficients, comparisons to genome neighbourhoods and gene expression coregulation of the genes that encode the corresponding enzymes. We fed the data into a machine learning system and yielded an accuracy of about 90%, when comparing our results to an experimentally performed genome wide knock-out screen. This study showed that our tool can effectively determine the biochemical lethality of enzymes in metabolic networks. The method can readily be used for pathogenic organisms to find potential drug targets.

1 Introduction

Defining drug targets and drug design is one of the major goals in biomedical research. Especially metabolic enzymes had successfully been tackled by specific drugs to inhibit essential processes of pathogenic organisms in the human host (see e.g. [Hopkins and Groom, 2002]). Analysing the metabolic network *in silico* supports identifying essential genes and the corresponding enzymes/reactions that are considered to be crucial for the survival of the organism [e.g. Rahmann and Schomburg, 2006; Yeh et al., 2004]. A general model for the metabolic network consists of alternating nodes of reactions and metabolites. They have been descibed by graph theoretical approaches [Girvan and Newman, 2002; Jeong et al., 2000; Schuster et al., 2000]. Graph-based algorithms have been set up to identify drug targets in pathogenic organisms. The term 'damage' was defined by [Lemke et al., 2004; Mombach et al., 2006] to assess the enzymes that may serve as drug targets when their inhibition influences a substantial number of downstream metabolic reactions and products. The concept of choke points and load points was used to find enzymes which uniquely consume or produce a certain metabolite [Rahman and Schomburg, 2006; Yeh et al., 2004]. Furthermore, flux balance analyses (FBA) is a widely used and well established method to assess the essentiality of genes for an organism [Edwards and Palsson, 2000; Edwards et al., 2002; Kauffman et al., 2003; Becker et al., 2007; Feist et al., 2007]. However, FBA approaches need clear definitions of nutrition availability and biomass production under specifically given environmental conditions (for a good overview of these aspects see e.g. [Schuetz et al., 2007]). Different objectives (biomass productions) were proposed for different biological systems [Ebenhoh and Heinrich, 2001; Hermann-Georg, 2004; Price et al, 2004; Knorr et al, 2007].

High-throughput experiments have been performed to reveal the essentiality of all genes in an organism. For example, for *Escherichia coli*, the essentiality of virtually all open reading frames was observed by a single knock-out screen (KEIO collection, [Baba et al., 2006]). This data enables to test the performance of an *in silico* metabolic model that predicts essential genes. By using FBA under aerobic glucose condition in the COBRA tool [Becker et al., 2007], the newly reconstructed metabolic network of *E.coli* [Feist et al., 2007] has been used to predict gene essentiality yielding 92% accuracy when taking a combination of several media conditions into account (88% for the rich media condition in the KEIO collection).

In this paper we propose an alternative approach by feeding a machine learning system with qualitative and quantitative descriptors derived from graph topology, biochemical knowledge, and genomic and transcriptomic data. Note that our descriptors didn't use any further information about the environmental conditions and biomass production objectives. Using the KEIO collection [Baba et al., 2006] as the gold standard, we yielded an overall accuracy of 90.79% for rich media conditions.

2 System and Methods

2.1 Data representation

The metabolic network of *E. coli* was set up from the EcoCyc database [Ingrid et al., 2005; Karp et al., 2005] as has been described previously [König et al., 2006]. Basically, the metabolic network was represented as an undirected bipartite graph consisting of metabolites and reactions as alternating nodes. Unspecific compounds such as water, ATP, etc. were discarded. Finally the network contained 1024 metabolites and 1060 reactions.

2.2 The gold standard

In order to demonstrate the efficiency of our approach we used as the gold standard data from the KEIO collection [Baba et al., 2006]. The dataset consisted of the phenotypic outcomes from a set of single knock-out mutants to investigate effects of the loss of one gene. Genes were knocked out by in-frame replacement of a PCR product containing a kanamycin resistance gene. The start-codon and the up-stream translational signal were not replaced and fully intact. After kanamycin treatment, in-frame single gene deletions were verified by PCR with loci specific primers. When they were unable to create a mutant that formed colonies on a plate, the mutated gene was considered to be essential. Out of 4,288 tested genes, for 303 genes no mutants were found and therefore defined as being essential. Genes were mapped to the corresponding proteins and enzymes. If enzymes consisted of a complex of proteins, the corresponding reactions were defined to be essential or non-essential only if all coding genes gave a consistent image. Otherwise they were discarded from our training and testing analysis. Furthermore, a few number of reactions were discarded from the analysis if the corresponding genes couldn't be defined. Finally, from 303 essential genes we determined a set of 110 essential reactions. 846 reactions were identified as non-essential from the experimentally validated data and 104 reactions could not clearly be identified.

2.3 Defining the features

A list of relevant features was obtained from two main analyses: network topology and genomics data. Table 1 shows an overview of all features and their abbreviations.

Producibility of the products of the knocked out reaction

We set up a graph based algorithm analysing the structure of biochemical networks to investigate the network when blocking a single reaction. We defined a reaction as essential for survival when basically the mutated network could not yield the products of the reaction from upstream substrates of the reaction. Hence, features were defined, to describe if the knocked out reaction was substantial for producing its downstream metabolites or if these products could still be produced by other pathways. The investigation for each tested knocked out reaction was defined by the following algorithm.

i. Selection of biochemical compounds acting as input nodes (substrates) and output nodes (products). The set of substrates S of the knocked out reaction defined the input nodes that were available compounds for the network. Similarly, the set of products P of the knocked out reaction defined the output nodes to be produced. Additionally, we included the substrates of the upstream reactions and the products of the downstream reactions into the sets S and P, respectively.
ii. Identification of the reactions which used only available compounds as substrates.
iii. Incorporation of the identified reactions and their products into the network. These products were set as new available compounds in the network.
iv. Repetition of steps ii and iii until no further reactions could be identified for incorporation.
v. Evaluating and counting of the output nodes that could be produced (products P).

After finishing the process, we used the number of defined output nodes that can be produced within the mutated network for two features, i.e. a quality feature defining if ≥ 1 product could not be produced (MTP), and the percentage of products which could not be produced (PCP, see Table 1).

Deviations, choke and load points and damage

Our features describing the possible deviations (NDV, APL, LSP) and damaged compounds/reactions (Dc, Dr, DDc, DDr, Dcc, Dcr, DDcc, DDcr) were obtained from using a breath-first-search (BFS) algorithm. The deviation features were used to find alternative pathways to produce products of the knocked out reaction by its substrates S. In the metabolic network, these substrates can also be consumed by other reactions yielding their products etc. Therefore, we kept track of alternative paths in the metabolic network for the potential of the organism to survive when a reaction was blocked. The organism may have many pathways to produce the products. Thus, we counted the number of possible alternative paths (NDV) found by BFS from the set of substrates S to each product of a reaction. Starting from S, the breadth first search explored the network for finding the products of the knocked out reaction. When the algorithm visited the target products, it stored the corresponding pathway and continued its search to find further alternative paths until the network was entirely explored or a maximal path length of 10 reactions was reached. We took the average path length (APL) and the shortest path length (LSP) of the deviations as features for the classifier.

A reaction that uniquely consumes or produces a certain metabolite in the metabolic network is considered as a choke point in the metabolic network and show high potential of essentiality [Rahman and Schomburg, 2006; Yeh et al., 2004]. We checked if an observed reaction was a choke point (feature CKP).

We used the definitions of damaged compounds/reactions reported by [Lemke et al., 2004; Mombach et al., 2006]. Basically, they defined damage by determining the potentially effected metabolites and reactions downstream of the knocked out reaction. We applied their definition for calculating the features Dr, Dc (Table 1). However, some damaged compounds/reactions might have been produced from alternative pathways. Therefore, we calculated the number of damaged compounds/reactions that did not have an alternative way to be reached from the substrates of the knocked out reaction (DDr, DDc). In addition to our analysis on damage compounds/reactions, we also told the machine the number of damaged chokepoints (Dcc, Dcr, DDcc, DDcr).

Rahman and Schomburg proposed the definition of a load score for a reaction [Rahman and Schomburg, 2006]. In accordance to them, we defined the load score (LDP) as the ratio of the number of pathways passing through a reaction and the number of neighbouring reactions, compared to the average load value in the metabolic network. Note that the load value qualitatively estimates compound concentrations and fluxes of a reaction.

Gene expression data, genomic data and miscellaneous

For our case study, we collected raw intensity values of gene expression data from a rather metabolic-reaction-unspecific study observing the regulation during oxygen deprivation [Covert et al., 2004]. The gene expression data of each data-set was mapped onto the corresponding reactions as described in 2.1. For a reaction that was catalysed by a complex of proteins, we took the mean of the gene expression values for the corresponding genes (for more details, see [König et al., 2006]). From this, the maximum correlation coefficient of all neighbouring reactions (COR) and the number of reactions having similar gene expression (correlation coeficient > 0.8, EP2) were calculated. Together with the number of reactions from the same gene (EP1), these features served the machine for estimating if the knocked out reaction was in a biosynthesis or degradation pathway. Note that genes in the same pathway often show co-regulation [Samal et al., 2006].

The rest of the features are given in Table 1. We also included the number of homologous genes that might have taken over the function of the knocked out gene. Homologous genes were searched using Blast [Altschul et al., 1997] against all open reading frames of *E. coli* with three different e-value cutoffs, i.e. 10^{-3}, 10^{-5}, and 10^{-10} (H03, H05 and H010 in Table 1, respectively). To provide an estimate of the velocity of the knocked out reaction we used the Michaelis Menten constants (KMV) of the reaction, taken from the Brenda enzymes database [Barthelmes et al., 2007] and categorised these values into fast, normal and slow. For reactions that had no entry for the Km values, the category normal was taken. The reaction direction (DIR) was taken from Ecocyc and set as undirected if no information was available.

Finally, we calculated the clustering coefficient [Barabasi and Oltvai, 2004; Wagner and Fell, 2001] to describe the local network density of the knocked out reaction (CCO in Table 1).

Table 1 List of all features

Shortform	explanation
Network topology features	
MTP	more than or equal to one product cannot be produced when blocking the reaction
PCP	the percentage of products which cannot be produced when blocking the reaction
NPW*	the number of Ecocyc pathways the reaction is involved in
NED*	the number of substrates of the reaction
NPD*	the number of products of the reaction
COG	the number of corresponding genes
DIR*	the direction of the reaction
NDV*	the number of deviations
APL	the average path length of the deviations
LSP*	the length of the shortest path of the deviations
CKP	the reaction is a chokepoint or not (see [Rahmann and Schomburg, 2006])
LDP	load score of the reaction (see [Rahmann and Schomburg, 2006])
NBR*	the number of neighbouring reactions
NNR	the number of neighbours of neighbouring reactions
Dc*	the number of damage compounds
Dr*	the number of damage reactions
Dcc*	the number of damage chokepoint compounds after discarding the observed reaction
Dcr*	the number of damage chokepoint reactions
DDc*	the number of damage compounds that can not be produced from deviations
DDr*	the number of damage reactions that cannot be activated from deviations
DDcc*	the number of damage chokepoint compounds that cannot be produced from deviations
DDcr*	the number of damage chokepoint reactions that cannot be activated
CCO	clustering coefficient of the reaction
Genomics data features	
H10*	the number of homologous genes with e-value cutoff e-10
H05*	the number of homologous genes with e-value cutoff e-5
H03*	the number of homologous genes with e-value cutoff e-3
EP1^{+}	the number of reactions derived from the same genes
EP2$^{*,+}$	the number of reactions that have similar expression (correlation coefficient >0.8)
COR^{+}	maximum of the correlation coefficients for all neighbouring reactions
KMV	Michaelis Menten constant

* the optimised features, $^{+}$ features from gene expression

2.4 Machine learning

We applied the Support Vector Machine implementation of the R package e1071 [Dimitriadou et al., 2006] to classify between essential and non-essential reactions on the metabolic network. A radial basis function was used as the kernel function. Parameter optimisation was performed for the regularisation term that defined the costs for false classifications (5 steps for each, range: 2^n, n= -4, -2, 0, 2, 4). The same range was taken for the kernel width γ. This optimization was realised by a grid search over the parameter ranges [Dimitriadou et al., 2006]. The sizes of the two classes differed significantly in our data set (essential: 11%, non-essential: 89%). Therefore, we weighted positive instances by a factor range of 1 to 25. The machine with the best accuracy was selected as the best classifier and its features as the optimised feature set.

Feature selection

Feature selection was done by a top-down approach. We trained the Support Vector Machines in terms of maximising the overall accuracy using all features. Each single feature was discarded from the data set and the performance of the machine again calculated. Testing te performance of the machine was done by a leave-one-out cross validation. The accuracy of the machines missing one feature was compared and the best kept for the next iteration. This was repeated until one feature remained.

3 Results

Due to the small data set, we performed a leave-one-out cross validation to measure the effectiveness of our machine. We gained an overall accuracy of 90.06% when we took all features (Table 2). When we did the top-down approach, we got the best accuracy with the following 19 features: NPW, NED, NPD, DIR, NDV, LSP, NBR, Dc, Dr, Dcc, Dcr, DDc, DDr, DDcc, DDcr, H10, H05, H03, and EP2. Training the machine with these 19 features, we got an accuracy of 90.79%, mainly because of more true positives. This set of features was used for obtaining the following results.

Table 2 Results with and without optimised feature selection

	All features	best classifier (with 19 features)
True positive	16	27
False positive	1	5
True negative	845	841
False negative	94	83
Sensitivity	15%	25%
Specificity	100%	99%
Positive prediction value	94%	84%
Negative prediction value	90%	91%
Overall accuracy	90.06%	90.79%

To yield a broad spectrum of different precisions and sensitivities, we varied the weight factor for the positive instances from the data set with the optimised features in the range of 1 to 25. Figure 1 shows the performance of each classifier with these different weight factors. The sensitivity increased rapidly from a small weight to higher weights, reaching a plateau for weight factors of more than 10. This behaviour is expected: with a small weight the classifier tended to be overwhelmed by the large negative class. More positive instances were recognised when the weight factor was increased. When the weight factor reached more than 10, a sensitivity plateau was reached as no further positive instances could be reached. For the first data point (most left in Figure 1) representing the result with a weight factor of 1, we yielded the highest specificity (99%) and the best precision (84%) as shown in Table 2 by the best classifier result with the 19 features.

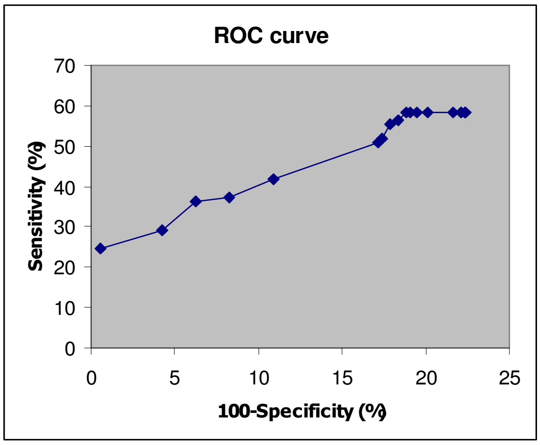


Figure1. Prediction results for the optimised feature set with different weight factors for the positive instances: each point represents the result for a different weight. From left to right, the weight was increased from 1 to 25.

From a recent analysis of flux balance in *E. coli*, the metabolic network was reconstructed with 2382 reactions and 1972 compounds. We performed a single reaction deletion on this network [Feist et al., 2007] and calculated flux values by FBA using the Cobra toolbox [Becker et al., 2007] to assess essential reactions under aerobic glucose conditions as described in the supplementary material of their article [Feist et al., 2007]. A reaction was assessed to be essential if the respective prediction of the mutated network's maximal biomass productions was < 1% of the wildtype's biomass productions. The biomass production was also taken as explained in [Feist et al., 2007]. Note that they didn't report *in silico* protocols for simulating rich media conditions.

From their network, out of 2382 reactions, 239 reactions were found to be essential in rich media according to the KEIO collection [Baba et al., 2006]. 1647 reactions were identified as non-essential for the experimental data and 496 reactions had no associated gene or could not clearly be identified. The FBA approach detected the essentiality of a reaction under aerobic glucose condition with an accuracy of 84.20%, a sensitivity of 46.44% and a specificity of 89.687% (calculated from 111 true positives, 170 false positives, 1477 true negatives and 128 false negatives). To ensure hypotheses for potential drug targets, the precision of the predictions is crucial. While the FBA got only 39.50% precision, our approach by using a weight of 1 got the precision of 84% while loosing 75% of positive instances. But, note that the media conditions of the FBA simulations couldn't be adjusted to the media conditions of the gold standard.

4 Discussion

Defining drug targets by modelling metabolic networks has been very successful with a broad variety of methods, especially with flux balance approaches. We have now set up a machine learning system that integrates features of the network topology and functional genomics properties in an elaborated way. We could show that the performance of our system is comparable and may even slightly outperform in some aspects flux balance approaches. Note that our system does not depend on setting up the correct environmental parameters such as nutrition and the set of output molecules defining the optimal biomass production. In contrast, our system uses high throughput data from functional genomics coming from microarray gene expression profiling and the genome sequence of the pathogenic organism. If such data has been raised or assembled from databases, our method may facilitate discovering potential drug targets, especially for pathogens for which the environmental parameters are changing and therefore are hard to define, as e.g. for intestinal infections. Note that our gene expression data was rather unspecific and did not account for a single mutant. To focus on relevant pathways we discarded highly connected compounds (like water, CO_2, ATP) from the network. However, the performance of the algorithm may be improved by a detailed enrolling of the metabolic pathways taking atom-mapping into account to discard further non-relevant paths (see [Rahman and Schomburg, 2006]). Furthermore, given the small test and training set, permutation tests could improve estimating the performance of the algorithm. In the future we hope to set up this system for a broad application discovering potential drug targets for a variety of substantial bacterial infections. A methodological challenge for this remains to transfer the method across organisms.

5 Acknowledgments

We like to thank the BMBF (Viroquant initiative), the Deutscher Akademischer Auslandsdienst, the Helmholtzgesellschaft (SB Cancer initiative) and the Commission on Higher Education (CHE), Thailand, for financial support.

6 References

[Altschul et al., 1997] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller and David J. Lipman: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25:3389-3402, 1997.

[Baba et al., 2006] Tomoya Baba, Takeshi Ara, Miki Hasegawa, Yuki Takai, Yoshiko Okumura, Miki Baba, Kirill A Datsenko, Masaru Tomita, Barry L Wanner, and Hirotada Mori. *Construction of Escherichia coli K-12 inframe, single-gene knockout mutants: the KEIO collection*, Molecular Systems Biology 2:2006.0008, February 2006.

[Barabasi and Oltyai, 2004] Albert-László Barabási and Zoltán N. Oltvai. *Network biology: understanding the cell's functional organization*, Nature Reviews Genetics 5:101-113, 2004.

[Barthelmes et al., 2007] Jens Barthelmes, Christian Ebeling, Antje Chang, Ida Schomburg, Dietmar Schomburg. *BRENDA, AMENDA and FRENDA: the enzyme information system in 2007*. Nucleic Acids Research 35 (Database-Issue): 511-514, 2007.

[Becker et al., 2007] Scott A Becker, Adam M Feist, Monica L Mo, Gregory Hannum, Bernhard Ø Palsson and Markus J Herrgard. *Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox*, Nature Protocols 2:727 – 738, 2007.

[Covert et al., 2004] Markus W. Covert, Eric M. Knight, Jennifer L. Reed, Markus J. Herrgard and Bernhard O. Palsson. *Integrating high-throughput and computational data elucidates bacterial networks.* Nature 429: 92-96, May 2004.

[Dimitriadou et al., 2006] Evgenia Dimitriadou, Kurt Hornik, Friedrich, Leisch, David Meyer, and Andreas Weingessel. *R-e1071: Misc Functions of the Department of Statistic (e1071)*, TU Wien, 2006.

[Ebenhöh et al., 2001] Oliver Ebenhöh and Reinhart Heinrich. *Evolutionary optimization of metabolic pathways. Theoretical reconstruction of the stoichiometry of ATP and NADH producing systems,* Bulletin of Mathematical Biology 63: 21–55, January 2001.

[Edwards and Palsson, 2000] Jeremy S Edwards and Bernhard O Palsson. *Metabolic flux balance analysis and the in silico analysis of Escherichia coli K-12 gene deletions.* BMC Bioinformatics, 1:1, July 2000.

[Edwards et al., 2002] Jeremy S Edwards, Markus W. Covert and Bernhard O Palsson. *Metabolic modelling of microbes: The flux-balance approach.* Environ Microbiol 4:133–133, 2002.

[Feist et al., 2007] Adam M Feist, Christopher S Henry, Jennifer L Reed, Markus Krummenacker, Andrew R Joyce1, Peter D Karp, Linda J Broadbelt, Vassily Hatzimanikatis and Bernhard Ø Palsson. *A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information*, Nature Molecular Systems Biology (121), June 2007.

[Girvan and Newman, 2002] Michelle Girvan and M. E. J. Newman. *Community structure in social and biological networks.* Proc. Natl Acad. Sci. USA, 99:7821–7826, 2002.

[Hermann-Georg, 2004] Holzhütter Hermann-Georg, *The principle of flux minimization and its application to estimate stationary fluxes in metabolic networks.* European journal of biochemistry, 271:2905–2922, 2004.

[Hopkins and Groom, 2002] *Andrew L. Hopkins and Colin R. Groom.* The druggable genome, Nat Rev Drug Discov., 1:727-30, 2002

[Ingrid et al., 2005] Ingrid M. Keseler, Julio Collado-Vides1, Socorro Gama-Castro1, John Ingraham, Suzanne Paley, Ian T. Paulsen, Mart1´n Peralta-Gil1 and Peter D. Karp: *EcoCyc: a comprehensive database resource for Escherichia coli*, Nucleic Acids Research, Vol. 33 (Database issue) , 2005.

[Jeong et al., 2000] Hyunseok Jeong, Bálint Tombor, Réka Albert, Zoltán N Oltvai, and Albert-László Barabási. *The large-scale organization of metabolic networks*. Nature, 407, 651–654, 2000.

[Kauffman et al., 2003] Kenneth J Kauffman, Purusharth Prakash and Jeremy S Edwards. *Advances in flux balance analysis.* Curr Opin Biotech 14:491–496, 2003.

[Karp et al., 2005] Peter D. Karp, Christos A. Ouzounis, Caroline Moore-Kochlacs, Leon Goldovsky, Pallavi Kaipa, Dag Ahrén, Sophia Tsoka, Nikos Darzentas, Victor Kunin and Núria López-Bigas. *Expansion of the BioCyc collection of pathway/genome databases to 160 genomes*, Nucleic Acids Research 19:6083-89 2005.

[Knorr et al., 2007] Andrea L. Knorr , Rishi Jain and Ranjan Srivastava. *Bayesian-based selection of metabolic objective functions*, Bioinformatics, 23:351–357, 2007.

[König et al., 2006] Rainer König, Gunnar Schramm, Marcus Oswald, Hanna Seitz, Sebastian Sager, Marc Zapatka, Gerhard Reinelt, Roland Eils. *Discovering functional gene expression patterns in the metabolic network of Escherichia coli with wavelets transforms.* BMC Bioinformatics, 7:119, 2006.

[Lemke et al., 2004] Ney Lemke, Fabiana Herédia, Cláudia K. Barcellos , Adriana N. dos Reis and José C. M. Mombach. *Essentiality and damage in metabolic networks,* Bioinformatics, Vol.20 no.1:115-119, 2004.

[Mombach et al., 2006] José C.M. Mombach, Ney Lemke, Norma M. da Silva, Rejane A. Ferreira, Eduardo Isaia, and Cláudia K Barcellos. *Bioinformatics analysis of mycoplasma metabolism: important enzymes, metabolic similarities, and redundancy*, Comput Biol Med. 36(5):542-52, May 2006.

[Price et al., 2004] Nathan D. Price, Jennifer L. Reed and Bernhard Ø. Palsson. *Genome-scale models of microbial cells: evaluating the consequences of constraints.* Nature Review Microbiology 2:886–897, 2004.

[Rahman and Schomburg, 2006] Syed Asad Rahman and Dietmar Schomburg. *Observing local and global properties of metabolic pathways: 'load points' and 'choke points' in the metabolic networks*, Bioinformatics, 22(14):1767-1774, 2006.

[Samal et al., 2006] Areejit Samal, Shalini Singh, Varun Giri, Sandeep Krishna, Nandula Raghuram, Sanjay Jain. *Low degree metabolites explain essential reactions and enhance modularity in biological networks*, BMC Bioinformatics, 7:118, 2006.

[Schuetz et al., 2007] Robert Schuetz, Lars Kuepfer,aa & Uwe Sauer, *Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli*, Nature Molecular Systems Biology (119), 2007.

[Schuster et al., 2000] Stefan Schuster, David A. Fell and Thomas Dandekar, *A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks.* Nature Biotechnology, 18:326–332, 2000.

[Wagner and Fell, 2001] Andreas Wagner and David Fell. The small world inside large Metabolic networks. In *Proceedings of The Royal Society, Biology Sciences*, 268(1478):1803-1810, September 2001.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Leraning Tools and Techniques, 2nd ed., 2005.

[Yeh et al., 2004] Iwei Yeh, Theodor Hanekamp, Sophia Tsoka, Peter D. Karp and Russ B. Altman. *Computational Analysis of Plasmodium falciparum Metabolism: Organizing Genomic Information to Facilitate Drug Discovery.* Genome Research 14:917-924, 2004

On the relevance of model orders to discriminative learning of Markov models

Jan Grau[1], **Jens Keilwagen**[2], **Ivo Grosse**[1,2], **and Stefan Posch**[1]
[1]Martin Luther University Halle–Wittenberg
[2]Leibniz Institute of Plant Genetics and Crop Plant Research (IPK) Gatersleben

Abstract

For a range of classification problems Markov models are used as the underlying statistical models. For generative learning of the Markov models and the performance achieved, the order of the Markov models is of importance. Here we investigate to which extend this fact holds for two discrimative approaches to learning. For two class problems we find the performance to depend solely on the maximum order using the maximum conditional likelihood, while this is not true for maximum supervised posterior in general.

1 Introduction

Many important questions in bioinformatics as well as in other application areas can be formulated as the task of classification which is to assign a label from a pre-defined set of discrete classes to an object. Examples from bioinformatics are the classification of plants into mutant and wild-type, the prediction of protein secondary structure, diagnosis of different cancer types, to partition DNA-sequences into coding and non-coding regions, and the detection of DNA-signals. In this paper, we concentrate on the last problem, although our theoretical results can be applied to other classification problems as well.

A broad range of techniques from machine learning consider the problem from a statistical point of view, where the sequences and classes observed are assumed to be the result of a stochastic process. In this context a prominent classifier is the Bayes classifier. The Bayes classifier assigns a sequence $\boldsymbol{x} \in A^L$ to the class $c \in C$ with the highest posterior class probability $P(c|\boldsymbol{x})$, i.e.

$$c^* = \operatorname*{argmax}_{c \in C} P(c|\boldsymbol{x}).$$

Many problems in bioinformatics as well as other fields of research reduce to two class problems where the task is to distinguish between a foreground class ($c = \mathrm{fg}$) and a background class ($c = \mathrm{bg}$). For DNA-signals such tasks may be to decide for a given sequence $\boldsymbol{x}$ if it is a transcription factor binding site or a DNA sequence without binding capability for the transcription factor considered. Another example is to discriminate between sequences that are involved in the splicing process and those that are not. For two class problems the decision of the Bayes classifier can be formulated as the following inequality:

$$\frac{P(\mathrm{fg}\,|\boldsymbol{x})}{P(\mathrm{bg}\,|\boldsymbol{x})} > 1$$

If the inequality holds for a sequence $\boldsymbol{x}$, it is assigned to the foreground class fg, otherwise to the background class bg. In the remainder of this paper we concentrate on two class problems with $C = \{\mathrm{fg}, \mathrm{bg}\}$.

With estimates of $P(c|\boldsymbol{x})$ available other types of classifiers can be realized as well, e.g. minimizing expected loss.

Within the generative approach these posterior class probabilities are determined from the class-conditional probabilities $P(\boldsymbol{x}|c)$ and the prior class probabilities $P(c)$ using Bayes' theorem. To infer the class-conditional probabilities, for each class a family of probability distributions is chosen and the corresponding parameters are estimated from data. This is carried out for each class individually, and often different families of probability distributions are chosen to appropriately model the statistical characteristics of each class. One popular family of distributions are those that can be represented by Bayesian networks [Heckerman *et al.*, 1994] and their specializations. From this family, Bayesian trees [Barash *et al.*, 2003] [Ben-Gal *et al.*, 2005] and Markov models [Salzberg, 1997] [Staden, 1984] [Stormo *et al.*, 1982] [Zhang and Marr, 1993] have been successfully applied to the problem of DNA-motif classification.

Bayesian networks belong to the class of graphical models, where the random variables emitting the symbols at each position are represented as nodes and potential dependencies between these random variables are represented as edges between the corresponding nodes in the graph. Markov models are a specialization of Bayesian networks where the distribution of the random variable at each position depends on the observations at a fixed number of directly preceeding positions. The number of preceeding positions considered is usually referred to as the *order* of the Markov model. The classification accuracy in the generative context usually strongly depends on the orders chosen. Hence it is worthwhile to consider varying combinations of orders in the different classes.

In this paper we investigate if and to what extend the choice of orders influences classification accuracy for discriminative approaches as well. In contrast to the generative approach, discriminative approaches directly infer the posterior class probabilities. This principle is more directly linked to the task of classification and the resulting classifiers have shown to be superior to generative approaches for a range of applications [Roos *et al.*, 2005] [Yakhnenko *et al.*, 2005] [Greiner *et al.*, 2005]. Choosing Markov models as the underlying families of probabilities we scrutinize the consequences of varying model orders for different classes on the classification results for maximum conditional likelihood and maximum supervised posterior estimation.

2 Maximum conditional likelihood

One discriminative approach is *maximum conditional likelihood*. It may be interpreted as the discriminative analogue of the widely-used maximum likelihood principle. Maximum conditional likelihood has been used for Bayesian network classifiers ([Greiner *et al.*, 2005], [Wettig *et al.*, 2002]), which include Markov models [Yakhnenko *et al.*, 2005] as a subfamily, and Markov random fields [Lafferty *et al.*, 2001], which are also called conditional random fields in this context.

We consider a data set $\boldsymbol{D} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$ of N sequences $\boldsymbol{x}_n \in A^L$, where A is the alphabet ($\{A, C, G, T\}$ for a DNA alphabet) and L denotes the length of the sequence $\boldsymbol{x}$. The vector $\boldsymbol{c} = (c_1, c_2, \ldots, c_N)$ denotes the correct classes of these sequences, i.e. c_n is the correct class of sequence $\boldsymbol{x}_n$. The maximum conditional likelihood estimate of parameters $\boldsymbol{\beta}$ is defined as

$$\hat{\boldsymbol{\beta}}^{\mathrm{MCL}} \quad = \quad \operatorname*{argmax}_{\boldsymbol{\beta}} P(\boldsymbol{c}|\boldsymbol{D}, \boldsymbol{\beta})$$

As we assume independent, identically distributed (i.i.d.) data points $(\boldsymbol{x}_n, c_n)$, the conditional likelihood factorizes into a product of the posterior class probabilities

$$P(\boldsymbol{c}|\boldsymbol{D}, \boldsymbol{\beta}) \quad = \quad \prod_{n=1}^{N} P(c_n|\boldsymbol{x}_n, \boldsymbol{\beta}),$$

which shows the relation of the maximum conditional likelihood approach to the classification criterion.

For convenience of formulation the posterior class probabilities are defined using functions $Q(c, \boldsymbol{x}; \boldsymbol{\beta}) > 0$ for each of the two classes:

$$P(c|\boldsymbol{x}, \boldsymbol{\beta}) \quad := \quad \frac{Q(c, \boldsymbol{x}; \boldsymbol{\beta})}{Q(c, \boldsymbol{x}; \boldsymbol{\beta}) + Q(\bar{c}, \boldsymbol{x}; \boldsymbol{\beta})} \qquad (1)$$

Here we denote $\bar{c}$ as the complement of class c, e.g. $\bar{\text{fg}} = bg$. This definition obviously defines proper probabilities for any function $Q(c, \boldsymbol{x}; \boldsymbol{\beta}) > 0$ as we have
$P(c|\boldsymbol{x}, \boldsymbol{\beta}) \geq 0$ and $\sum_{c \in C} P(c|\boldsymbol{x}, \boldsymbol{\beta}) = 1$.

In this paper, we use inhomogeneous Markov models of order d_c as the underlying statistical models. Accordingly, we define functions $Q_{\mathrm{MM}(\boldsymbol{d})}(c, \boldsymbol{x}; \boldsymbol{\beta})$ that model the Markov assumption, which is indicated by the subscript $\mathrm{MM}(\boldsymbol{d})$. With $\boldsymbol{d}$ we denote the vector of model orders for all classes and $d_{c,l} := min(d_c, l-1)$ is the length of the context considered at position l for model order d_c. We use the following functions to define the class probabilities:

$$\begin{aligned} Q_{\mathrm{MM}(\boldsymbol{d})}(c, \boldsymbol{x}; \boldsymbol{\beta}) &:= \exp\left(\beta_c\right) \prod_{l=1}^{L} \exp\left(\beta_{l, x_l|x_{l-1}\ldots x_{l-d_{c,l}}, c}\right) \\ &= \exp\left(\beta_c + \sum_{l=1}^{L} \beta_{l, x_l|x_{l-1}\ldots x_{l-d_{c,l}}, c}\right), \end{aligned} \qquad (2)$$

where $\beta_c \in \mathbb{R}$ and $\beta_{l, x_l|x_{l-1}\ldots x_{l-d_{c,l}}, c} \in \mathbb{R}$. The parameter $\beta_{l, x_l|x_{l-1}\ldots x_{l-d_{c,l}}, c}$ can be transformed to the probability of observing symbol x_l at position l in class c given the preceeding observations $x_{l-1} \ldots x_{l-d_{c,l}}$ and β_c can be transformed to to the probability of class c. For a Markov model of order 0 this transformation is given by

$$\begin{aligned} P(x_l|c, \boldsymbol{\beta}) &= \frac{\exp(\beta_{l,x_l|c})}{\sum_{a \in A} \exp(\beta_{l,a|c})} \\ P(c|\boldsymbol{\beta}) &= \frac{\exp(\beta_c) \sum_{\boldsymbol{x} \in A^L} \prod_{l=1}^{L} \exp(\beta_{l,x_l|c})}{\sum_{\tilde{c} \in C} \exp(\beta_{\tilde{c}}) \sum_{\boldsymbol{x} \in A^L} \prod_{l=1}^{L} \exp(\beta_{l,x_l|\tilde{c}})} \end{aligned}$$

For the relation between $\boldsymbol{\beta}$ and the conditional probabilities $P(x_l|x_{l-1} \ldots x_{l-d_{c,l}}, c, \boldsymbol{\beta})$ of higher order Markov models see e.g. [Wettig *et al.*, 2002],[Grau *et al.*, 2007]. We choose this parameterization of Markov models, because the conditional likelihood is known to be a concave function of $\boldsymbol{\beta}$ ([Wettig *et al.*, 2002], [Feelders and Ivanovs, 2006]).

Equation (1) can be rewritten as

$$P(c|\boldsymbol{x}, \boldsymbol{\beta}) \quad = \quad \frac{1}{1 + \frac{Q(\bar{c}, \boldsymbol{x}; \boldsymbol{\beta})}{Q(c, \boldsymbol{x}; \boldsymbol{\beta})}}. \qquad (3)$$

If we insert definition (2) of the function Q into (3), we obtain

$$\begin{aligned} P(c|\boldsymbol{x}, \boldsymbol{\beta}) &= \frac{1}{1 + \frac{\exp\left(\beta_{\bar{c}} + \sum_{l=1}^{L} \beta_{l, x_l|x_{l-1}\cdots x_{l-d_{\bar{c},l}}, \bar{c}}\right)}{\exp\left(\beta_c + \sum_{l=1}^{L} \beta_{l, x_l|x_{l-1}\cdots x_{l-d_{c,l}}, c}\right)}} \\ &= \frac{1}{1 + \exp(-\phi(\boldsymbol{x}, c; \boldsymbol{\beta}))}, \end{aligned} \qquad (4)$$

where

$$\begin{aligned} &\phi(\boldsymbol{x}, c; \boldsymbol{\beta}) := (\beta_c - \beta_{\bar{c}}) + \\ &\quad \sum_{\mathrm{l}=1}^{\mathrm{L}} \left(\beta_{\mathrm{l,x_l|x_{l-1}\ldots x_{l-d_{c,l}},c}} - \beta_{\mathrm{l,x_l|x_{l-1}\ldots x_{l-d_{\bar{c},l}},\bar{c}}}\right) \\ &= (-1)^{\delta(c=\mathrm{bg})} \left[(\beta_{\mathrm{fg}} - \beta_{\mathrm{bg}}) + \right. \\ &\quad \left. \sum_{\mathrm{l}=1}^{\mathrm{L}} \left(\beta_{\mathrm{l,x_l|x_{l-1}\ldots x_{l-d_{fg,l}},fg}} - \beta_{\mathrm{l,x_l|x_{l-1}\ldots x_{l-d_{bg,l}},bg}}\right) \right] \end{aligned} \qquad (5)$$

Noticing that $\phi(\boldsymbol{x}, c; \boldsymbol{\beta})$ is a linear function of the parameters $\boldsymbol{\beta}$, the functional form (4) reveals the equivalence of maximum conditional likelihood using Markov models to *logistic regression* ([Bishop, 2006]). The selection of the parameters from $\boldsymbol{\beta}$ contributing to the sum in (5) according to the observation $\boldsymbol{x}$ and class c may be viewed as the nonlinear transformation of $(\boldsymbol{x}, c)$ into an extended feature space.

Equation (5) also shows the ambiguity of this parameterization. We consider two sets of parameters $\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\beta}}$ and assume without loss of generality $d_{\mathrm{fg}} \geq d_{\mathrm{bg}}$. It is easy to see that these parameters yield the same posterior class probabilities, i.e. we have

$$\phi(\boldsymbol{x}, c; \boldsymbol{\beta}) = \phi(\boldsymbol{x}, c; \tilde{\boldsymbol{\beta}}),$$

if the following identities hold:

$$\begin{aligned} &\beta_{\mathrm{fg}} - \beta_{\mathrm{bg}} = \tilde{\beta}_{\mathrm{fg}} - \tilde{\beta}_{\mathrm{bg}} \\ &\beta_{l, x_l|x_{l-1}\ldots x_{l-d_{\mathrm{fg},l}}, \mathrm{fg}} - \beta_{l, x_l|x_{l-1}\ldots x_{l-d_{\mathrm{bg},l}}, \mathrm{bg}} \\ &\quad = \tilde{\beta}_{\mathrm{l,x_l|x_{l-1}\ldots x_{l-d_{fg,l}},fg}} - \tilde{\beta}_{\mathrm{l,x_l|x_{l-1}\ldots x_{l-d_{bg,l}},bg}} \end{aligned} \qquad (6)$$

For identical model orders $d_{\mathrm{fg}} = d_{\mathrm{bg}}$ this overparameterization is exploited in [Feelders and Ivanovs, 2006] to half the number of parameters with a new parameterization of Q:

$$\begin{aligned} &\phi(\boldsymbol{x}, c; \boldsymbol{\gamma}) = \\ &\quad (-1)^{\delta(\mathrm{c=bg})} \left[\gamma + \textstyle\sum_{\mathrm{l}=1}^{\mathrm{L}} \gamma_{\mathrm{l,x_l|x_{l-1}\ldots x_{l-d_{fg,l}}}} \right] \end{aligned} \qquad (7)$$

To further reduce the number of parameters we fix the non-free parameters β_{bg} and $\beta_{l,|A||\boldsymbol{b},c}$ to 0 [Grau *et al.*, 2007], which follows a proposition of Meila [Meila-Predoviciu, 1999].

Now we show that the set of posterior class probabilities described by (4) is determined by the maximum of the two model orders d_{fg} and d_{bg}.

Definition Let $\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}}$ be the set of all posterior class probabilities described by (4) based on Markov models of orders $(d_{\text{fg}}, d_{\text{bg}})$.

Definition A learning procedure for $\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}}$ has the *max-order property* iff for each pair $\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}}$ and $\mathcal{M}_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}}$ with $\max(d_{\text{fg}}, d_{\text{bg}}) = \max(\tilde{d}_{\text{fg}}, \tilde{d}_{\text{bg}})$ and all data sets it learns the same posterior class probability.

Lemma The maximum conditional likelihood estimate has max-order property.

Proof First we note, that trivially we have
$$\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}} \subseteq \mathcal{M}_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}} \text{ if } d_{\text{fg}} \leq \tilde{d}_{\text{fg}} \text{ and } d_{\text{bg}} \leq \tilde{d}_{\text{bg}}.$$
This obviously holds in analogy for Markov models in the generative setting using maximum likelihood.

Second we use (7) to state that
$$\mathcal{M}_{d,d} = \mathcal{M}_{d,0} = \mathcal{M}_{0,d}$$
which is not true in the generative setting.

Combining both and assuming $d_{\text{fg}} \geq d_{\text{bg}}$ we get
$$\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}} \subseteq \mathcal{M}_{d_{\text{fg}},d_{\text{fg}}} = \mathcal{M}_{d_{\text{fg}},0} \subseteq \mathcal{M}_{d_{\text{fg}},d_{\text{bg}}}$$
and thus $\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}} = \mathcal{M}_{d_{\text{fg}},d_{\text{fg}}}$. The same argument holds for $d_{\text{bg}} \geq d_{\text{fg}}$ which in combination gives
$$\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}} = \mathcal{M}_{d_{\max},d_{\max}} \text{ , with } \mathrm{d_{max}} := \max(\mathrm{d_{fg}}, \mathrm{d_{bg}})$$
□

The max-order property can also be generalized to more general classes of graphical models. If the structure of one of the models G_1 is completely contained in the structure of the other model G_2, i.e. G_2 represents at least all dependencies present in G_1, the power of the classifier is determined by the more complex model G_2.

To prepare the discussion in the next section we give the following

Definition A learning procedure for $\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}}$ has the *swap-order property* iff for each pair $\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}}$ and $\mathcal{M}_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}}$ with $(d_{\text{fg}}, d_{\text{bg}}) = (\tilde{d}_{\text{bg}}, \tilde{d}_{\text{fg}})$ and all data sets it learns the same posterior class probability.

Corollary The maximum conditional likelihood estimate has swap-order property.

In this case it is easy to give a transformation of parameters $\boldsymbol{\beta}$ in $\mathcal{M}_{d_{\text{bg}},d_{\text{fg}}}$ to parameters $\tilde{\boldsymbol{\beta}}$ in $\mathcal{M}_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}}$ yielding the same posterior class probability.
$$\begin{aligned} \tilde{\beta}_c &= \beta_c \\ \tilde{\beta}_{l,a|\boldsymbol{b},c} &= -\beta_{l,a|\boldsymbol{b},\bar{c}},\ a \in A, \boldsymbol{b} \in A^{\tilde{d}_{c,l}} = A^{d_{\bar{c},l}} \end{aligned}$$

As a consequence of the max-order property, the posterior class probabilities estimated by maximum conditional likelihood are determined by the maximum order of both Markov models and the resulting classifiers are identical. This is in strong contrast to generative learning using maximum likelihood.

3 Maximum supervised posterior

Another discriminative approach is *maximum supervised posterior* [Grünwald *et al.*, 2002] [Wettig *et al.*, 2002] [Cerquides and de Mántaras, 2005]
$$\hat{\boldsymbol{\beta}}^{\text{MSP}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} P(\boldsymbol{c}|\boldsymbol{D}, \boldsymbol{\beta}) P(\boldsymbol{\beta}|\boldsymbol{\alpha})$$
where $P(\boldsymbol{\beta}|\boldsymbol{\alpha})$ is a prior on the parameters $\boldsymbol{\beta}$ with hyper-parameters $\boldsymbol{\alpha}$. This definition of the maximum supervised posterior is in analogy to maximum a posteriori in the generative paradigm, where the likelihood $P(\boldsymbol{D}, \boldsymbol{c}|\boldsymbol{\beta})$ is multiplied by a prior as well. Maximum supervised posterior has been applied to data sets of the UCI machine learning repository [Roos *et al.*, 2005] and the recognition of transcription factor binding sites [Grau *et al.*, 2007].

3.1 Max-order and swap-order properties

In the following, we examine under which conditions the max-order and the swap-order property also hold for maximum supervised posterior employing Markov models for a two class problem. We associate with each family $\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}}$ a prior distribution $P_{d_{\text{fg}},d_{\text{bg}}}(\boldsymbol{\beta}|\boldsymbol{\alpha}_{d_{\text{fg}},d_{\text{bg}}})$ on its parameters $\boldsymbol{\beta}$ for a fixed $\boldsymbol{\alpha}_{d_{\text{fg}},d_{\text{bg}}}$. We consider a transformation T of parameters $\boldsymbol{\beta}$ in $\mathcal{M}_{d_{\text{fg}},d_{\text{bg}}}$ to parameters $\tilde{\boldsymbol{\beta}}$ in $\mathcal{M}_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}}$ as given explicitly for swap-order in (8). Since we already know from the last section that both properties hold for the conditional likelihood, a sufficient condition for both properties to hold also for maximum supervised posterior with the given prior is:
$$P_{d_{\text{fg}},d_{\text{bg}}}(\boldsymbol{\beta}|\boldsymbol{\alpha}_{d_{\text{fg}},d_{\text{bg}}}) = P_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}}(T(\boldsymbol{\beta})|\boldsymbol{\alpha}_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}}) \qquad (8)$$

In the following, we consider priors assuming independent parameters. Thus $P_{d_{\text{fg}},d_{\text{bg}}}(\boldsymbol{\beta}|\boldsymbol{\alpha}_{d_{\text{fg}},d_{\text{bg}}})$ decomposes into a product of univariate priors. The swap-order property holds for a prior if
$$\begin{aligned} &\forall l \in [1, L], \forall a \in A, \forall \boldsymbol{b} \in A^{d_{c,l}} = A^{\tilde{d}_{\bar{c},l}}, \forall c \in \{\text{fg}, \text{bg}\} \\ &\quad P_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}}(\tilde{\beta}_c = \nu|\boldsymbol{\alpha}_{\tilde{\boldsymbol{d}}_{\mathbf{fg}},\tilde{\boldsymbol{d}}_{\mathbf{bg}}}) \\ &\qquad = P_{d_{\text{fg}},d_{\text{bg}}}(\beta_c = \nu|\boldsymbol{\alpha}_{\boldsymbol{d}_{\mathbf{fg}},\boldsymbol{d}_{\mathbf{bg}}}) \\ &\quad P_{\tilde{d}_{\text{fg}},\tilde{d}_{\text{bg}}}(\tilde{\beta}_{l,a|\boldsymbol{b},\bar{c}} = -\nu|\boldsymbol{\alpha}_{\tilde{\boldsymbol{d}}_{\mathbf{fg}},\tilde{\boldsymbol{d}}_{\mathbf{bg}}}) \\ &\qquad = P_{d_{\text{fg}},d_{\text{bg}}}(\beta_{l,a|\boldsymbol{b},c} = \nu|\boldsymbol{\alpha}_{\boldsymbol{d}_{\mathbf{fg}},\boldsymbol{d}_{\mathbf{bg}}}) \quad . \end{aligned} \qquad (9)$$

A prior $P(\boldsymbol{\beta}|\boldsymbol{\alpha})$ has the max-order property, if it assigns a priori probabilities to the differences $\beta_{\text{fg}} - \beta_{\text{bg}}$ and $\beta_{l,a|\boldsymbol{b},\text{fg}} - \beta_{l,a|\boldsymbol{b}',\text{bg}}, a \in A, \boldsymbol{b} \in A^{d_{\text{fg},l}}, \boldsymbol{b}' \in A^{d_{\text{bg},l}}$ of (5). This is equivalent to using $\phi(\boldsymbol{x}, c|\boldsymbol{\gamma})$ instead of $\phi(\boldsymbol{x}, c|\boldsymbol{\beta})$ in (4) and defining a prior $P(\boldsymbol{\gamma}|\boldsymbol{\alpha})$ on the parameters $\boldsymbol{\gamma}$. The problem with such a prior is that a priori we do not have a good intuition for the density of the $\boldsymbol{\gamma}$-parameters. In contrast, we can derive some a priori assumptions about the $\boldsymbol{\beta}$-parameters from priors used in the generative context. In the generative context, a common a priori assumption is that all nucleotides at each position occur with the same probability. For the $\boldsymbol{\beta}$-parameters this amounts to assuming $\beta_{l,a|\boldsymbol{b},c} = 0$.

In this paper, we consider Gaussian priors with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ as hyper-parameters
$$P(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Sigma}). \qquad (10)$$
Since we assume independent parameters, this leads to a diagonal covariance matrix $\boldsymbol{\Sigma}$. The entries on the main diagonal are the a priori variances for each independent parameter. Hence we can define the Gaussian prior depending on $\boldsymbol{\mu}$ and a vector of variances $\boldsymbol{\sigma^2}$, $P(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\sigma^2})$.

In the following we investigate two different choices of mean vector and variances. In the first case, we choose $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\sigma^2} = \mathbf{1}$, i.e. the prior is a standard Gaussian density. Using these variances and means, the Gaussian prior satisfies the swap-order but generally not the max-order property.

In the second case, we choose μ_{fg} and σ^2_{fg} using a priori knowledge about the distribution of the classes, and

$\mu_{l,a|\boldsymbol{b},c} = 0$ and $\sigma_{l,a|\boldsymbol{b},c} = \kappa_c \cdot A^{d_{c,l}+1}$. The heuristic for the choice of $\sigma_{l,a|\boldsymbol{b},c}$ was proposed in [Grau *et al.*, 2007] and leads to a variance which is exponentially increasing with increasing local order $d_{c,l}$. With this choice of means and variances the Gaussian prior in general does satisfy neither the swap-order nor the max-order property, unless $\kappa_{\text{fg}} = \kappa_{\text{bg}}$. In this case the swap-order property holds.

In the following, we study to which degree these theoretical results have a relevant influence on real world classification problems. To this end, we consider combinations of Markov models of order $d_{\text{fg}} = 0$ to 2 and $d_{\text{bg}} = 0$ to 2 for the classification of transcription factor binding sites.

3.2 Experimental results

The foreground data set ($c = \text{fg}$) is a set of 149 transcription factor binding sites for the transcription factor C/EBP, which belongs to the class of *basic domain* factors. The data set stems from the TRANSFAC® database (rel. 8.1, 2004), which comprises experimentally verified transcription factor binding sites collected from the scientific literature. The background data set ($c = \text{bg}$) contains 267 sequences from second exons of human genes with $68,141$ bp in total.

In order to assess the classifiers, we use three measures of accuracy. The first measure is the area under the ROC curve (AUC), which judges the overall performance of a classifier. The second measure is the false positive rate ($\text{FPR} = 1 - \text{Sp} = \frac{FP}{TN+FP}$) for a fixed sensitivity of 95%. FPR measures the fraction of incorrectly classified background sequences if a classifier correctly predicts 95 out of 100 foreground sequences. The third measure is the sensitivity ($\text{Sn} = \frac{TP}{TP+FN}$) for a fixed specificity ($\text{Sp} = \frac{TN}{TN+FP}$) of 99.9%. Sn measures the fraction of correctly classified foreground sequences if a classifier erroneously predicts one out of 1000 background sequences to be a foreground sequence.

In order to estimate all three performance measures in a robust way together with their standard errors, we use a *k-fold stratified holdout sampling* procedure:

```
for k:=1 to K do
  1. Randomly partition the foreground and the background data set each into a training and a test set by drawing sequences from the original datasets such that the test set contains 10% of the nucleotides
  2. Train the classifier on the two training sets using all overlapping $L$-mers of the background sequences, where $L$ is the length of the foreground sequences
  3. Compute $r_{\boldsymbol{x}_n} = \frac{P(\text{fg}\,|\boldsymbol{x}_n,\boldsymbol{\beta})}{P(\text{bg}\,|\boldsymbol{x}_n,\boldsymbol{\beta})}$ for each sequence in the foreground test set and for each overlapping $L$-mer of the sequences in the background test set. Use the $r_{\boldsymbol{x}_n}$ to
     a) compute the area under curve (AUC),
     b) determine a threshold $T$ such that 95% of the foreground sequences are classified correctly, i.e. $r_{\boldsymbol{x}_n} > T$. This corresponds to fixing the sensitivity of the classifier to 0.95. Use this threshold to compute FPR on the $L$-mers of the background sequences.
     c) determine a threshold $T$ such that 99.9% of the $L$-mers of the background sequences are classified correctly, i.e. $r_{\boldsymbol{x}_n} \leq T$. This corresponds to fixing Sp of the classifier to 0.999. Use this threshold to compute Sn on the foreground sequences.
done
```

For each of the three measures the mean and standard error over the K iterations are computed. We use $K = 1000$. In the following, these are the values we use to assess the classifiers. We regard differences of more than twice the standard error as significant. The standard error for AUC is less than 0.0007, and for FPR and Sn it is less than 0.4.

For a prior satisfying the max-order property we expect the measures for $(d_{\text{fg}}, d_{\text{bg}}) = (0,1)$, $(1,1)$, $(1,0)$, and the measures for $(0,2)$, $(1,2)$, $(2,2)$, $(2,1)$, $(2,0)$ to vary only due to effects induced by random sampling. For a prior satisfying the swap-order property we expect the matrix of values to be symmetric except for deviations due to sampling.

The results for maximum supervised posterior with a standard Gaussian prior are presented in table 1. As this prior has the swap-order property we expect the table to be approximately symmetric. For AUC we find no deviation from the symmetry. The largest deviation for FPR is 0.3 ($(2,1)$ vs. $(1,2)$) which is less than once the standard error and thus clearly not significant. For Sn the largest deviation is 0.5 ($(2,1)$ vs. $(1,2)$, $(1,0)$ vs. $(0,1)$) which again is not significant.

	$d_{\text{bg}} = 0$	$d_{\text{bg}} = 1$	$d_{\text{bg}} = 2$
$d_{\text{fg}} = 0$	0.951 4.8 25.9	0.950 4.7 24.1	0.949 4.4 23.8
$d_{\text{fg}} = 1$	0.950 4.6 23.6	0.948 5.1 22.4	0.947 5.3 23.9
$d_{\text{fg}} = 2$	0.949 4.5 24.1	0.947 5.0 23.4	0.942 6.9 24.8

Table 1: Measures of accuracy for maximum supervised posterior and the standard Gaussian prior. In each cell, the first line shows the AUC, the second line FPR, and the third line Sn. Double lines indicate areas with identical maximum order.

If, on the other hand, we compare the performance for identical maximum orders, the observed differences are significant. For AUC we find the largest difference for $(0,2)$ vs. $(2,2)$, and $(2,0)$ vs. $(2,2)$, respectively, which is 0.007 (10 times standard error). For FPR the largest deviation is 2.4 (6 times standard error) for $(2,0)$ vs. $(2,2)$, and 2.5 for $(0,2)$ vs. $(2,2)$, respectively. For Sn we find differences of 1.2 (3 times standard error) for $(1,0)$ vs. $(1,1)$, and 1.7 for $(0,1)$ vs. $(1,1)$.

Hence, it may be worthwhile to inspect all elements of the upper (or lower) triangular matrix, when searching for the optimal classifier for a given problem. On the other hand – as we already know from the theoretical results – we only need to consider a triangular and not the full matrix.

Next we examine to which extend the behavior changes, if we use differing variances for the parameters of the two models and thus loose the swap-order property. In order to use the Gaussian prior from [Grau *et al.*, 2007], we first must determine reasonable values for κ_{fg}, κ_{bg}, μ_{fg}, and σ^2_{fg}. In [Grau *et al.*, 2007] we choose μ_{fg} and σ^2_{fg} according to an empirical study by [Stepanova *et al.*, 2005], which leads to $\mu_{\text{fg}} = -8.634$ and $\sigma^2_{\text{fg}} = 5.082$, and we determine appropriate values for κ_{fg} and κ_{bg} in a pre-study on a data set for another transcription factor (Sp-1), which are $\kappa_{\text{fg}} = 2$ and $\kappa_{\text{bg}} = 0.005$. The results for the Gaussian prior using these values are presented in table 2.

	$d_{\text{bg}} = 0$	$d_{\text{bg}} = 1$	$d_{\text{bg}} = 2$
$d_{\text{fg}} = 0$	0.951 4.5 25.5	0.952 4.1 26.3	0.953 3.8 26.9
$d_{\text{fg}} = 1$	0.948 5.5 22.3	0.948 5.6 22.0	0.949 5.1 21.5
$d_{\text{fg}} = 2$	0.919 14.8 15.7	0.92 14.4 15.6	0.92 14.4 15.5

Table 2: Measures of accuracy for maximum supervised posterior and the Gaussian prior with differing variances. In each cell, the first line shows the AUC, the second line FPR, and the third line Sn. Double lines indicate areas with identical maximum order.

Considering AUC, we find the largest difference between $(2, 0)$ and $(0, 2)$, which is 0.034 (more than 45 times standard error). Interestingly, if either the max-order or the swap-order property held for this prior, we would expect these very values to be nearly equal . Considering FPR and Sn the pattern is similar: The largest difference for FPR is 11, which is again observed between $(0, 2)$ and $(2, 0)$. This corresponds to more than 25 standard errors. For Sn we find the most substantial deviation between $(2, 2)$ and $(0, 2)$ amounting to 11.4 (more than 25 standard errors). The differences between $(0, 2)$, and $(2, 0)$ and $(2, 1)$ are in the same range (11.3 and 11.2, respectively).

In our study, the differences of accuracy between combinations of Markov models with the same maximum order are highly significant. From these results, we can conclude that in general it is worthwhile to consider all combinations of orders when using maximum supervised posterior in conjunction with a general Gaussian prior.

4 Conclusion

In this paper we use Markov models as the underlying statistical models to construct classifiers. We consider two discriminative approaches, namely maximum conditional likelihood and maximum supervised posterior, to learn the parameters of the Markov models. For these approaches we investigate to what extend the performance of a classifier using the trained models is depending on the choice of orders of the Markov models.

For maximum conditional likelihood we find through theoretical considerations that for two class problems the power of the classifier depends solely on the maximum of the two orders chosen. We call this the max-order property. This is in stark contrast to generative learning, e.g. using maximum likelihood, where in general the combination of model orders influences classification results. This observation can be exploited in computational experiments to reduce computation time: If we consider Markov models up to order $d - 1$, it is not necessary to compute the results for all d^2 combinations of model orders. Rather, it is sufficient to vary the maximum order from 0 to $d - 1$. The order of the second Markov model can be e.g. set to the maximum order, or can be fixed to 0.

For maximum supervised posterior the situation is different and depends on the prior on the parameters employed. For the choice of independent standard Gaussian densities as the prior we show that swapping the orders leaves the estimated posterior class probabilities invariant. However the max-order property does not hold in general. The same is true if we allow the variances of the Gaussians to vary between the different parameters. We empirically study the extent of these influences for the classification of transcription factor binding sites. For Markov models of varying orders we find that the differences between combinations of models having the same maximum order may exceed 40 times the standard error, which is highly significant. Hence we recommend to examine all combinations of orders up to a chosen order d when using maximum supervised posterior.

Acknowledgements

We thank the German Ministry of Education and Research (BMBF Grant No. 0312706A/D) for financial support.

References

[Barash *et al.*, 2003] Y. Barash, G. Elidan, N. Friedman, and T. Kaplan. Modeling dependencies in protein-dna binding sites. In *RECOMB '03: Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 28–37, New York, NY, USA, 2003. ACM Press.

[Ben-Gal *et al.*, 2005] I. Ben-Gal, A. Shani, A. Gohr, J. Grau, S. Arviv, A. Shmilovici, S. Posch, and I. Grosse. Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics*, 21(11):2657–2666, 2005.

[Bishop, 2006] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Cerquides and de Mántaras, 2005] J. Cerquides and R. López de Mántaras. Robust bayesian linear classifier ensembles. In *ECML*, pages 72–83, 2005.

[Feelders and Ivanovs, 2006] A. Feelders and J. Ivanovs. Discriminative scoring of bayesian network classifiers: a comparative study. In *Proceedings of the Third European workshop on probabilistic graphical models*, pages 75–82, 2006.

[Grau *et al.*, 2007] Jan Grau, Jens Keilwagen, Alexander Kel, Ivo Grosse, and Stefan Posch. Supervised posteriors for dna-motif classification. In *Proceedings of the German Conference on Bioinformatics*, 2007. to appear.

[Greiner *et al.*, 2005] R. Greiner, X. Su, B. Shen, and W. Zhou. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning Journal*, 59(3):297–322, 2005.

[Grünwald *et al.*, 2002] P. Grünwald, P. Kontkanen, P. Myllymäki, T. Roos, H. Tirri, and H. Wettig. Supervised posterior distributions. Presented at the Seventh Valencia International Meeting on Bayesian Statistics, 2002.

[Heckerman *et al.*, 1994] David Heckerman, Dan Geiger, and David Maxwell Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. In *KDD Workshop*, pages 85–96. Morgan Kaufmann, 1994.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[Meila-Predoviciu, 1999] M. Meila-Predoviciu. *Learning with Mixtures of Trees*. PhD thesis, Massachusetts Institute of Technology, 1999.

[Roos *et al.*, 2005] T. Roos, H. Wettig, P. Grünwald, P. Myllymäki, and H. Tirri. On discriminative bayesian network classifiers and logistic regression. *Machine Learning*, 59(3):267–296, June 2005.

[Salzberg, 1997] S. L. Salzberg. A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Comput. Appl. Biosci.*, 13(4):365–376, 1997.

[Staden, 1984] R. Staden. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research*, 12:505–519, 1984.

[Stepanova *et al.*, 2005] M. Stepanova, T. Tiazhelova, M. Skoblov, and A. Baranova. A comparative analysis of relative occurrence of transcription factor binding sites in vertebrate genomes and gene promoter areas. *Bioinformatics*, 21(9):1789–1796, 2005.

[Stormo *et al.*, 1982] G. D. Stormo, T. D. Schneider, L. M. Gold, and A. Ehrenfeucht. Use of the 'perceptron' algorithm to distinguish translational initiation sites. *NAR*, 10:2997–3010, 1982.

[Wettig *et al.*, 2002] H. Wettig, P. Grünwald, T. Roos, P. Myllymäki, and H. Tirri. On supervised learning of bayesian network parameters. Technical Report HIIT Technical Report 2002-1, Helsinki Institute for Information Technology HIIT, 2002.

[Yakhnenko *et al.*, 2005] O. Yakhnenko, A. Silvescu, and V. Honavar. Discriminatively trained markov model for sequence classification. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 498–505, Washington, DC, USA, 2005. IEEE Computer Society.

[Zhang and Marr, 1993] M.O. Zhang and T.G. Marr. A weight array method for splicing signal analysis. *Comput. Appl. Biosci.*, 9(5):499–509, 1993.

Recognition of splice sites using maximum conditional likelihood

Jens Keilwagen[1], **Jan Grau**[2], **Stefan Posch**[2], **and Ivo Grosse**[1,2]

[1] Leibniz Institute of Plant Genetics and Crop Plant Research, Gatersleben, Germany
[2] Institute for Computer Science, Martin-Luther-University Halle, Germany

Jens.Keilwagen@ipk-gatersleben.de

Abstract

Many different approaches have been proposed for the problem of splice sites recognition. Employing statistical models there are at least two complementary strategies to improve the performance of prediction, the usage of more complex models and the application of alternative training methods. Here we present a discriminative approach for simple Markov models that outperforms quite complex maximum entropy models.

1 Introduction

The recognition of splice sites is one of the main challenges in eukaryotic gene prediction. Splice sites are at the borders between exonic and intronic DNA constituting eukaryotic genes. Two types of splice sites are distinguished, namely donor (5') and acceptor (3') sites. After the transcription of a gene into pre-mRNA, splicing removes the introns and joins the exons resulting in mature mRNA. This modification has a major impact on the function of the resulting gene product. For this reason the computational recognition of splice sites is an important issue.

Many different approaches have been proposed to predict splice sites, i.e. to distinguish donor or acceptor sites from a class of decoy sequences (e.g. [Stormo *et al.*, 1982; Staden, 1984; Zhang and Marr, 1993; Salzberg, 1997; Burge and Karlin, 1997; Hatzigeorgiou *et al.*, 1996; Brunak *et al.*, 1991; Sonnenburg *et al.*, 2002]). For approaches using statistical models there are at least two complementary strategies to improve the performance of prediction. On the one hand, more complex and appropriate models can be employed, e.g. extending position weight matrix to higher order or using non-sequential dependencies as with Bayesian networks [Cai *et al.*, 2000; Y. Barash and Kaplan, 2003] or permuted Markov models [Ellrott *et al.*, 2002]. On the other hand, different training approaches can be applied. In this paper, we consider the latter direction.

Often *generative* approaches as the maximum likelihood (ML) principle are used for parameter estimation. Generative approaches aim at an accurate estimations of the distribution of nucleotides within the sequences. Optimization is accomplished for each of the classes separately. In contrast, *discriminative* approaches focus on the problem of assigning sequences to a set of known classes and directly estimate the posterior class probabilities. One example of discriminative learning is the maximum conditional likelihood (MCL) approach which may be viewed as the discriminative analogue to ML. Discriminative learning approaches have been successfully applied to protein sequences ([Yakhnenko *et al.*, 2005]) and prediction of transcription factors bindings sites ([Grau *et al.*, 2007b]). Here we investigate if MCL can be used to improve the prediction of splice sites.

This paper is structured as follows: In section 2 we introduce inhomogeneous Markov models, the maximum likelihood and the maximum conditional likelihood principle. In section 3 we present and discuss our result, and end with conclusions in section 4.

2 Methods

2.1 Classification

The Bayes classifier assigns a sequence $\underline{x} = (x_1, x_2, \ldots, x_L)$ of length L to class $c^* \in C$ using

$$c^* = \operatorname*{argmax}_{c \in C} P(c|\underline{x}) \tag{1}$$

$$= \operatorname*{argmax}_{c \in C} P(c, \underline{x}) = \operatorname*{argmax}_{c \in C} P(c) \cdot P(\underline{x}|c), \tag{2}$$

where $P(c|\underline{x})$ denotes the posterior probability of class c given sequence $\underline{x}$, $P(c,\underline{x})$ denotes the joint probability of class c and sequence $\underline{x}$, $P(c)$ denotes the prior probability of class c, and $P(\underline{x}|c)$ denotes the likelihood of sequence $\underline{x}$ given class c.

Typically, a parametric family of probability distributions is chosen, and the parameters are estimated from a set of training data. We assume a set of training data of N independent and identically distributed (i.i.d.) data points $(\underline{x}_n, c_n)$, and we denote $\underline{D} = (\underline{x}_1, \ldots, \underline{x}_N)$ and $\underline{c} = (c_1, \ldots, c_N)$. The family of probability distributions we consider in the following are inhomogeneous Markov models (iMM). We discuss two principles for estimating their parameters, namely the maximum likelihood (ML) principle and the maximum conditional likelihood (MCL) principle.

In this paper we consider only two-class problems. In this context we denote the first class as *real* and the second as *decoy*. Furthermore we use the *alphabet* $\Sigma = \{A, C, G, T\}$ for all positions of the sequences.

2.2 Markov models

IMMs are widely used for modeling splice sites [Zhang and Marr, 1993; Salzberg, 1997]. For an iMM of order d_c (iMM(d_c)) each observation at position $\ell \in [1, L]$ may depend only on its $d_{c,\ell} = \min\{d_c, \ell - 1\}$ predecessors, resulting in

$$P(\underline{x}|c) = \prod_{\ell=1}^{L} P_\ell(x_\ell | x_{\ell - d_{c,\ell}}, \ldots, x_{\ell-1}, c). \tag{3}$$

The observations $x_{\ell-d_{c,\ell}},\dots,x_{\ell-1}$ are called the context of x_ℓ, which is empty for $\ell = 1$. An $\mathrm{iMM}(0)$, which assumes all L positions conditionally independent given the class, is often called a position weight matrix (PWM) model [Stormo *et al.*, 1982; Staden, 1984],

There are many different parameterizations for iMMs. One popular parameterization of an $\mathrm{iMM}(d_c)$ is given by

$$\theta_c := P(c) \tag{4}$$

$$\theta_{\ell,a|\underline{b},c} := P_\ell(a|\underline{b},c). \tag{5}$$

where θ_c and $\theta_{\ell,a|\underline{b},c}$ are non-negative for all $c \in C$, $\ell \in [1,L]$, $a \in \Sigma$ and $\underline{b} \in \Sigma^{d_{c,\ell}}$ and fulfill the constraints $\sum_{c\in C}\theta_c = 1$ and $\sum_{a\in\Sigma}\theta_{\ell,a|\underline{b},c} = 1$ for all $c \in C$, $\ell \in [1,L]$ and $\underline{b} \in \Sigma^{d_{c,\ell}}$. Another popular parameterization derived from Markov random fields (MRFs) and conditional random fields (CRFs) [Lafferty *et al.*, 2001; Wettig *et al.*, 2002] is given by

$$P(c,\underline{x}|\underline{\lambda}) \propto \exp\left(\lambda_c + \sum_{\ell=1}^{L}\lambda_{\ell,x_\ell|x_{\ell-d_{c,\ell}},\dots,x_{\ell-1},c}\right),$$

where for all $c \in C$, $\ell \in [1,L]$, $a \in \Sigma$ and $\underline{b} \in \Sigma^{d_{c,\ell}}$: $\lambda_c, \lambda_{\ell,a|\underline{b},c} \in \mathbb{R}$.

2.3 Maximum likelihood

The ML principle suggests to choose those parameters $\underline{\theta}$ that maximize the likelihood $P(\underline{D},\underline{c}|\underline{\theta})$ of the complete data set $(\underline{D},\underline{c})$,

$$\begin{aligned}\hat{\underline{\theta}}^{\mathrm{ML}} &= \operatorname*{argmax}_{\underline{\theta}} P(\underline{D},\underline{c}|\underline{\theta}) = \operatorname*{argmax}_{\underline{\theta}} \prod_{n=1}^{N} P(\underline{x}_n, c_n|\underline{\theta})\\ &= \operatorname*{argmax}_{\underline{\theta}} \prod_{c\in C} P(c|\underline{\theta}) \prod_{n,\text{ where } c_n=c} P(\underline{x}_n|c,\underline{\theta})\end{aligned}$$

If the parameters of $P(\underline{x}|c,\underline{\theta})$ are pairwise independent between the classes and also from the prior class probabilities, the ML estimates can be obtained for each class separately. ML is called generative because it aims at an accurate estimation of the distribution $P(\underline{x},c|\underline{\theta})$.

For an $\mathrm{iMM}(d_c)$, we obtain the familiar ML estimates

$$\hat{\theta}_c^{\mathrm{ML}} = \frac{N_c}{N} \tag{6}$$

$$\hat{\theta}_{\ell,a|\underline{b},c}^{\mathrm{ML}} = \frac{N_{\ell,a|\underline{b},c}}{\sum_{\tilde{a}\in\Sigma} N_{l,\tilde{a}|\underline{b},c}}, \tag{7}$$

for $c \in C$, $\ell \in [1,L]$, $a \in \Sigma$, and $\underline{b} \in \Sigma^{d_{c,\ell}}$. Here, N_c denotes the number of sequences of class c in the data set and $N_{\ell,a|\underline{b},c}$ denotes the number of occurrences of symbol a at position ℓ given context $\underline{b}$ in all sequences of class c.

2.4 Maximum conditional likelihood

The MCL principle suggests to choose those parameters $\underline{\theta}$ that maximize the conditional likelihood,

$$\hat{\underline{\theta}}^{\mathrm{MCL}} = \operatorname*{argmax}_{\underline{\theta}} P(\underline{c}|\underline{D},\underline{\theta}) = \operatorname*{argmax}_{\underline{\theta}} \prod_{n=1}^{N} P(c_n|\underline{x}_n,\underline{\theta}).$$

The MCL principle is more directly linked to the classification rule (1) than the ML principle, because it focuses on the posterior class probabilities, and it has been successfully applied to classifiers based on Markov models [Yakhnenko *et al.*, 2005] and Bayesian networks [Wettig *et al.*, 2002; Greiner *et al.*, 2005; Grossman and Domingos, 2004].

In contrast to ML estimators, MCL estimators cannot be obtained analytically for several popular models including iMMs, Bayesian networks, and MRFs. Hence, numerical optimization techniques, such as gradient ascent, must be recruited [Wallach, 2002]. Unfortunately, neither the conditional likelihood nor the log conditional likelihood are guaranteed to be concave functions of the parameters $\underline{\theta}$ [Wettig *et al.*, 2002]. Hence, local maxima or saddle points could exist, and numerical optimization techniques are not guaranteed to reach the global maximum. However, the log conditional likelihood of Markov models

$$\begin{aligned}&\log P(\underline{c}|\underline{D},\underline{\lambda})\\ &= \sum_{c\in C} N_c\lambda_c + \sum_{\ell=1}^{L}\sum_{\underline{b}\in\Sigma^{d_{c,\ell}}}\sum_{a\in\Sigma} N_{\ell,a|\underline{b},c}\lambda_{\ell,a|\underline{b},c}\\ &\quad - \sum_{\underline{x}\in D}\log\left(\sum_{c\in C}\exp\left(\lambda_c + \sum_{\ell=1}^{L}\lambda_{\ell,x_\ell|x_{\ell-d_{c,\ell}},\dots,x_{\ell-1},c}\right)\right)\end{aligned}$$

can be shown to be a concave function of $\underline{\lambda}$ [Lafferty *et al.*, 2001], resulting in a guaranteed convergence of any local optimization algorithm to the single (despite in general degenerate) global maximum.

2.5 Performance Measures

Following [Yeo and Burge, 2004], we use the the maximum correlation coefficient ($\mathrm{CC}_{\max}$) and the area under the ROC curve (AUC) as performance measures. $\mathrm{CC}_{\max}$ varies between -1 and 1, yielding a value of 0 for a classifier just randomly guessing and a value of 1 for a perfect classifier. The AUC, which indicates the overall performance of a classifier, varies between 0.0 and 1, yielding a value of 0.5 for random guessing and a value of 1 for a perfect classifier.

2.6 Data

We use the data sets of donor and acceptor splice sites from [Yeo and Burge, 2004]. In these data sets, donor sites match the pattern $\mathrm{N_3GTN_4}$, and acceptor sites match the pattern $\mathrm{N_{18}AGN_3}$. Following [Yeo and Burge, 2004], we eliminate the consensus dinucleotide GT from all donor sites and the consensus dinucleotide AG from all acceptor sites, yielding donor sites of length 7 bp and acceptor sites of length 21 bp. This is sensible, since real and decoy sequences in train and test data set contain these dinucleotids. Table 1 lists the number of sequences of in each data set.

		donor sites	acceptor sites
train	real	8,415	8,465
	decoy	179,438	180,957
test	real	4,208	4,233
	decoy	89,717	90,494

Table 1: Number of sequences of the data sets of [Yeo and Burge, 2004].

3 Results and Discussion

We train classifiers using iMMs of order $d \in [0,3]$ as foreground and background models on the training data sets, and we evaluate their classification performance on the test set using the performance measures $\mathrm{CC}_{\max}$ and AUC. Specifically, we evaluate all 16 model combinations in the generative approach, and 4 different model combinations (using the same order d for the foreground and the background model) in the discriminative approach. This restriction is justified because in [Feelders and Ivanovs, 2006] it was shown that for MCL the parameters of one of the classes are redundant. Using this idea one can easily prove that for two-class problems the power of a classifier using Markov models is determined by the maximum of the orders of the two employed models [Grau *et al.*, 2007a].

For comparison of both estimation methods for increasing order d of the models, we proceed as follows. For a given order d we choose as the ML classifier the best combination of model orders where the foreground or background order is d and the other order is limited to at most d. Its performance is compared to the MCL classifier where both foreground and background order are set to d. In addition we relate these results with data from [Yeo and Burge, 2004], where a wealth of different models for the recognition of splice sites are studied using a generative approach. The values given here deviate slightly from the figures given in [Yeo and Burge, 2004] due to the different implementations of numerical optimization employed.

As an alternative generative approach the maximum a posteriori principle is often used employing the idea of Bayesian inference. For iMM usually a product of Dirichlet densities is chosen as the prior for the parameters. Adopting this approach for parameter estimation with an equivalent sample size of 1 and 4 the performance is essentially identical to ML estimation as reported in this section (data not shown). This may be due to the size of the data sets. In the following we focus on the differences between the estimation methods without any prior, since to our knowledge no sensible prior is defined that can be used for both estimation methods.

3.1 Donor sites

Figure 1 shows AUC and $\mathrm{CC}_{\max}$ for increasing model order d and estimation techniques. For the ML approach the maximum correlation coefficient increases from order 0 to order 2 and then decreases. For order 0 we obtain $\mathrm{CC}_{\max}$ of 0.5934, and for order 1 an increase of 0.0549 to $\mathrm{CC}_{\max}$ of 0.6483 indicating that statistical dependencies among nearest neighbors are present in donor splice sites. Neglecting these, as the PWM does, results in poor performance. Increasing the order from 1 to 2, we observe a slight improvement to $\mathrm{CC}_{\max}$ = 0.6491, indicating that there are also statistical dependencies among second-nearest neighbors. For model order 3, the $\mathrm{CC}_{\max}$ decreases, indicating that an $\mathrm{iMM}(3)$ is already over-fitted. Here, we find that the optimal order is 2, which is consistent with the observations of [Cai *et al.*, 2000].

[Yeo and Burge, 2004] find as the best model the maximum entropy model me2x5 which incorporates all dependencies between two positions. This model is quite complex and can not be trained analytically. The latter is also true for the MCL models studied in this paper. The me2x5 improves the value of the $\mathrm{CC}_{\max}$ of the best classifier by 0.0076.

For the MCL approach we find a qualitatively similar behavior as for the ML approach with regard to varying model order. Performance increases from order 0 to 2, and it decreases for order 3 which again can be explained by over-fitting. Comparing the performance we find that the discriminatively trained classifier outperforms its generative counterpart for all model orders. For model orders 1 and 2 it also outperforms the model me2x5. The best classifier is a discriminatively trained classfier based on $\mathrm{iMM}(2)$, which gives a $\mathrm{CC}_{\max}$ of 0.6610.

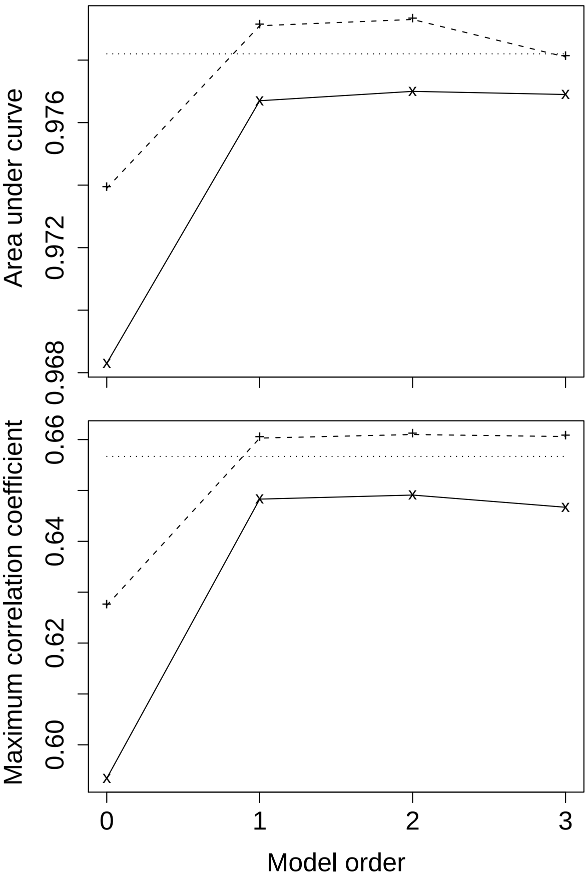


Figure 1: AUC (top) and $\mathrm{CC}_{\max}$ (bottom) for the recognition of donor splice sites. The x-axis gives the model order. The solid line shows ML for Markov models, the dashed line MCL for Markov models, and the dotted line shows the result of the me2x5 model.

Considering AUC for the ML approach the performance increases from 0.9683 for order 0 to 0.9770 for order 2. For order 3 we observe a slight decrease of performance. The best ML-classifier yields an AUC of 0.9770. This value can be improved about 0.0012 by the classifier based on me2x5.

For the MCL approach the values of the AUC are increasing for model order 0 to 2, while it decreases for model order 3. Comparing the different approaches we observe that for all model orders MCL is superior to ML. For model order 1 and to 2 the MCL approach outperforms the classifier based on the maximum entropy model. The best classifier is a discriminatively trained classifier based on $\mathrm{iMM}(2)$, which gives an AUC of 0.9793.

Summarizing, the best discriminatively trained classifier improves the values obtained by the me2x5 model by 0.0043 for $\mathrm{CC}_{\max}$ and 0.0011 for the AUC. This gain is comparable to the improvement of the me2x5 model the iMM with ML estimation.

Figure 2 shows the part of the ROC curves around 95% true positive rate for the best classifiers based on generatively trained iMMs, me2x5 and discriminatively trained iMMs. In this interval the classifier based on me2x5 out-

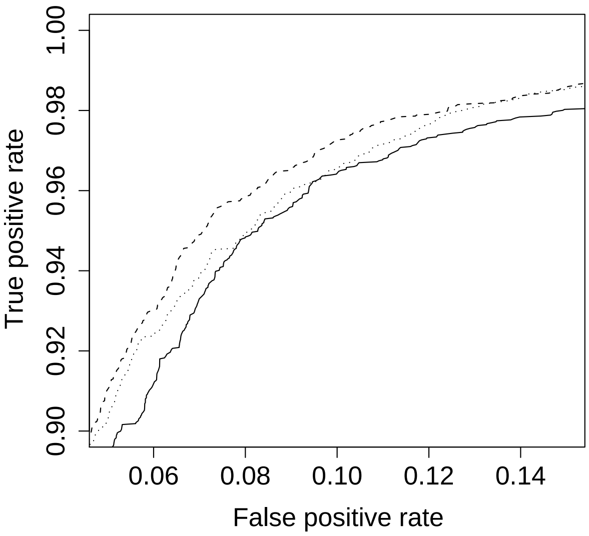


Figure 2: A part of the ROC curves for the recognition of donor splice sites for the optimal generatively trained classifier based on iMMs (solid), the optimal generatively trained classifier based on me2x5 (dotted), and the optimal discriminatively trained classifier based on iMMs (dashed). We observe that discriminatively trained classifier is consistently superior to both generatively trained classifiers.

performs the generatively trained classifier based on iMMs and the discriminatively trained classifier outperforms this one.

3.2 Acceptor sites

For acceptor sites we obtain similar results as for donor sites. We present these results in figure 3. The performance is increasing for order 0 to 2 and decreasing for order 3 for both training methods. For the ML approach $\mathrm{CC}_{\max}$ could be improved from 0.5595 for model order 0 to 0.6312 for model order 2, indicating that both nearest neighbor and second nearest neighbor dependencies are worth to be modeled. The model proposed by [Yeo and Burge, 2004] is an approximation of the maximum entropy model me2x2. The model me2x2 incorporates all dependencies between two positions that have a distance of at most 2 positions. Since maximum entropy models can in general only be learned numerically and the runtime is mainly determined by a sum over all possible outcomes of $\underline{x}$ the runtime grows exponentially with the length of the sequences. This is the reason why for length 21 no maximum entropy model could be learned generatively in acceptable time. In [Yeo and Burge, 2004] this is circumvented by an approximation of the maximum entropy model using small overlapping fragments. For these fragments it is possible to train the corresponding maximum entropy models. Considering $\mathrm{CC}_{\max}$ the corresponding classifier performs insignificantly worse than the generatively trained classifier of two inhomogeneous Markov models of order 2 with an $\mathrm{CC}_{\max}$ of 0.6293. The discriminatively trained classifiers clearly outperform their generative counterparts. The best discriminatively trained classifier gives a $\mathrm{CC}_{\max}$ of 0.6441.

For the AUC and the ML approach we obtain a similar behavior. From model order 0 with 0.9624 to model order 2 with 0.9747 the performance is increasing, while it slightly decreases for order 3. The model from [Yeo and Burge, 2004] outperforms these classifiers and we obtain an improvement of 0.0007 to an AUC of 0.9754. If we consider the discriminatively trained classifier we observe a similar behavior as for $\mathrm{CC}_{\max}$. For order 0 to 2 the performance is increasing, while it decreases for order 3. Comparing the different approaches we observe that for order 0 to 2 MCL is superior to ML, while for order 3 ML is superior to MCL. For model order 1 and 2 the MCL approach outperforms the approximation of the me2x2 yielding an AUC of 0.9770 for order 2, which is an improvement of 0.0016. This is more than twice the improvement that was made by [Yeo and Burge, 2004].

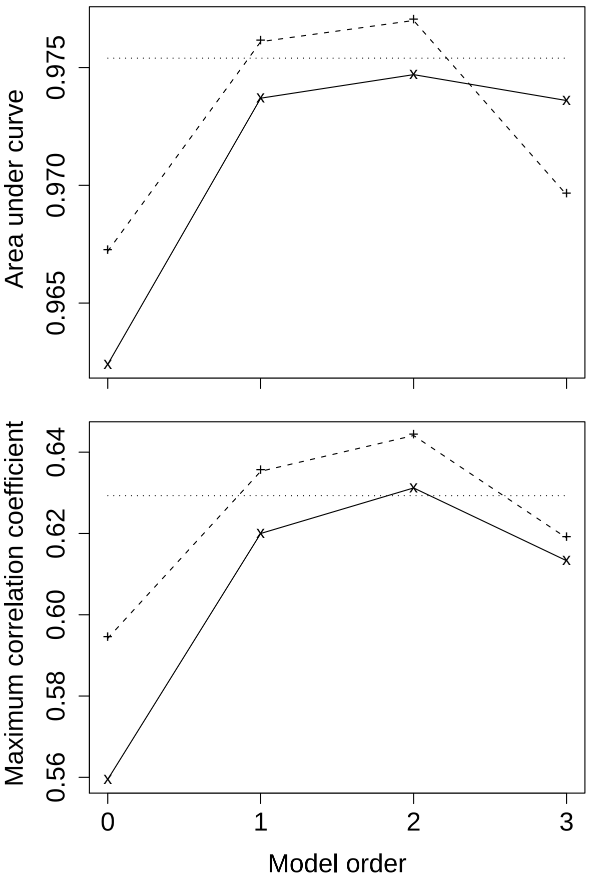


Figure 3: AUC (top) and $\mathrm{CC}_{\max}$ (bottom) for the recognition of acceptor splice sites. The x-axis gives the model order. The solid line shows ML for Markov models, the dashed line MCL for Markov models, and the dotted line the result of the me2x2 model.

Figure 4 shows the ROC curves for best classifiers based on generatively trained iMMs, me2x2 and discriminatively trained iMMs. We observe that the discriminatively trained classier is consistently superior to both generatively trained classifiers.

3.3 Discussion

We observe that for donor sites discriminatively trained classifiers outperform their generative counterparts for all model orders and for all performance measures. For acceptor sites the behaviour is the same with the only exception for model order 3. Here the ML approach performs better than the discriminative, however both models are already

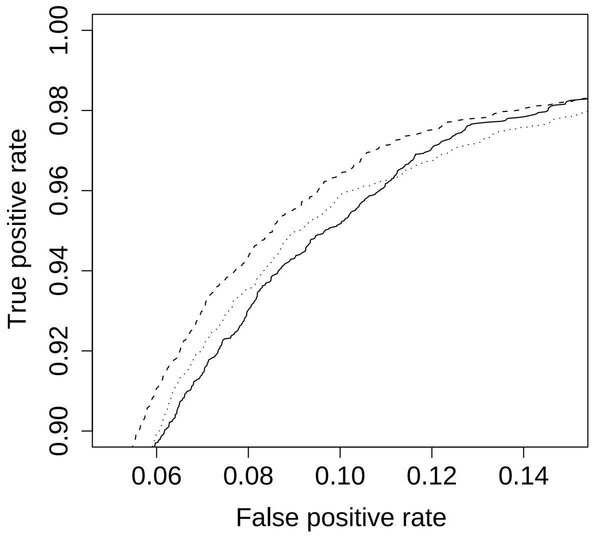


Figure 4: A part of the ROC curves for the recognition of acceptor splice sites for the optimal generatively trained classifier based on iMMs (solid), the optimal generatively trained classifier based on me2x2 (dotted), and the optimal discriminatively trained classifier based on iMMs (dashed).

over-fitted. Thus overall the MCL approach clearly outperforms the generative ML approach. Furthermore the discriminatively trained classifier considerably outperforms the maximum entropy models of [Yeo and Burge, 2004] as well.

We observe for both types of splice sites that dependencies between nearest and second nearest neighbor exist. These dependencies are modeled by an $\mathrm{iMM}(2)$. We observe that order 2 gives the best results in all cases, which coincides with results from other researchers [Cai *et al.*, 2000]. However the ML approach only focuses on the dependencies within one class without taking the other class into account, since it tries to model the distribution of one class perfectly. If those dependencies are partly overlapping it is not worth to model them for classification, since this will not improve the classification accuracy. This might be a reason why the MCL approach taking the dependencies of both class jointly into account is superior to the ML approach.

4 Conclusion

In this paper we apply the discriminative MCL approach using iMMs of varying order to the prediction of splice sites and compare the results to generative ML approach with iMMs of the same maximum order and the best models proposed by [Yeo and Burge, 2004]. Considering the results for the optimal model orders, for donor splice sites MCL improves the $\mathrm{CC}_{\max}$ of 0.6491 for ML and 0.6567 for me2x5 to 0.6610. For the AUC we observe a similar increase from 0.9770 (ML) and 0.9782 (me2x5) to 0.9793. For acceptor sites the behaviour is similar. Again, MCL yielding $\mathrm{CC}_{\max}$ of 0.6441 is superior to ML with 0.6312. In this case the classifier based on me2x5 gives worse performance than both the iMM based classifiers. For the AUC we observed a dramatic improvement from 0.9747 (ML) over 0.9754 (me2x5) to 0.9770 for the discriminative approach.

For almost all orders we found the discriminatively trained classifier to outperform its generative counterpart. Furthermore the discriminatively learned simple Markov model outperforms the complex maximum entropy model.

These findings suggest that it could be worthwile to apply discriminative approaches for parameter estimation, such as MCL, to other pattern recognition problems in bioinformatics, e.g. the recognition of exonic or intronic splicing enhancers or silencers, transcription factor binding sites, nucleosome binding sites, or ribosome binding sites. Furthermore the discriminative approach can be used for a wide range of statistical models, that have already proven to perform well in generative approaches. Especially for variable order approach [Rissanen, 1983; Ron *et al.*, 1996; Buehlmann, 1997; Zhao *et al.*, 2005; Ben-Gal *et al.*, 2005; Castelo and Kocka, 2003].

Acknowledgement

We thank the anonymous reviewers for providing helpfull hints, and BMBF (grant no. 0312706A) for financial support.

References

[Ben-Gal *et al.*, 2005] I. Ben-Gal, A. Shani, A. Gohr, J. Grau, S. Arviv, A. Shmilovici, S. Posch, and I. Grosse. Identification of transcription factor binding sites with variable-order bayesian networks. *Bioinformatics*, 21:2657–2666, 2005.

[Brunak *et al.*, 1991] Søren Brunak, Jacob Engelbrecht, and Steen Knudsen. Prediction of human mRNA donor and acceptor sites from the DNA sequence. *Journal Molecular Biology*, 220:49–65, 1991.

[Buehlmann, 1997] P. Buehlmann. Model selection for variable length markov chains and tuning the context algorithm. Technical Report 82, Statistics, ETH Zentrum, CH-8092 Zuerich, Switzerland, September 1997.

[Burge and Karlin, 1997] C. Burge and S. Karlin. *Prediction of complete gene structures in human genomic DNA.*, volume 268. J. Mol. Biol., 1997.

[Cai *et al.*, 2000] Deyou Cai, Arthur L. Delcher, Ben Kao, and Simon Kasif. Modeling splice sites with bayes networks. *Bioinformatics*, 16(2):152–158, 2000.

[Castelo and Kocka, 2003] Robert Castelo and Tomás Kocka. On inclusion-driven learning of bayesian networks. *Journal of Machine Learning Research*, 4:527–574, 2003.

[Ellrott *et al.*, 2002] K. Ellrott, C. Yang, F. M. Sladek, and T. Jiang. Identifying transcription factor binding sites through markov chain optimization. *In Proceedings of the European Conference on Computational Biology (ECCB 2002)*, pages 100–109, 2002.

[Feelders and Ivanovs, 2006] A. Feelders and J. Ivanovs. Discriminative scoring of bayesian network classifiers: a comparative study. In *Proceedings of the third European workshop on probabilistic graphical models*, pages 75–82, 2006.

[Grau *et al.*, 2007a] Jan Grau, Jens Keilwagen, Ivo Grosse, and Stefan Posch. Discriminative learning of markov models with varying order. In *Proceedings of LWA*, 2007. submitted.

[Grau *et al.*, 2007b] Jan Grau, Jens Keilwagen, Alexander Kel, Ivo Grosse, and Stefan Posch. Supervised posteriors for dna-motif classification. In *Proceedings of the German Conference on Bioinformatics*, 2007. to appear.

[Greiner *et al.*, 2005] R. Greiner, X. Su, B. Shen, and W. Zhou. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning Journal*, 59(3):297–322, 2005.

[Grossman and Domingos, 2004] D. Grossman and P. Domingos. Learning bayesian network classifiers by maximizing conditional likelihood. In *ICML*, pages 361–368. ACM Press, 2004.

[Hatzigeorgiou *et al.*, 1996] A. Hatzigeorgiou, N. Mache, and M. Reczko. Functional site prediction on the dna sequence by artificial neural networks. In *IJSIS '96: Proceedings of the 1996 IEEE International Joint Symposia on Intelligence and Systems*, page 12, Washington, DC, USA, 1996. IEEE Computer Society.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[Rissanen, 1983] J. Rissanen. A universal data compression system. *IEEE Trans. Inform. Theory*, 29(5):656–664, 1983.

[Ron *et al.*, 1996] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25:117–149, 1996.

[Salzberg, 1997] S. L. Salzberg. A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Comput. Appl. Biosci.*, 13(4):365–376, 1997.

[Sonnenburg *et al.*, 2002] Sören Sonnenburg, Gunnar Rätsch, Arun K. Jagota, and Klaus-Robert Müller. New methods for splice site recognition. In *ICANN '02: Proceedings of the International Conference on Artificial Neural Networks*, pages 329–336, London, UK, 2002. Springer-Verlag.

[Staden, 1984] R. Staden. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research*, 12:505–519, 1984.

[Stormo *et al.*, 1982] G. D. Stormo, T. D. Schneider, L. M. Gold, and A. Ehrenfeucht. Use of the 'perceptron' algorithm to distinguish translational initiation sites. *NAR*, 10:2997–3010, 1982.

[Wallach, 2002] Hanna Wallach. Efficient training of conditional random fields. Master's thesis, University of Edinburgh, 2002.

[Wettig *et al.*, 2002] H. Wettig, P. Grünwald, T. Roos, P. Myllymäki, and H. Tirri. On supervised learning of bayesian network parameters. Technical Report HIIT Technical Report 2002-1, Helsinki Institute for Information Technology HIIT, 2002.

[Y. Barash and Kaplan, 2003] N. Friedman Y. Barash, G. Elidan and T. Kaplan. Modeling Dependencies in Protein-DNA Binding Sites. *In proceedings of Seventh Annual International Conference on Computational Molecular Biology*, pages 28–37, 2003.

[Yakhnenko *et al.*, 2005] O. Yakhnenko, A. Silvescu, and V. Honavar. Discriminatively trained markov model for sequence classification. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 498–505, Washington, DC, USA, 2005. IEEE Computer Society.

[Yeo and Burge, 2004] Gene Yeo and Christopher B. Burge. Maximum entropy modeling of short sequence motifs with applications to rna splicing signals. *Journal of Computational Biology*, 11(2/3):377–394, 2004.

[Zhang and Marr, 1993] M.O. Zhang and T.G. Marr. A weight array method for splicing signal analysis. *Comput. Appl. Biosci.*, 9(5):499–509, 1993.

[Zhao *et al.*, 2005] Xiaoyue Zhao, Haiyan Huang, and Terence P. Speed. Finding short dna motifs using permuted markov models. *Journal of Computational Biology*, 12(6):894–906, 2005.

Analysing latent topics in large-scale proteomics databases

S. Klie, A. Hinneburg
Institute of Computer Science
Martin-Luther-University Halle-Wittenberg, Germany
{klie,hinneburg}@informatik.uni-halle.de

L. Martens, J. A. Vizcaino, R. Cote, P. Jones, R. Apweiler, H. Hermjakob
EMBL Outstation, European Bioinformatics Institute,
Wellcome Trust Genome Campus,
Hinxton, Cambridge, UK
{lennart.martens,juan,rcote,pjones,apweiler,hhe}@ebi.ac.uk

Abstract

Since the advent of public data repositories for proteomics data, readily accessible results from high-throughput experiments have been accumulating steadily. Several large-scale projects in particular have contributed substantially to the amount of identifications available to the community. Despite the considerable body of information amassed, very few successful analysis have been performed and published on this data, levelling off the ultimate value of these projects far below their potential. In order to illustrate that these repositories should be considered sources of detailed knowledge instead of data graveyards, we here present a novel way of analyzing the information contained in proteomics experiments with a 'latent semantic analysis'. We apply this information retrieval approach to the peptide identification data contributed by the Plasma Proteome Project. Interestingly, this analysis is able to overcome the fundamental difficulties of analyzing such divergent and heterogeneous data emerging from large scale proteomics studies employing a vast spectrum of different sample treatment and mass-spectrometry technologies. Moreover, it yields several concrete recommendations for optimizing proteomics project planning as well as the choice of technologies used in the experiments. It is clear from these results that the analysis of large bodies of publicly available proteomics data holds great promise and is currently underexploited.

1 Introduction

The field of proteomics has undergone several dramatic changes over the past few years. Advances in instrumentation and separation technologies [1, 6] have enabled the advent of high-throughput analysis methods that generate large amounts of proteomics identifications per experiment. Many of these datasets were initially only published as supplementary information in PDF format and, while available, were not readily accessible to the community. Obviously, this situation led to large-scale data loss and was perceived as a major problem in the field [10, 22].

Several public proteomics data repositories, including the Global Proteome Machine (GPM) [2], the Proteomics Identifications Database (PRIDE) [13, 18] and PeptideAtlas [5] were constructed to turn the available data into accessible data, thereby reversing the trend of increasing data loss.

As a case in point, several large-scale proteomics projects that have recently been undertaken by the Human Proteome Organization (HUPO), including the Plasma Proteome Project (PPP) [21] and the Brain Proteome Project (BPP) [9], have published all of their assembled data in one or more of these repositories. As a result, their findings are readily accessible to interested researchers. It is therefore remarkable to see that very little additional information has so far been extracted from the available data. One of the rare examples where the analysis of large proteomics datasets resulted in a practical application is the recent effort by Mallick and co-workers in which several properties of a large amount of identified peptides were used to fine-tune an algorithm that can predict proteotypic peptides from sequence databases [17].

We here present a novel way to reveal the information that lies hidden in large bodies of proteomics data, by analyzing them for latent semantic patterns. The analysis performed here focused on the HUPO PPP data as available in the PRIDE database. Briefly, the HUPO PPP sent out a variety of plasma and serum samples, collected from different ethnic groups and at different locales worldwide. All of the five resulting plasma samples were additionally treated with one of three distinct methods of anticoagulation: EDTA, citrate or heparin. The total amount of distinct samples thus amounts to twenty: five serum samples, and three times five plasma samples [21].

We used the original peptide sequences to evaluate experiment similarity by performing a latent semantic analysis, a technique often employed in natural language processing. Our results suggest that LSA as well as it probabilistic variant PLSA can be considered useful analysis tools of such data yielding results which cannot easily be obtained by conventional means.

2 Latent semantic analysis

In order to assess inter-experiment similarity in an all-against-all comparison, an information retrieval (IR) method called latent semantic analysis (LSA; also referred to as latent semantic indexing, LSI) is employed. The fundamentals of LSA are well understood and it has been widely used for various information retrieval tasks. The general idea of LSA is to map document into some latent semantic space, in which the dimensions consists of latent topics. The main task is, to reduce the documents from a word-based representation to a topic-based representation, which reduces the influence of noise (random words) during the similarity computation between pairs of documents. Applied to the context of proteomics, experiments take the role of documents while peptides identified in one experiment (more precisely their amino acid sequence) act as terms. The algorithm reports a similarity score for each pair of experiments, based on the latent topics in peptide representation of the experiments.

2.1 Vector Space representation of proteomics data

LSA computes latent topics from a vector representation of the documents (vector space model, VSM). A term-document matrix $W \in \mathbb{R}^{n,m}$ represents a document collection of m documents over a vocabulary of n terms. A value $w_{i,j}$ is the number of occurrences of a particular term i (rows of W) within the jth document (columns of W). In case of proteomics experiments the words are peptides detected by mass spectrometry. As no quantitative information about the peptides is available but just the information about the occurrence, the document term matrix is a bit-matrix in this case.

Each column-vector $\vec{w}_{\cdot,j}$ in W can be interpreted as a document-vector $\vec{d_j}$ which characterizes a document (proteomics experiment) by its terms (peptides) [24]. The similarity between two documents $\vec{a}$ and $\vec{b}$ represented by their documents vectors is determined by cosine similarity, which gives the cosine of the angle between $\vec{a}$ and $\vec{b}$:

$$\begin{aligned} sim(\vec{a},\vec{b}) = \cos\alpha(\vec{a},\vec{b}) &= \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\cdot\|\vec{b}\|} \\ &= \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \end{aligned} \tag{1}$$

The similarity is between zero and one, due to the normalization. To counteract poorly differentiating, often-occurring terms each row of W is weighted by the inverse document frequency (IDF). IDF of a term t is defined as

$$idf_t = \log\frac{N_D}{n_t} \tag{2}$$

with N_D is the overall number of documents and n_t is the number of documents which contain the term t. The effect of IDF-weighting is that poorly differentiating, often-occurring terms will have a much lower weight than highly specific, rare terms.

The vector space model has several drawbacks. First, it relies on exact term matching (because of the scalar product in (1)), thus making it impossible for the model to detect synonyms. A proteomics example of a synonym is the substitution of the isobaric amino acids isoleucine and leucine within a peptide sequence. A second problem is the sparseness of non-zero values in the obtained document/term matrix [14]. The matrix is thus largely composed of zeroes, highlighting the fact that many peptides failed to be identified in more than one experiment, which is a common situation in shotgun proteomics experiments.

2.2 Singular Value Decomposition

In order to reduce the sparseness of the document/term matrix and to detect hidden (latent) term relations, LSA projects the original documents vectors into a lower dimensional semantic space where documents which contain repeatedly co-occurring terms will have a similar vector representation. This effectively overcomes the fundamental deficiencies of the exact term-matching employed in a VSM [15]. As such, LSA might predict that a given term should be associated with a document, even though no such association was observed in the original matrix [4]. The core principle for achieving this is the application of singular value decomposition (SVD), a type of factor analysis which can be applied to any rectangular matrix in the form of:

$$W = U\Sigma V^T \tag{3}$$

where $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{m\times m}$ are orthogonal matrices (i.e. $UU^T = I$ and $VV^T = I$) and $\Sigma = \mathrm{diag}(\sigma_1,\ldots,\sigma_r)$ is a diagonal matrix with $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq 0$ and $r = \min(m,n)$. LSA makes use of the matrix approximation theorem, which states that

$$\underset{W_k \text{ has rank } k}{\mathrm{argmin}} \|W - W_k\|_2 = U_k\Sigma_k V_k^T \tag{4}$$

with U_k and V_k consist of the first k columns of U and V respectively and $\Sigma_k = \mathrm{diag}(\sigma_1,\ldots,\sigma_k)$. So, for $k \leq r$ W_k is the best rank k approximation of W in the least square sense. The approximation error is bounded with respect to the Frobenius norm by $\|W - W_k\|_F \leq \sigma_{k+1}$. The column vectors of $(\Sigma_k V_k^T)$ are the new document vectors in the latent space. The mapping of some original document vector $\vec{d}$ into the latent space is described by $U^t\vec{d}$.

The SVD alters the original values in the matrix W by new estimates, based on the observed co-occurrences of terms and their 'true semantic meaning' within the whole corpus of documents [8]. The latter is achieved because terms with a common meaning are roughly mapped to the same direction in the latent space. By leaving out the smallest singular values, 'weak patterns' or noise are filtered out. The choice of k determines the degree of reduction, and it is therefore important to note that a high k value (corresponding to a weak reduction) might not be able to filter out noise or unimportant fluctuations in the source data, while a very small k value (strong reduction) will retain too little information from the original data structure [7].

2.3 Probabilistic Latent Semantic Analysis

Probabilistic latent semantic indexing (PLSI) introduced in [11, 12] extends the vector space model and Latent Semantic Analysis by SVD. It learns topics hidden in the data. Formally, we denote the set of documents by $D = \{d_1, ..., d_m\}$ and the vocabulary by $T = \{t_1, ..., t_n\}$. The term frequency of a word $t_i \in T$ in a document $d_j \in D$ is denoted as $w_{i,j}$ and the reweighed quantity is $\hat{w}_{i,j} = w_{i,j} \cdot idf(t_i)$. The training data consists of the set of document-word pairs (d_j, t_i) which correspond to non-zero entries $w_{i,j}$ in the term-document matrix W. The joint probability of such a pair is modeled according to the employed aspect model as $P(d_j,t_i) = \sum_{z\in Z} P(z)\cdot P(t_i|z)\cdot P(d_j|z)$. The z are hidden variables, which can take K different discrete values $z \in Z = \{z_1, ..., z_K\}$. In the context of text retrieval z is interpreted as an indicator for a topic. Two assumptions are made by the aspect model. First, it assumes pairs (d_j, t_i) to be statistically independent. Second, conditional independence between t_i and d_j is assumed for a given value for z.

The probabilities necessary for the joint probability $P(d_j, t_i)$, namely $P(z)$, $P(t_i|z)$ and $P(d_j|z)$, are derived by an expectation maximization (EM) learning procedure. The idea is to find values for unknown probabilities, which maximize the complete data likelihood

$$\begin{aligned} P(W, z) &= \prod_{w_{i,j} \neq 0} [P(z) \cdot P(t_i|z) \cdot P(d_j|z)] \quad (5) \\ &= \prod_{d_j \in D} \prod_{t_i \in V} [P(z) \cdot P(t_i|z) \cdot P(d_j|z)]^{w_{i,j}} \quad (6) \\ & \quad (7) \end{aligned}$$

In the E-step the posteriors for z are computed.

$$P(z|d_j, t_i) = \frac{P(z) \cdot P(t_i|z) \cdot P(d_j|z)}{\sum_{z' \in Z} P(z') \cdot P(t_i|z') \cdot P(d_j|z')} \quad (8)$$

The subsequent M-step maximizes the expectation of the complete data likelihood respectively to the model parameters, namely $P(z)$, $P(t_i|z)$ and $P(d_j|z)$.

$$\begin{aligned} P(d_j|z) &= \frac{\sum_{t_i \in V} P(z|d_j, t_i) \cdot w_{i,j}}{\sum_{t_i \in V} \sum_{d_{j'} \in D} P(z|d_{j'}, t_i) \cdot w_{i,j'}} \quad (9) \\ P(t_i|z) &= \frac{\sum_{d_j \in D} P(z|d_j, t_i) \cdot w_{i,j}}{\sum_{t_{i'} \in V} \sum_{d_j \in D} P(z|d_j, t_{i'}) \cdot w_{i,j'}} \quad (10) \\ P(z) &= \frac{\sum_{t_i \in V} \sum_{d_j \in D} P(z|d_j, t_i) \cdot w_{i,j}}{\sum_{t_i \in V} \sum_{d_j \in D} w_{i,j}} \quad (11) \\ & \quad (12) \end{aligned}$$

The EM algorithm starts with random values for the model parameters and converges by alternating E- and M-step to a local maximum of the likelihood.

Documents are compared with respect to the K-dimensional topic mixture distribution $(P(z_1|d_j), \dots, P(z_K|d_j))$. The components are derived by Bayes rule

$$P(z_k|d_j) = \frac{P(d_j|z_k)P(d_j)}{\sum_{k'=1}^{K} P(d_j|z_{k'})P(d_j)} \quad (13)$$

with $P(d_j) = \frac{sum_{i=1}^{n} w_{i,j}}{M}$.

Unlike the topic vectors found by LSI, namely the columns of U_k, the topic vectors by PLSI $(P(t_1|z), \dots, P(t_n|z))$ for different z are not forced to be orthogonal to each other. This gives PLSI more flexibility to approximate the empirical distribution $P(d_j, t_i)$.

2.4 Similarity score calculation and indexing of the HUPO PPP dataset

The HUPO PPP dataset was obtained from the PRIDE database[1], under accession numbers 4 to 98. These data sets are also accessible as PRIDE XML files via FTP[2].

All 95 Hupo PPP experiments and their corresponding peptides are directly taken from the PRIDE database and give a term/document matrix of the dimensions $25,052 \times 95$. Entries of the term/document matrix are weighted using IDF. In contrast to IR-applications on natural language, no further pruning of unique terms was performed. A close look at the term/document matrix reveals the following differences compared to natural language datasets: while in a corpus of natural language text documents (for example the TREC Spanish AFP collection or the TREC Volume 3 corpus) the amount of unique words is below 2% [3], the Hupo PPP data set has $14,808$ unique peptides (59%). This finding seems to contradict the intuitive assumption that proteomics experiments from the same tissue should yield highly similar results. The lack of reproducibility across proteomics experiments plays a considerable role in this divergence of the results [23]. This is illustrated for the HUPO PPP data by the fact that not even a single peptide is seen in every experiment. Moreover, only 37 peptides out of the $25,052$ peptides are found in at least half of the experiments. In contrast, more than 70% of the reported proteins are only found in one or two experiments. This effect can be further explained by the wide array of techniques applied by the HUPO PPP contributors, with the express purpose of enhancing coverage and allowing subsequent method evaluation [21]. Furthermore, a shotgun proteomics approach to analyze a complex mixture typically results in approximately 30% of all proteins identified by only a single peptide [20].

As a result, non-zero entries constitute less than 4% of the document/term matrix, which, although better than a typical natural language set, is still quite poor, especially considering the fact that we analyzed a small experiment corpus and that this data set describes the proteome of a single tissue, namely plasma. In order to analyze the ability of LSA to compensate for this sparseness of the document/term matrix, the similarity of pair of the 95 experiments is computed using different values for k and the standard cosine measure, which gives $95(95-1)/2 = 4,465$ similarity scores. The choice of k depends on the distribution of the singular values of the original document/term matrix. A rapid drop of the values in the sorted sequence of singular values (i.e. $\sigma_l - \sigma_{l+1}$ is large and σ_{l+1} is small) indicates that W_l is good approximation with low error. In our case, we used $k = 75$ for small degree of compression and $k = 15$ for high compression. For comparison, similarity scores are also directly computed from the vector space representation.

3 RESULTS

As expected, the distribution of the similarity scores obtained for the HUPO PPP experiments with the VSM shows a low overall similarity (93.5% pairs of experiments have a similarity less or equal than 0.1).

Even the similarities for k=75 show no drastic improvements; the similarity scores rise only slightly (88.5% pairs of experiments have a similarity less or equal than 0.1). However, with a strong dimensionality reduction (k = 15) of the HUPO PPP data, latent semantic relationships between terms as well as co-occurrences of peptides within the replicate experiments are amplified. Consequently, the inter-experiment similarities rise, resulting in the fact that now only 35% (in contrast to 93.5% for the VSM) of the experiments pairs have a similarity of 0.1 or less. This effectively compensates for the sparseness of the matrix with all entries are non-zero. To validate the results and show that LSA indeed resulted in meaningful experiment similarity, the obtained scores were visualized in a gray-scale map with white representing a score of 0, black a score of 1. The experiments are grouped by metadata, i.e. used technology (depletion step(s) applied; protein fractionation technique; search engine and finally mass spectrometer type). Since all 95 experiments originate from the same tissue, it

[1] http://www.ebi.ac.uk/pride

[2] ftp://ftp.ebi.ac.uk/pub/databases/pride

is reasonable to expect LSA to amplify the intra-similarities within the same technology group.

3.1 Influence of Technologies in the Hupo PPP dataset

A gray-scale map of the similarity scores between all 95 experiments, obtained after LSA with $k = 15$, is shown in figure 1. The experiments have been grouped according to the four abovementioned technologies, and these have been annotated above, below and to the left-hand side of the map. Obviously, an experiment is identical to itself, which is why the top-left to lower-right diagonal is black. A great amount of experiments have a high similarity (dark areas) as expected when looking on one single tissue and using a low k, although some experiments are less similar to the other experiments than expected. Those experiments form distinct clusters, each with high internal similarity (I, II and IV). The first cluster (I) represents only ESI FT-ICR[3] experiments. This uniqueness in the instrument used, together with the use of proprietary VIPER search engine most probably contributes to the dissimilarity from the rest of the experiments. Cluster (II) is derived from a set of 2D-PAGE experiments, and these can be compared to cluster (IV), because both represent experiments performed by the same lab with the same technology. The only difference is the use of the top-6 protein depletion[4] on the biological sample in cluster (IV), while no depletion was employed in cluster (II). The very low similarity between these two clusters (nearly white overlap regions) shows that removal of the six most abundant proteins in plasma resulted in the detection of an almost completely different part of the plasma proteome. Three experiments (III) have very low similarity with the other experiments combined with a low similarity between each other. The reason for this could be the combination of a CHO-affinity (aldehyde affinity) fractionation and the SEQUEST search engine which no other experiment employed. As all three experiments are from the same laboratory (which only contributed these three experiments) and they have a rather low similarity among themselves, it seems plausible that these experiments are outliers and even might indicate suspect results. Another group of experiments also sticks out (band V). This group comprises experiments that employed a peptide shotgun approach (with no protein separation technique) on top-6 depleted samples. The shotgun experiments thus reveal very little similarity, both within the repeated experiments as well as compared to the rest of the experiments. The low similarities of the three clusters (I,II,IV) and the shotgun experiments (none for separation technique) with all other experiments indicates that they contribute unique peptide identifications (i.e.: they cannot be semantically connected to other peptide identifications). However, in contrast to the shotgun experiments, the 2D-PAGE cluster (II,IV) are strongly internally consistent, hinting at a high reproducibility of the method. The total number of peptide identifications for the HUPO PPP data set reveal that shotgun experiments contributed a major part of the overall unique peptide identifications. Finally, all observed clusters in this analysis derive from differences introduced by the various methodologies and technology platforms employed, rather than from differences between the samples which shows that a strong bias is introduced.

3.2 Sample Analysis

A second experimental setup was used to evaluate the performance of LSA to compute meaningful similarities on proteomics data: instead of the natural grouping of peptides by experiment, all peptides found by any number of experiments of the same biological sample (plasma or serum) and anticoagulation treatment (EDTA, citrate or heparin) where selected and grouped, which resulted in a $5 \times 25,052$ document term matrix. Again, a VSM approach is not able to produce meaningful similarities, whereas LSA with k=2 (we would expect the document/term matrix to capture two semantic topics: plasma and serum) yields easily interpretable results. Those similarity scores are visualized and annotated in figure 3: VSM (on the left) only shows the (trivial) high similarities of one sample/anticoagulation group with itself, whereas LSA is able to resolve the similarity of the 4 groups of peptides originating from the plasma samples. Obviously the very low similarity between plasma and serum is caused by the fact that the serum samples do not contain any proteins (and therefore peptides) associated with clotting, such as Fibrin.

3.3 Interpretation of the peptide-based LSA

Literature suggests that a comparison based on exact matching of peptide sequences dramatically underestimates the overlap between experiments [19]. Therefore, it is particularly interesting to see how LSA groups peptides to latent topics, which are the dimensions of the latent space.

This analysis can be carried out by studying the term representation $U_k\Sigma_K$. A k-means clustering [25, 26] performed on the rows of $U_k\Sigma_K$ allows detection of these related terms. Upon analysis of the resulting groups, two distinct patterns emerge. First of all, LSA resolves peptides distinguished only by the occurrence of isobaric amino acids (e.g.: isoleucine/leucine). Since these amino acids are indistinguishable to the mass spectrometer, their substitution does not affect the semantic representation of the containing peptide sequence. Second, peptides that represent subsequences of longer peptides, either through missed cleavages (e.g.: YLGNATAIFFLPDEGK and YLGNATAIFFLPDEGKLQHLENELT), in-source decay or in vivo proteolytic degradation (e.g.: YLGNATAIFFLPDEGKLQHLENELT and YLGNATAIFFLPDEGKLQHLENELTHD) are all grouped together with the longer sequence. These two effects can be compared to synonyms in natural language.

3.4 Comparision with PLSA

We used also PLSA to determine the inter-experiment similarities of the Hupo PPP data set. This allows to validate the results obtained by LSA as well as to compare the performance of different models. Since PLSA cannot benefit from terms that appear only in a single document (unique peptides), the data set was reduced to peptides which occur at least in two experiments.

The number of latent topics for PLSA is set to $k = 5$, because the results of LSA showed 5 strong semantic topics, namely the two 2D-Page experiment-clusters, the Viper-search engine results, the shotgun-experiments, and the rest of the data set. The similarities between all pairs of experiments are visualized in a greyscale map in figure 2. The observed similarities cluster in an analogous way compared to LSA – the 2D-Page, viper search engine and shotgun experiments are dissimilar while the rest of the experiments

[3] EletroSpray Ionization Fourier-Transform Ion Cyclotron Resonance instrument

[4] The six most frequent, known proteins are removed.

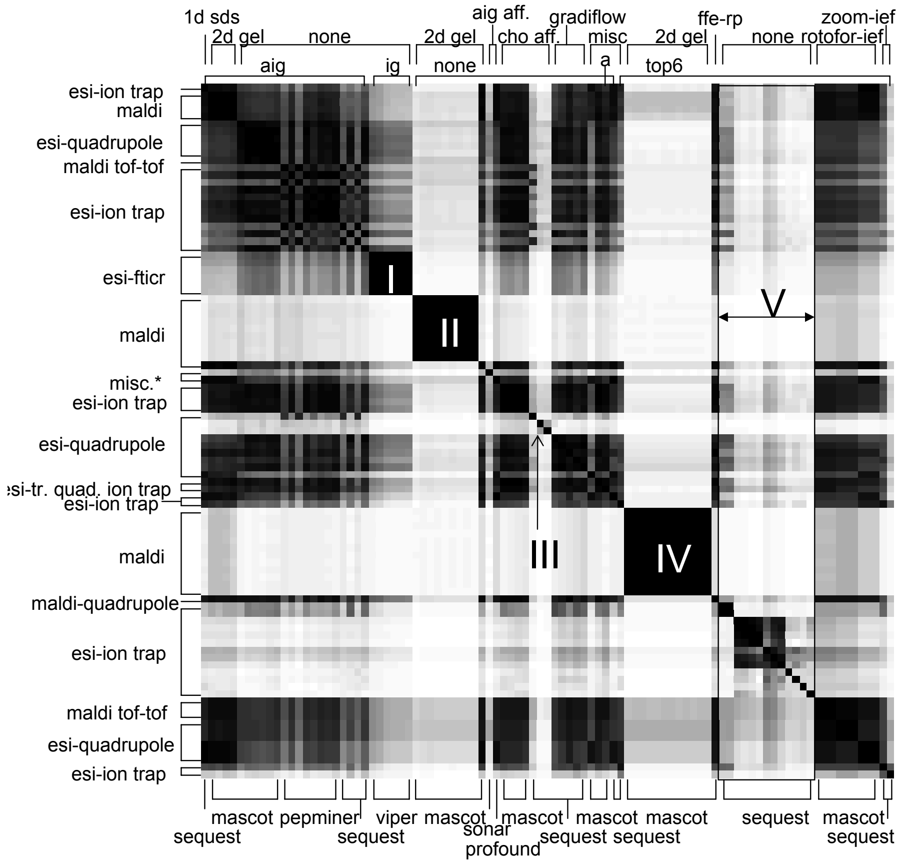


Figure 1: A visualization of the inter-experiment HUPO PPP similarity matrix, obtained from an LSA with $k = 15$. Dark indicates high similarity. The 95 experiments have been grouped by depletion technique, search engine, separation method and mass spectrometer (annotated above, to the left and below).

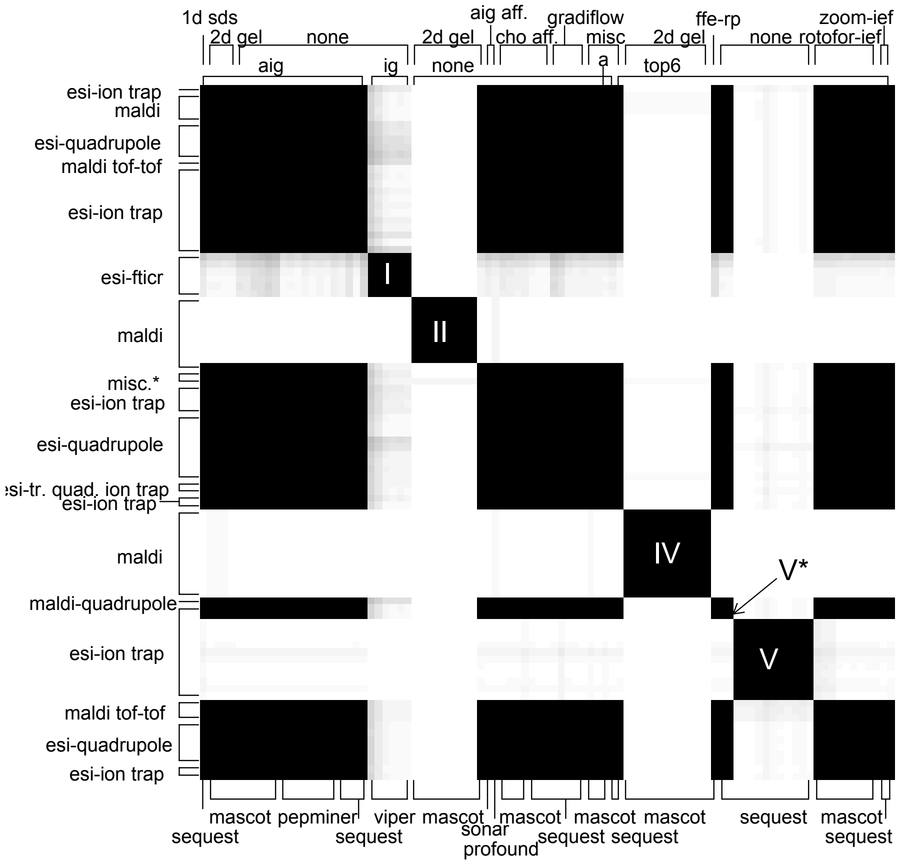


Figure 2: A visualization of the inter-experiment HUPO PPP similarity matrix, obtained from an PLSA with $K = 5$. Dark indicates high similarity. The 95 experiments have been grouped by depletion technique, search engine, separation method and mass spectrometer (annotated above, to the left and below).

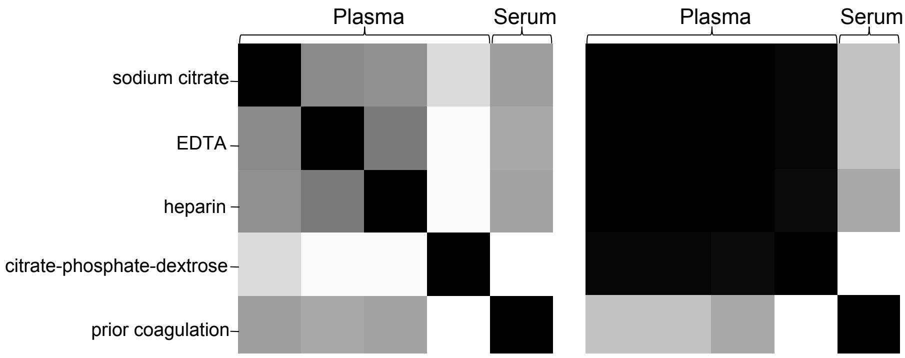


Figure 3: : HUPO sample similarity matrices, for the VSM on the left and LSA with k =2 on the right. This grayscale map visualizes the differences in similarities resulting from a VSM and LSA (k=2) analysis on all peptides from the HUPO PPP dataset. Peptides found by any of the 95 experiments are grouped by the biological sample type (plasma or serum) - annotated above - and the anticoagulation method used - annotated to the left.

have a high similarity among each other. This confirms the LSA results. A major improvement of PLSA is the clearer distinction between the clusters of experiments. Additionally, experiments within a cluster have a very high similarity in common (illustrated by the dark black colouring of the corresponding areas and absence of grey areas in figure 2). This is due to the facts that PLSA is more flexible in associating experiments with topics.

On the plasma proteome data, this results in a clear association of an experiment to exactly one latent topic. Thus an experiment has a high probability (> 0.9) to belong to one of the five topics, while the probability for other topics (other entries in $P(z|d)$) is very low. Note, that this clear association to a single topic is generally not the case for PLSA. Usually, an experiment is represented by a specific mixture of topics. The observed clear distinction between clusters of experiments once more illustrates the complementary nature of the proteomics technologies employed.

The second difference between PLSA and LSA is shown by the results concerning the misclassified experiment annotated as III in figure 1: PLSA associates those experiments to the remainder of the other experiments. This might disprove the previously assumption made after LSA that these experiments could be outliers. Also, the pruning of unique peptides could be responsible for the high similarity of the previously misclassified experiments to the rest. Interestingly, in repeated experiments with different starting configurations for the EM-algorithm, those 3 experiments where sometimes separated. This leaves the issue as an open question.

The third difference is constituted by the shotgun experiments: with LSA those experiments had a low similarity between each other, resulting from the fact that each of those experiments contributed a substantial high amount of new peptide identifications compared to other experiments while having only a small set of peptides in common. The unique peptides lead to a low similarity between the shotgun experiments, but now with PLSA those experiments show high similarity. The reasons are two-fold. The first is that unique peptides were pruned. This lead to a shortened vector space representation with regards to the number of rows of the document term matrix resulting in a higher percentage of peptides the experiments have in common. The second cause is again based on the nature of the PLSA method. Those experiments are represented by similar mixture proportions of the latent topics, even though their representation in vector space differs.

Interestingly two shotgun experiments (annotated as V*) are not similar to the other shotgun experiments, in contrast they are similar to the majority of the remaining Hupo PPP experiments. There are several explanations for this finding. Most likely this is caused by the underlying separation technique employed in these two experiments. While all experiments from the shotgun-experiments cluster (annotated as V) employed an additional scx-separation step on the peptides after tryptic digestion, the two dissimilar experiments just relied on RP-HPLC as a protein separation method. This findings correlate to the results in [16]. However, another explanation could be the fact that those two experiments originate from a different lab than the rest of the shotgun experiments. Therefore it seems plausible, that internal lab-specific optimisations/variations of protocol-steps which are not annotated or even falsely annotated in the source database yielded different results.

The comparison of LSA and PLSA illustrates the advantages of PLSA on proteomics data. The more principled approach of PLSA based on the statistical latent class model has a sound statistical foundation. The higher flexibility allows PLSA to generate more realistic association of experiments to topics. Furthermore, the choice of an appropriate value for k, the number of latent space dimension, directly correlates to the semantic topics captured by PLSA, whereas with LSA the choice of k was based more or less on trial and error and ad hoc heuristics.

However, PLSA also has some disadvantages. Namely the training procedure of the EM-Step could get stuck in a local optimum of the likelihood function which leads to distributions for $P(d|z)$, $P(z)$ and $P(w|z)$ with less quality and therefore a model of low accuracy. In our experiments, the EM-step of PLSA had major difficulties on the peptide/experiment matrix without pruning the unique peptides. In combination with LSA nevertheless, PLSA could be an important tool for analyzing proteomics data.

4 Discussion/Conclusion

We have demonstrated a novel application of LSA by comparing peptide lists derived from many different proteomics experiments performed on the same tissue. By applying LSA to the data from the Hupo PPP study, we were able to show that this method can handle the very diverse and heterogeneous data arising from proteomics experiments and compute meaningful similarities.

A large amount of experiments have a high similarity after LSA, which shows the strength of the method. However, some experimental setups, namely 2D-PAGE and shotgun approaches, strongly bias the set of observed peptides, which LSA cannot compensate for. Our results confirm visually, that if the goal of a project is to achieve maximal proteome coverage for a particular sample, shotgun proteomics experiments, repeated over multiple replicates achieve the most gain. 2D-PAGE analysis should not be disregarded as an analytical tool however, since it can complement a substantial fraction of unique identifications. Due to the high internal reproducibility of 2D-PAGE analyses as performed in the HUPO PPP, it seems that carrying out many replicates of this technology does not necessarily lead to a proportional increase in novel peptide identifications. In the specific case of plasma, the influence of various depletion techniques is also of interest. While methods employing top-6 depletion contributed more than 50% of the identifications, about 10% of all proteins were only found when no depletion was used at all.

The relatively simple task of comparing different sample types demonstrates that the fundamental difficulties arising from the origin of the data could be overcome through the utilization of an LSA analysis and its key principle of peptide/experiment association data representation in a lower dimensional 'latent space'. It is important to consider that the latent semantic analysis employed here greatly benefits from the large number of varying experimental repetitions on the same sample.

4.1 Interpretation of the semantic associations

An interesting finding is the ability of LSA to detect semantic relationships between apparently unrelated sets of peptide sequences, based solely on co-occurrences within experiments. We have found that at least some of these semantic links can be explained by underlying methodological or biological concepts and can be compared to synonyms found in natural language. The application of LSA to replicated shotgun experiments might help to alleviate one of the primary caveats of peptide-centric proteomics: the protein inference problem.

Since the semantic structures underlying protein lists (at least in part) represent entities of biological interest, the nature of the semantic relationships that occur at that level are also of considerable interest. Potential candidates of biological importance include protein complexes or protein components of the same pathway.

4.2 Future perspectives

It is clear from these findings that large collections of heterogeneous proteomics datasets can be mined relatively easily to obtain valuable information with LSA. The analysis carried out opens many paths for further investigations. By extending the analysis to include other tissue data sets (for instance the HUPO Brain Proteome Project (HUPO BPP) [9], and eventually any available proteomics data) and by carefully choosing an appropriate value for k, the focus of investigation could be shifted from the fine-grained effects resulting from the application of different technology platforms, to the coarse-grained distinctions derived from differences in tissue type, disease state or developmental stage.

It is of significant interest to get a better understanding of the semantic similarities peptide share in the latent semantic space resulting from singular value decomposition. Other methods, especially 'probabilistic latent semantic indexing' described in [12] which has a solid statistical foundation should be examined.

5 Acknowledgements

This work has been supported by the EU "ProDaC" grant number LSHG-CT-2006-036814 and the BBSRC "ISPIDER" grant.

References

[1] R. Aebersold and M. Mann. Mass spectrometry-based proteomics. *Nature*, 422:198–207, 2003.

[2] R. Craig, J. P. Cortens, and R. C. Beavis. Open source system for analyzing, validating, and storing protein identification data. *J Proteome Res*, 3:1234–1242, 2004.

[3] M. W. Davis and W. C. Ogden. Free resources and advanced alignment for cross-language text retrieval. *TREC*, pages 385–395, 1997.

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

[5] F. Desiere, E. W. Deutsch, A. I. Nesvizhskii, and P. Mallick. Integration with the human genome of peptide sequences obtained by high-throughput mass spectrometry. *Genome Biol*, 6:R9, 2005.

[6] B. Domon and R. Aebersold. Mass spectrometry and protein analysis. *Science*, 312:212–217, 2006.

[7] S. T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23:229–236, 1991.

[8] G. W. Furnas, S. Deerwester, S. Dumais, and T. K. Landauer. Information retrieval using a singular value decomposition model of latent semantic structure. *SIGIR '88: Proceedings of the 11th annual Int SIGIR conference on Research and Development in Information Retrieval*.

[9] M. Hamacher, R. Apweiler, G. Arnold, and A. Becker. Hupo brain proteome project: summary of the pilot phase and introduction of a comprehensive data reprocessing strategy. *Proteomics*, 6:4890–4898, 2006.

[10] H. Hermjakob and R. Apweiler. The proteomics identifications database (pride) and the proteomexchange consortium: making proteomics data accessible. *Expert Rev Proteomics*, 3:1–3, 2006.

[11] T. Hofmann. Probabilistic latent semantic analysis. *Uncertainity in Artificial Intelligence*.

[12] T. Hofmann. Probabilistic latent semantic indexing. *Proceedings of the 22nd Annual Int SIGIR Conference on Research and Development in Information Retrieval*.

[13] P. Jones, R. G. Cote, L. Martens, and A. F. Quinn. Pride: a public repository of protein and peptide identifications for the proteomics community. *Nucleic Acids Res*, 34:D659–663, 2006.

[14] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoustics, Speech and Signal Processing*, 35:400–401, 1987.

[15] T. K. Landauer, P. W. Foltz, and D. Laham. Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[16] X. Li, Y. Gong, Y. Wang, S. Wu, Y. Cai, P. He, Z. Lu, W. Ying, Y. Zhang, L. Jiao, H. He, Z. Zhang, F. He, X. Zhao, and X. Qian. Comparison of alternative analytical techniques for the characterisation of the human serum proteome in hupo plasma proteome project. *Proteomics*, 5:3423–3441, 2005.

[17] Mallick, Schirle, Chen, and Flory. Computational prediction of proteotypic peptides for quantitative proteomics. *Nat Biotechnol*, 25:125–131, 2007.

[18] L. Martens, H. Hermjakob, P. Jones, and M. Adamski. Pride: the proteomics identifications database. *Proteomics*, 5:3537–3545, 2005.

[19] L. Martens, M. Muller, C. Stephan, and M. Hamacher. A comparison of the hupo brain proteome project pilot with other proteomics studies. *Proteomics*, 6:5076–5086, 2006.

[20] A. I. Nesvizhskii and R. Aebersold. Interpretation of shotgun proteomic data: The protein inference problem. *Mol Cell Proteomics*, 4:1419–1440, 2005.

[21] G. S. Omenn, D. J. States, M. Adamski, and T. W. Blackwell. Overview of the hupo plasma proteome project: results from the pilot phase with 35 collaborating laboratories and multiple analytical groups, generating a core dataset of 3020 proteins and a publicly-available database. *Proteomics*, 5:3226–3245, 2005.

[22] J. Prince, M. W. Carlson, R. Wang, P. Lu, and E. M. Marcotte. The need for a public proteomics repository. *Nat Biotechnol*, 22:471–472, 2004.

[23] K. A. Reidegeld, M. Muller, C. Stephan, and M. Bluggel. The power of cooperative investigation: summary and comparison of the hupo brain proteome project pilot study results. *Proteomics*, 6:4997–5014, 2006.

[24] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the Acm*, 18:613–620, 1975.

[25] A. Smellie. Accelerated k-means clustering in metric spaces. *Journal of Chemical Information and Computer Sciences*, 44:1929–1935, 2004.

[26] D. Steinley. K-means clustering: A half-century synthesis. *British Journal of Mathematical & Statistical Psychology*, 59:1–34, 2006.

U*C: Distance and Density Clustering based on Grid Projections

Alfred Ultsch
Databionics Research Group
University of Marburg, Germany
ultsch@informatik.uni-marburg.com

Abstract

A new clustering algorithm based on grid projections is proposed. This algorithm, called U*C, uses distance information together with density structures. The number of clusters is determined automatically. The validity of the clusters found can be judged by the U*-Matrix visualization on top of the grid. A U*-Matrix gives a combined visualization of distance and density structures of a high dimensional data set. For a set of critical clustering problems the superiority of U*C clustering compared to standard clustering algorithms such as K-means and hierarchical clustering is shown.

1 Introduction

Projections of high dimensional data onto two dimensions have the advantage that the structures of the data can be visually inspected. Some projections, in particular the neighborhood preserving projections, like SOM, can be used to visualize cluster structures of the high dimensional data space. We assume here a projection from data space onto a finite grid of points. ESOM (Ultsch 1999) are such a projection. Other projections like PCA, ICA or MDS can also be used if their output is discretized to a grid. ESOM have the property to disentangle cluster structures that are linear not separable. The ChainLink example (see Fig. 1) was among the first to illustrate this (Ultsch and Vetter 1994). Distance relationships in a high dimensional data space can be visualized on top of a grid in form of a U-Matrix (Ultsch 1990). The recently introduced P-Matrix allows the visualization of density structures of the high dimensional data space, see Ultsch 2003). In this paper we present a combined visualization of distance and density in form of the U*-Matrix (see Chapter 2). In Chapter 3 we define a novel clustering algorithm, called U*C, which uses distance and density information.. U*C is tested on some crucial clustering problems (Chapter 4). Its performance is compared to K-means and popular hierarchical clustering algorithms (Chapter 5). The validity and precision of the resulting clusters can be judged using the U*-Matrix. The performance of U*C is discussed in Chapter 6.

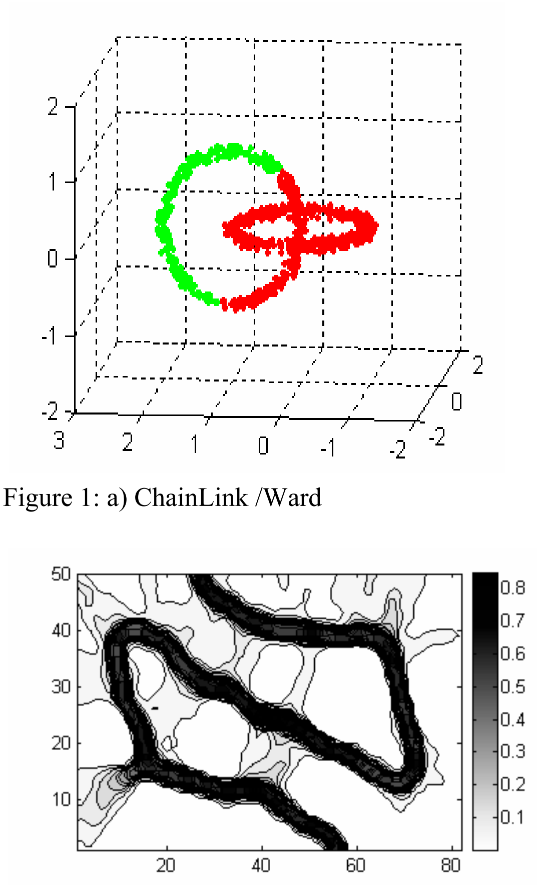


Figure 1: a) ChainLink /Ward

b) U-Matrix showing disentangled clusters

2 Visualizations of Distance and Density

Let m: D-> G be a mapping from a high dimensional data space $D \subset R^n$ onto a finite set of positions $M = \{n_1, \dots , n_k\} \subset R^2$ arranged on a grid. Each position has its two dimensional coordinates and aweight vector $w_j \in W = \{w_1,\dots,w_k\}$ which is the image of a Voronoi region in D: the data set $E = \{x_1,\dots,x_d\}$ with $x_i \in D$ is mapped to a position in M such that a data point x_i is mapped to its bestmatch $bm(x_i)= n_b \in M$ with $d(x,w_b) \leq d(x, w_j)\ \forall\ w_j \in W$, where d is the distance on the data set. The set of immediate neighbors of a position n_i on the grid is denoted by N(i). We call m a grid projection if m has minimal, or at least tolerable, topological errors. Compare Ultsch and Herrmann (2005) for grid dimensions and surface structures to minimize projection errors. The U-height for each position n_i is the average distance of n_i's weight vectors to the weight vectors of its immediate neighbors N(i). The U-height uh(i) is calculated as follows:

$$uh(i) = \frac{1}{n}\sum_{j} d(w_i, w_j)$$

j ∈ N(i), n = |N(i)|.

A display of all U-heights on top of the grid is called a U-Matrix (Ultsch (1990). A single U-height shows the local distance structure. The local average distance at wi is shown at position ni. The overall structure of densities emerges, if a global view of a U-Matrix is regarded. Figure 1b shows an example of a U-Matrix on an ESOM with grid size 50x82. Recently the P-Matrix has been introduced (Ultsch 2003). The P-Matrix also uses the ESOM map as floor space layout. This makes P-Matrix compatible with an U-Matrix. Instead of the local distances, however, density values in data space measured at the neuron's weights are used as height values. The P-height of a neuron n, with associated weight vector w(n), is defined as:P-height(n) = p(w(n),X),where p(x,X) is an empirical density estimation at point x in the data space X. For each neuron n of a ESOM the P-Matrix displays the density measured in the data space at point w(n), where w(n) is the weight vector associated with neuron n of the ESOM. In principle any density estimation, which works for the input data set of the ESOM can be used. See [10] for an overview on density estimations for multivariate data. Here we use a density estimation called Pareto Density Estimation (PDE) (Scott, D.W. (1992)). PDE calculates the density estimation at some point x as the number of points inside a hypersphere (Pareto sphere) around x. The radius of the hyperspheres is called the Pareto radius. The Pareto radius is derived from information optimal sets. A P-Matrix is defined in the same manner as an U-Matrix. The U-Matrix reveals the (local) distance structures, while the P-Matrix gives insights into the density structures of a high dimensional data set. The elements of a P-Matrix are called P-heights. These are the number of points inside the Pareto sphere.Properties of a P-Matrix are:

- the position of the projections of the data on the ESOM reflect the topology of the input space, this is inherited from the underlying SOM algorithm
- neurons with large P-heights are situated in dense regions of the data space
- neurons with small P-height are "lonesome" in the data space
- outliers in the input space are found in „funnels".
- "ditches" on a P-Matrix point to cluster boundaries
- „plateaus" on a P-Matrix point to regions with equal densities

One can see, that many, but not all, properties of the P-Matrix are the inverse of an U-Matrix display. In contrast to the U-Matrix, which is based on the distance structure of the data space, the P-Matrix is based on the data's density structure. This gives a new and complementary insight into the high dimensional data space.The P-height ph(i) for a position ni is a measure of the density of data points in the vicinity of wi : ph(i) =|{x ∈ E | d(x, w_j) < r >0, r ∈ R}|. A display of all P-heights on top of the grid G is called a P-Matrix (Ultsch 2003). Figure 3b, 4b and 5b show examples of P-Matrices. The P- height is the number of data points within a hypersphere of radius r. The radius r should be chosen such that ph(i) approximates the probability density function of the data points. The usage of the ParetoRadius, as described in Ultsch 2003), is a good choice for r. Median filtering within the N(i) window can be applied to the P-Matrix. This reduces local fluctuations (noise) in the density estimation without disturbing the overall picture. Such a filtering preserves, however, the density gradients important for clustering. For the identification of clusters in data sets it is sometimes not enough to consider distances between the data points.

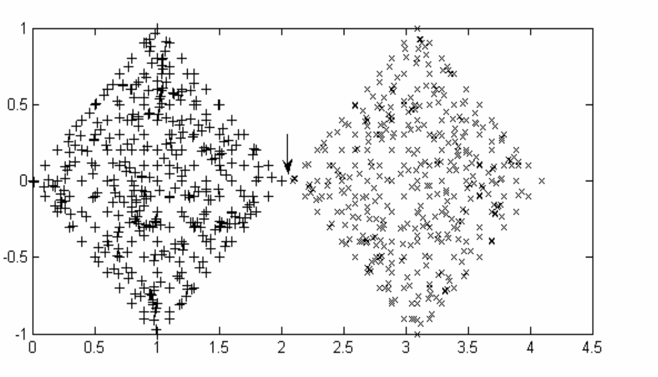


Figure 2: TwoDiamonds data set

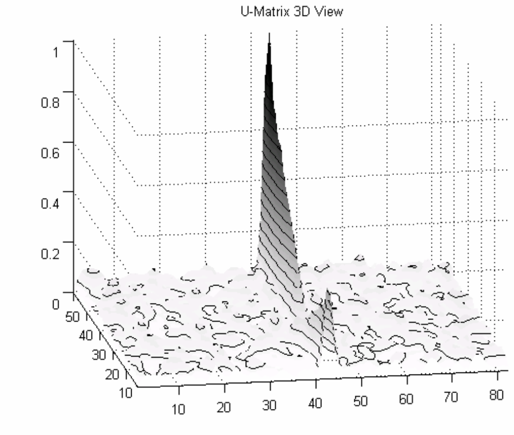
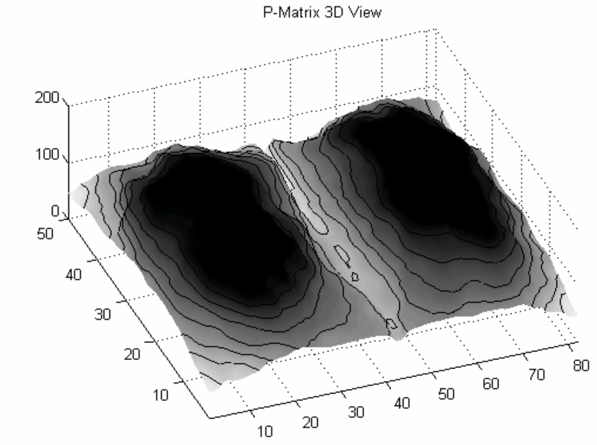


Figure 3: a) U-Matrix of TwoDiamonds



3b) P-Matrix of TwoDiamonds

Consider, for example, the TwoDiamonds data set depicted in Figure 2. The data consists of two clusters of points on a plane. Inside each "diamond" the values for each data point were drawn independently from uniform distributions. At the central region, marked with an arrow in Figure 2, the distances between the data points are very small. For distance based cluster algorithms it is hard to detect correct boundaries for the clusters. Clustering methods such as single linkage, complete linkage, Ward and others, that rely on the density structures of the data produce classification errors. The picture changes, however, when the data's density is regarded (see Figures 2, 3 and 5). The density at the touching point of the two diamonds is only half as big as the densities in the center regions of the clusters.

As the TwoDiamonds data set shows, a combination of distance relationships and density relationships is necessary to acieve an appropriate clustering in this case.

In dense regions of the data space the local distances depicted in an U-Matrix are presumably distances measured inside a cluster. Such distances may be disregarded for the purpose of clustering. In thin populated regions of the data space, however, the distances matter. In this case the U-Matrix heights correspond to cluster boundaries. This leads to the definition of an matrix which combines the distance based U-Matrix and the density based P-Matrix. The combination of a U-Matrix and a P-Matrix is called U*-Matrix.

The U*-Matrix is derived from an U-Matrix following these lines:

- when the data density around a weight vector of a neuron is equal to the average data density, the heights shown in an U*-Matrix should be the same as in the corresponding U-Matrix.
- when the data density around a weight vector of a neuron is big, local distances are primarily distances inside a cluster. In this case the U*-Matrix heights should be low.
- when the data density around a weight vector of a neuron is lower than average, local distances are primarily distances at a border of a cluster. In this case the U*-Matrix heights should be higher than the corresponding U-height.

The U*-height u*h(i) for a position n_i is the U-height multiplied with the probability that the local density, as measured by ph(i), is low. As an estimate of this probability the empirical density function can be used:

$$\mathrm{plow(i)} = Pr(\text{data density is low for position } n_i)$$

$$\simeq \frac{\left|\{\mathrm{p} \in \mathrm{P\text{-}matrix} \mid \mathrm{p} > \mathrm{ph(i)}\}\right|}{\left|\{\mathrm{p} \in \mathrm{P\text{-}matrix}\}\right|} \quad \text{(i).}$$

Plow(i) is a measure for the probability that the density in data space is low at position ni. The U*height is then calculated as u*h(i) = uh(i) . plow(i). If the local data density is low: u*h(i) = uh(i). This happens at the presumed border of clusters. If the data density is high, then u*h(i) = 0. This occurs in the central regions of clusters. For positions with median density holds: u*h(i) = uh(i)*0.5. The U*-Matrix exhibits therefore the local data distances as heights, when the data density is low (cluster border). If the data density is high, the distances are scaled down to zero (cluster center). An alternative to formula (i) is, to adjust the multiplication factor such that u*h(i) = uh(i) for median P-heights and u*h(i) = 0 for the top 20 percent of P-heights (Ultsch 2003).

The cluster structure of the data can be seen more clearly on the U*-Matrix than on the U-Matrix. Figure 4 compares, for example, the U-Matrix taken from Kaski et al 1999 to a U*-Matrix of the same data set. Since density and distance play different roles in the definition of clusters, we think that the three different matrices, U-, P- and U*-Matrix together give an appropriate impression of the cluster structure of any high dimensional data (see Fig. 4).

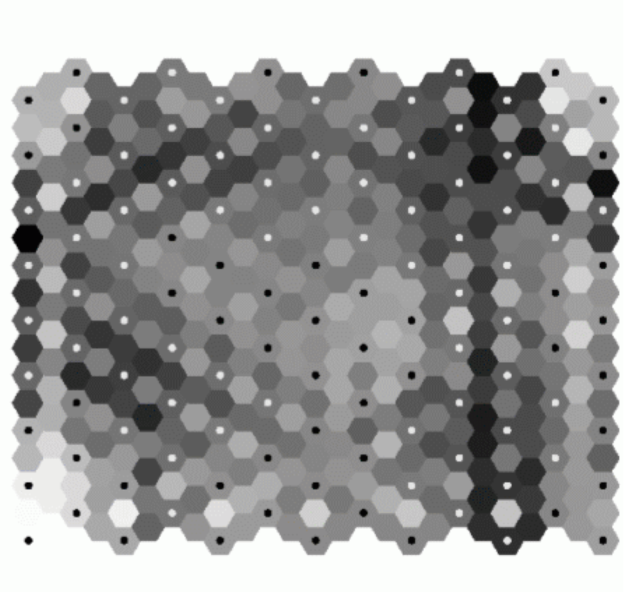
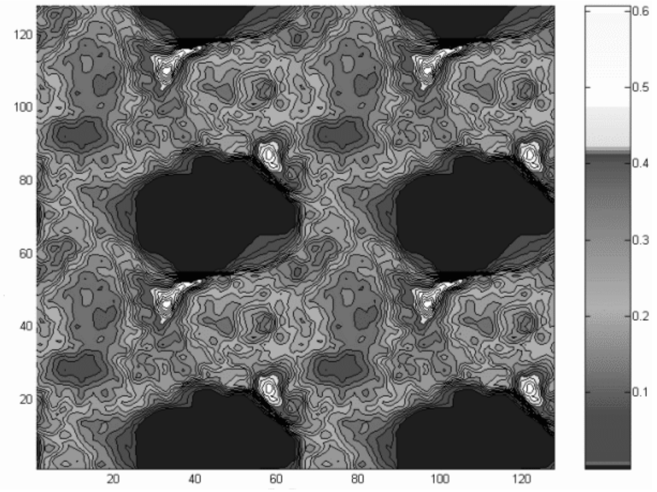


Figure 4: a) U-Matrix/ Kaski et al (1999),



4 b) U*-Matrix, same data.

For the U-, P- and U*-Matrix the grid (also called the map) is the floor space for a landscape like visualization of distance- and density structures of the high dimensional data space. Structures emerge on top of the grid by regarding an overall view of many positions. Single positions are only tiny parts of these structures (Ultsch (1999)). The U-,U*- and P-Matrix visualizations of the data are called emergent grid projections (EGP).

3 U*C Clustering Algorithm

A topological correct mapping m projects a cluster onto a coherent area on the grid (cluster area). Points within the cluster are mapped to the inside of the cluster area. Data points at the border (surface) of the cluster are projected to the border of the cluster area. Consider a data point x at the surface of a cluster C, with ni = bm(x). The weight vectors of its neighbors N(i) are either within the cluster, in a different cluster or interpolate between clusters. If we assume that the inter cluster distances are locally larger than the inner cluster distances, then the U-heights in N(i) will be large in such directions which point away from the cluster center. This means, a gradient descent on the U-Matrix will lead away from cluster borders. A movement from one position ni to another position nj with the result that wj is more within a cluster C than wi is called immersive. For data points well within C, a gradient descent on a U-Matrix will, however, not necessarily be immersive. The P-heights follow the density structure of a cluster. Under the assumption that the core parts of a cluster are those regions with largest density, a gradient ascent on the P-Matrix is immersive. Clusters may also be defined by density alone instead of distance. See, for example, the EngyTime data set shown in Figure 5a and its density structure as shown by the P-Matrix in Figure 5b. This data set represents situations where the data generation can be described appropriately by Gauss Mixture Models.

At the borders of a cluster the measurement of density is, however, critical. At cluster borders the local density of the points should decrease substantially. In most cases the cluster borders are defined either by low point densities (see Figure 5) or by “empty space” between clusters (= large inter cluster distances). For empirical estimates of the point density a gradient ascent on a P-Matrix may therefore not be immersive for points at cluster borders. An immersion is a movement on a grid which is composed of a gradient descent on the U-Matrix followed by a gradient ascent on the P-Matrix will be continuosly immersive. Let I denote the end points of immersion starting from every position on a grid. If the density within a cluster is constant, immersion will not converge to a single point for a cluster for all starting points within a cluster.

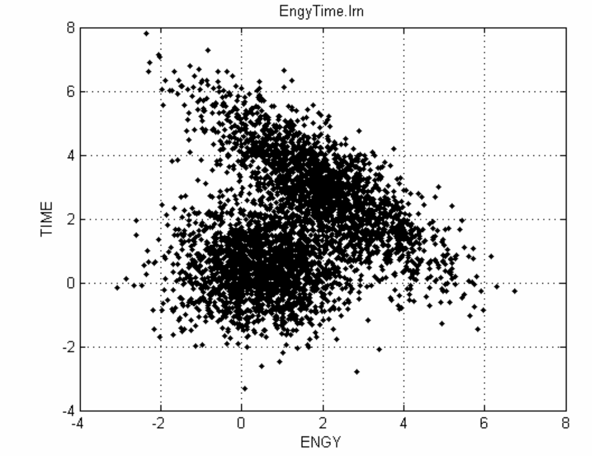
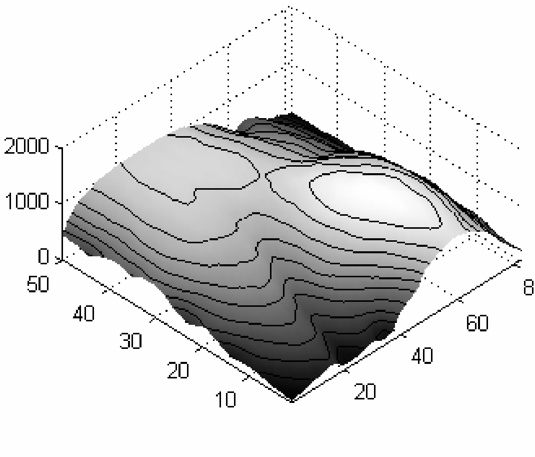


Figure 5: a) EngyTime/SingleLinkage clustering



5 b) P-Matrix of EngyTime

The U*-Matrix is then used to determine which points in I belong to the same cluster. The watersheds of the U*-Matrix are calculated using the algorithm described in Luc/Soille (1991). Points that are separated by a watershed are assigned to different clusters, points within the same basin to a single cluster. The following pseudocode summarizes the U*C clustering algorithm described above.

```
U*C clustering Algorithm:  given U-Matrix, P-Matrix, U*-Matrix, I = {};
Immersion:  For all positions n of the grid:
    1) from position n follow a gradient descent on the U-Matrix until a minimum is reached in position u
    2) from position u follow a gradient ascent on the P-Matrix until a maximum is reached in position p.
    3) $I = I \cup \{p\}$;  Immersion(n) = p.
Cluster assignment:
    1) calculate the watersheds for the U*-Matrix ( e.g. using Luc/Soille (1991)).
    2) partition I using these watersheds into clusters $C_1, \ldots C_c$
    3) assign a data point x  to a cluster $C_j$ if  Immersion(bm(x) ) $\in C_{j.}$
```

4 Fundamental Clustering Problems Suite

The efficiency of the U*C clustering algorithm is tested using a set of ten clustering problems called Hepta, Lsun, Tetra, Chainlink, Atom, EngyTime, Target, TwoDiamonds, WingNut and GolfBall. Any reasonable clustering algorithm (see e.g. Jain and Dubes 1998) should be able to solve these problems correctly. As can be seen below, however, standard algorithms like K-means, and hierarchical clustering algorithms, like single linkage and Ward have severe difficulties on several data sets. The suite of data sets is called Fundamental Clustering Problem Suite (FCPS). The suite can be downloaded from the website of the author(http://www.uni-marburg.de/fb12/datenbionik/).

FCPS poses some hard clustering problems. Chainlink and Atom are not separable by hyperplanes. The GolfBall data set consists of points that are equidistant on the surface of a sphere. This data set is used to address the problem to impose cluster structures when no such structure is present. The problem of outliers is addressed by the Target data set shown in Figure 6.

The following is a short description of the data set and the problem it poses to cluster algorithms. Pictures of the data sets are shown in the Figures of this paper.

FCPS data sets

		Nr of	**Nr. of**
Name	**Cases**	**Variables**	**Clusters**
Hepta	212	3	7
Lsun	400	2	3
Tetra	400	3	4
Chainlink	1000	3	2
Atom	800	3	2
EngyTime	4096	2	2
Target	770	2	6
TwoDiamonds	800	2	2
WingNut	1070	2	2

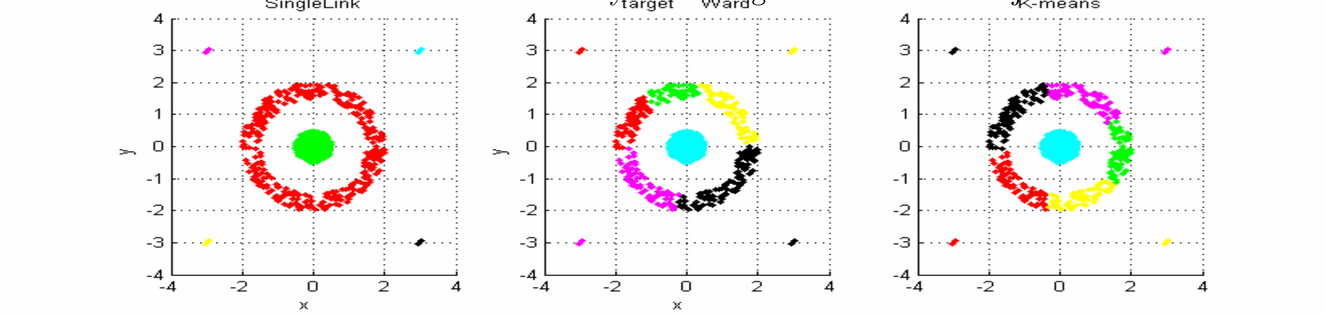


Figure 6: Single Linkage, Ward and K-means clustering algorithms on the Target data set

FCPS Problems

Name	Main Clustering Problem
Hepta	different densities in clusters
Lsun	different variances in clusters
Tetra	large inner dist. vs. small inter
Chainlink	not separable by linear surf.
Atom	lin. not sep., diff. dens./variances
EngyTime	density defined clusters
Target	outliers
TwoDiamonds	touching clusters
WingNut	largest densities at cluster borders
GolfBall	equidistant points, no cluster at all

5 Results

The results of U*C Clustering are compared to K-means as the most popular partitioning cluster algorithm. The hierarchical cluster algorithms SingleLinkage and Ward were also applied to the FCPS data sets. All algorithms were used as implemented in MATLAB™. Since K-means converges to a local minimum of a cost function, the best of 100 repetitions with random initializations is reported. The correct number of clusters was given as parameter to all standard algorithms. Shown are the overall accuracies. Performances lower than 80% are emphasized. There is no data set on which U*C performs worse than any of the other clustering algorithms.

Data Set	Single Linkage	Ward	K- neans	U*C
Hepta	100 %	100 %	100 %	100 %
Lsun	100 %	50 %	50 %	100 %
Tetra	0.01 %	90 %	100 %	100 %
Chainlink	100 %	50 %	50 %	100 %
Atom	100 %	50 %	50 %	100 %
EngyTime	0 %	90 %	90 %	90 %
Target	100 %	25 %	25 %	100 %
TwoDiamonds	0 %	100 %	100 %	100 %
WingNut	0 %	80 %	80 %	100 %
GolfBall	100 %	50 % (best)	?	100 %

Table 1: Accuracy of the clustering algorithms on FCPS

SingleLinkage (SL) clustering imposes a chain of data points as cluster model on the data. This algorithm is usually misled, if the local inner cluster distances are in the same range as the inter cluster distances. This can be seen separate clusters that are not separable by hyperplanes, e.g. ChainLink and Atom. The fundamental cluster model of Ward is a hyperellipsoid. For data sets which do not fit this model (5 of the 10 sets in FCPS), Ward produces an erroneous clustering. Most clustering algorithms require the knowledge of the number of clusters. U*C determines the number of clusters automatically. It is, however, crucial to assess the validity of a clustering. Dendrograms are comonly used in hierarchical clustering algorithms to address this. They are, however, sometimes misleading as the following example shows. Figure 7a shows a dendrogram of the GolfBall data using Ward hierarchical clustering. Such a dendrogram would suggest 3 or 6 cluster.

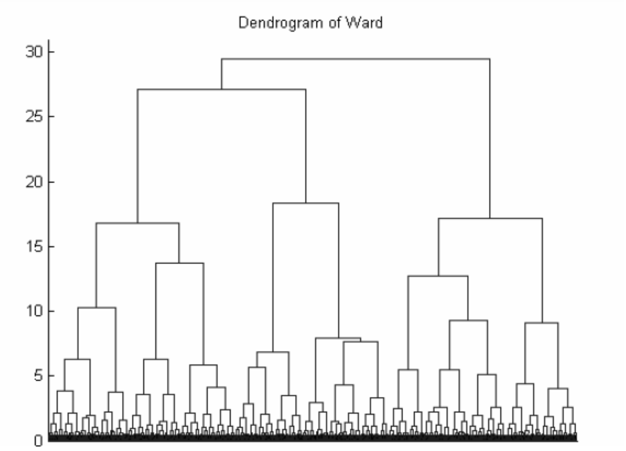
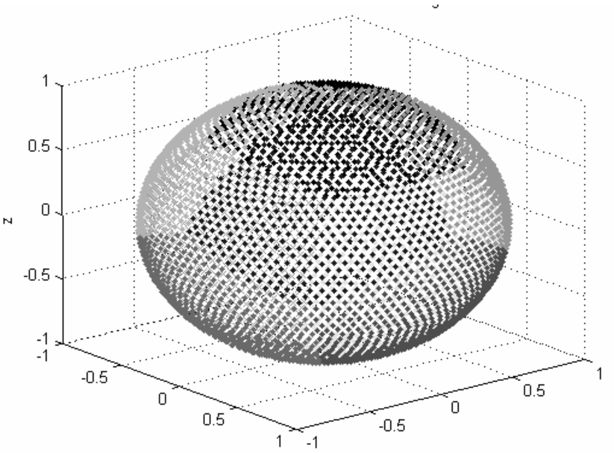


Figure 7: a) GolfBall: Ward Clustering Dendrogram



7b) Golf Ball / K-means

6 Conclusion

A new clustering algorithm called U*C based on grid projections is proposed. This algorithm uses distance structures (U-Matrix) as well as density structures (P-Matrix) of the data set. No particular geometrical cluster model is imposed on the data by U*C. Other clustering algorithms impose such a model and are performing poor, if the data set is of a different structure. The number of clusters is determined automatically in U*C. The correctness and validity of the clusters found can be assessed directly using the U*-Matrix visualization. The U*-Matrix shows a combined picture of distance and density structures of a high dimensional data set. U*C performs superior to standard clustering algorithms such as K-means and the most popular hierarchical algorithms. This is demonstrated on a group of data sets which represent fundamental clustering problems, like different variances, outliers and other structural difficulties. U*C and other tools for ESOM (see Ultsch and Mörchen 2005) can be downloaded from our web site.

References

Jain, A.K., Dubes, R.C. (1998): Algorithms for Clustering Data, New York, Wiley.

Kaski S., Nikkilä,J., Törönen,P., Castrén,E., Wong, G. (1999): Analysis and Visualisation of Gene Expression Data using Self Organizing Maps, Proc NSIP.

Kohonen, T. (1992), “Self-Organized formation of topologically correct feature maps”, Biological Cybernetics, Vol.43, pp.59-69, 1982.

Luc,V, Soille, P. (1991): Watersheds in Digital Spaces: An Efficient Algorithm Based on Im-mersion Simulations, IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol. 13(6), 583-598.

Rand, W. M. (1971): Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66, 846–850.

Scott,D.W. (1992): “Mulivariate Density Estimation”, Wiley-Interscience, .

Ultsch, A., Mörchen,F. (2005): ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM, Dept. of Computer Science University of Marburg, Research Report. 46.

Ultsch, A., Vetter, C. (1994): Selforganizing Feature Maps vs. statistical clustering, Dept. of Computer Science University of Marburg, Research Report 9.

Ultsch,A. (2003): Maps for the Visualization of high-dimensional Data Spaces, In Proc. WSOM, Kyushu, Japan, 225-230.

Ultsch,A., (1999): Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series, E. Oja and S. Kaski (eds), Kohonen Maps, 33-46.

Ultsch,A., (2003): U*-Matrix: A Tool to visualize Clusters in high dimensional Data, Dept. of Computer Science University of Marburg, Research Report 36.

Ultsch,A., Herrmann, L. (2005), The architecture of Emergent Self-Organizing Maps to reduce projection errors, Proc ESANN, Brugges 2005.

Ultsch,A., Siemon, H.P. (1990): Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis, In Proc. Intern. Neural Networks, Kluwer Academic Press, Paris, 305-308.

Vesanto J. et al.,(1999): ”Self-organizing map in Matlab: the SOM toolbox”, Proceedings of the Matlab DSP Conference, pp 35--40, Espoo, Finland, November,

Towards Confounding-Aware Subgroup Discovery

Martin Atzmueller and **Frank Puppe**
Department of Computer Science,
University of Würzburg, Germany
{atzmueller, puppe}@informatik.uni-wuerzburg.de

Abstract

This paper presents a semi-automatic approach for confounding-aware subgroup discovery: We present a method that provides the means for detecting potentially confounded subgroup patterns, other unconfounded relations, and/or patterns that are affected by effect-modification.

Since there is no purely automatic test for confounding, the discovered relations are subsequently presented to the user in a semi-automatic approach: Then, the interesting relations can be evaluated and validated interactively. Furthermore, we show how to utilize (causal) domain knowledge for improving the results of the automatic algorithm, since confounding is itself a causal concept. The applicability and benefit of the presented technique is illustrated by examples from a case-study using data from a fielded system in the medical domain.

1 Introduction

Subgroup discovery (e.g., [Wrobel, 1997; Klösgen, 1996; 2002; Lavrac *et al.*, 2004; Atzmueller *et al.*, 2005]) is a powerful approach for explorative and descriptive data mining for obtaining interesting relations between a specific target (dependent) variable and usually many explaining (independent) variables describing the respective subgroups. The interesting subgroups can be defined as subsets of the target population with a (distributional) unusualness concerning a certain property we are interested in. The risk of coronary heart disease (target variable), for example, is significantly higher in the subgroup of smokers with a positive family history than in the general population.

When interpreting and applying the discovered subgroup patterns, it is often necessary to consider the patterns in a causal context. However, considering an association as having a causal interpretation can often lead to incorrect results, due to the basic tenet of statistical analysis that association does not imply causation (cf., [Cooper, 1997]). The estimated *effect*, i.e., the quality of the subgroup may be due to associations with other *confounding* factors that were not considered in the quality computation.

Basically, confounding can be described as a bias in the estimation of the effect of the subgroup on the target concept due to attributes affecting the target concept that are not contained in the subgroup description [McNamee, 2003]. For instance, the quality of a subgroup may be confounded by other variables that are associated with the independent variables, and are a direct cause of the (dependent) target variable. In this case, it is necessary to identify potential confounders, and to measure or to control their influence concerning the subgroup and the target concept.

Let us assume, for example, that ice cream consumption and murder rates are highly correlated. However, this does not necessarily mean that ice cream incites murder or that murder increases the demand for ice cream. Instead, both ice cream and murder rates might be joint effects of a common cause or confounding factor, namely, hot weather.

Confounding factors can provide a significant bias with respect to the discovered patterns, such that the interestingness of a pattern may be decreased significantly, or the interestingness of a former non-interesting pattern may be significantly increased. To the best of the authors' knowledge, the issue of confounding has unfortunately not received much attention in data mining research so far, while handling confounding is a crucial step for certain domains, e.g., for the medical domain.

In this paper, we propose a semi-automatic subgroup discovery approach for detecting a set of potentially confounded candidate subgroup patterns besides other unconfounded relations, and/or patterns affected by effect-modification. Since there is no purely automatic test for confounding, the discovered candidate relations are presented to the user in a semi-automatic approach: Then, these can be analyzed in more detail using interactive visual techniques.

Furthermore, we utilize (causal) domain knowledge for improving the results of the algorithm, since confounding is itself a causal concept: We apply knowledge about causal relations between attributes that can be incrementally refined: We consider attributes that are *acausal*, i.e., have no causes, and attributes that are known to be directly causally related to other attributes. Additionally, both concepts can be combined, e.g., in the medical domain certain variables such as *Sex* have no causes, and it is known that subgroup patterns containing these are causal risk factors for certain diseases. The applicability and benefit of the presented technique is illustrated by examples from a case-study in the medical domain.

The rest of the paper is organized as follows: First, we discuss the background of subgroup discovery, the concept of confounding, and a method for controlling confounding in Section 2. After that we present the semi-automatic approach for confounding-aware subgroup discovery in Section 3. Exemplary results of a case study applying the presented approach are given in Section 4 using data from a fielded system in the medical domain. Finally, Section 5 concludes the paper with a discussion of the presented work and presents promising directions for further work.

2 Background

In this section, we first introduce the necessary notions concerning the used knowledge representation and define the setting for subgroup discovery. After that, we introduce the concept of confounding, criteria for its identification, and describe the basic stratification technique for analyzing and controlling confounding.

2.1 Subgroup Discovery

The main application areas of subgroup discovery (e.g., [Wrobel, 1997; Klösgen, 2002; Lavrac *et al.*, 2004; Atzmueller *et al.*, 2005]) are exploration and descriptive induction, to obtain an overview of the relations between a (dependent) target variable and a set of explaining (independent) variables. As in the *MIDOS* approach [Wrobel, 1997], we consider subgroups that are, e.g., as large as possible, and have the most unusual (distributional) characteristics with respect to the concept of interest given by a binary target variable. Therefore, not necessarily complete relations but also partial relations, i.e., (small) subgroups with "interesting" characteristics can be sufficient.

Subgroup discovery mainly relies on the subgroup description language, the quality function, and the search strategy. Often, heuristic methods based on beam-search, e.g., [Lavrac *et al.*, 2004] but also efficient exhaustive algorithms, e.g., the SD-Map algorithm [Atzmueller and Puppe, 2006], are applied. The description language specifies the individuals (cases) belonging to the subgroup.

With respect to the applied knowledge representation, let Ω_A denote the set of all attributes. For each attribute $a \in \Omega_A$ a range $dom(a)$ of values is defined. We consider nominal attributes so that numeric attributes need to be discretized accordingly. Let CB be the case base (data set) containing all available cases (instances): A case $c \in CB$ is given by the n-tuple $c = ((a_1 = v_1), (a_2 = v_2), \ldots, (a_n = v_n))$ of $n = |\Omega_A|$ attribute values, $v_i \in dom(a_i)$, $a_i \in \Omega_A$.

For a common single-relational propositional language a subgroup description can then be defined as follows:

Definition 1 (Subgroup Description)**.** *A subgroup description $sd = e_1 \wedge e_2 \wedge \cdots \wedge e_l$ is defined by the conjunction of a set of l selectors $e_i = (a_i, V_i)$: Each selector denotes a selection expression on the domain of an attribute $a_i \in \Omega_A, V_i \subseteq dom(a_i)$. The function $sd(s)$ returns the subgroup description of the subgroup s. We define Ω_E as the (universal) set of selectors and Ω_{sd} as the set of all possible subgroup descriptions.*

A quality function measures the interestingness of the subgroup. Typical quality criteria include the difference in the distribution of the target variable concerning the subgroup and the general population, and the subgroup size.

Definition 2 (Quality Function)**.** *Given a particular target variable $t \in \Omega_E$, a quality function $q : \Omega_{sd} \times \Omega_E \rightarrow R$ is used in order to evaluate a subgroup description $sd \in \Omega_{sd}$, and to rank the discovered subgroups.*

Several quality functions were proposed (cf., [Wrobel, 1997; Klösgen, 1996; 2002; Lavrac *et al.*, 2004; Atzmueller and Puppe, 2006]), e.g., the functions q_{BT} and q_{RG}:

$$q_{BT} = \frac{(p - p_0) \cdot \sqrt{n}}{\sqrt{p_0 \cdot (1 - p_0)}} \cdot \sqrt{\frac{N}{N - n}}, \quad q_{RG} = \frac{p - p_0}{p_0 \cdot (1 - p_0)},$$

where p is the relative frequency of the target variable in the subgroup, p_0 is the relative frequency of the target variable in the total population, $N = |CB|$ is the size of the total population, and n denotes the size of the subgroup.

In contrast to the quality function q_{BT} (the classic binomial test), the quality function q_{RG} only compares the target shares of the subgroup and the total population measuring the *relative gain*. Therefore, a minimum support threshold $\mathcal{T}_{Supp}$ is necessary to discover significant subgroups, for which $n \geq \mathcal{T}_{Supp}$.

The result of subgroup discovery is a set of subgroups. Usually the best k subgroups, and/or the subgroups with a quality above a certain quality threshold are obtained and returned to the user. Since subgroup discovery methods are not necessarily covering algorithms the discovered subgroups can overlap significantly and their estimated quality (effect) might be confounded by external variables. Then, these need first to be identified before they can be adjusted for, in order to obtain an unbiased effect measure.

2.2 The Concept of Confounding

Confounding can be described as a bias in the estimation of the effect of the subgroup on the target concept due to attributes affecting the target concept that are not contained in the subgroup description [McNamee, 2003]. Thus, confounding is caused by a lack of comparability between subgroup and complementary group due to a difference in the distribution of the target concept caused by other factors.

An extreme case for confounding is presented by *Simpson's Paradox*: The (positive) effect (association) between a given variable X and a variable T is countered by a negative association given a third factor F, i.e., X and T are negatively correlated in the subpopulations defined by the values of F [Simpson, 1951]. For binary variables X, T, F this can be formulated as

$$\begin{aligned} P(T|X) &> P(T|\neg X)\,, \\ P(T|X,F) &< P(T|\neg X,F)\,, \\ P(T|X,\neg F) &< P(T|\neg X,\neg F)\,, \end{aligned}$$

i.e., the event X increases the probability of T in a given population while it decreases the probability of T in the subpopulations given by the restrictions on F and $\neg F$.

For the example shown in Figure 1, let us assume that there is a positive correlation between the event X that describes *people that do not consume soft drinks* and T specifying the diagnosis *diabetes*. This association implies that people not consuming soft drinks are affected more often by diabetes (50% non-soft-drinkers vs. 40% soft-drinkers). However, this is due to age, if older people (given by F) consume soft drinks less often than younger people, and if diabetes occurs more often for older people, inverting the effect.

Criteria for Confounders There are three criteria that must be satisfied for a potential **confounding factor** F [McNamee, 2003; Pearl, 2000], given the factors X contained in a subgroup description and a target concept T:

1. A confounding factor F must be a *cause* for the target concept T, e.g., an independent risk factor for a certain disease.
2. The factor F must be associated/correlated with the subgroup (factors) X.
3. A confounding factor F must *not* be affected by the subgroup (factors) X, i.e., if F is caused by X then F is not considered as a confounder.

Combined	T	¬T	∑	Rate (T)
X	25	25	50	50%
¬X	20	30	50	40%
∑	45	55	100	

Restricted on F	T	¬T	∑	Rate (T)
X	24	16	40	60%
¬X	8	2	10	80%
∑	32	18	50	

Restricted on ¬F	T	¬T	∑	Rate (T)
X	1	9	10	10%
¬X	12	28	40	30%
∑	13	37	50	

Figure 1: Example: Simpson's Paradox

However, these criteria are only necessary but not sufficient to identify confounders. If purely automatic methods are applied for detecting confounding, then such approaches may label some variables as confounders incorrectly, e.g., if the real confounders have not been measured, or if their contributions cancel out. Thus, user interaction is rather important for validating confounded relations. Furthermore, the identification of confounding requires causal knowledge since confounding is itself a causal concept [Pearl, 2000].

Another situation closely related to confounding is given by **effect modification**: Then, a third factor F does not necessarily need to be associated with the subgroup described by the factors X; F can be an additional factor that increases the effect of X in a certain subpopulation only, pointing to new subgroup descriptions that are interesting by themselves. Furthermore, both effect modification and confounding can also occur in combination.

2.3 Stratification for Handling Confounding

One method for controlling confounding factors is given by *stratification*, e.g., [McNamee, 2003]: Then, the relation that is suspected to be confounded is analyzed in different strata, i.e., restrictions to partitions of the population given by the individual values of the potential confounder. A necessary precondition for this method is that the confounding variable satisfies the criteria defined above in Section 2.2. For example, in the medical domain typical factors suspected of confounding are given by the attributes age, gender, or body-mass-index (BMI): Considering the potential confounder *age*, we could split on age groups such as $age < 50$, $age\ 50 - 70$, and $age > 70$ as the different strata. Then, the stratification method analyzes the subgroup on the different levels or partitions of the potential confounder, i.e., the subgroup – target relation is analyzed in the subpopulations given by $age < 50$, $age = 50 - 70$ and $age > 70$.

For analysis, the (crude) association strength, i.e., the subgroup quality considering the general population is compared to an adjusted quality given by a weighted sum of the associations restricted to the individual strata. If the association strength differs comparing the adjusted quality and the crude quality value, then this is an indication for confounding and/or effect modification. Otherwise, if the association persists across the different age groups/strata then the confounding association cannot be proved. However, if the adjusted quality (within the strata) differs significantly from the crude association in the whole population, then the factor under consideration is a candidate for confounding. Furthermore, if the strength of the association differs significantly between the different strata, then this is a sign for effect modification. To distinguish between these situations, interactive inspection by the user is often necessary, since confounding is a causal-concept dependent on causal domain knowledge.

3 Confounding-Aware Subgroup Discovery

The proposed semi-automatic approach for confounding-aware subgroup discovery applies an automatic method for proposing potentially confounded and/or effect-modified relations that can then be analyzed interactively.

In this section, we first describe the algorithm for confounding-aware subgroup discovery that applies an arbitrary subgroup discovery method both for unstratified and stratified subgroup discovery. The results of the discovery runs are then compared in order to identify subgroups that are potentially confounded, or that fit the characteristics of effect modification.

In the context of data mining, a similar approach was described in [Fabris and Freitas, 1999] with respect to the analysis of *Simpson's* paradox. However, the presented approach is more general since we do not only consider subgroups that fit into the Simpson's paradox pattern, but we consider the general situation of confounding, for which Simpson's paradox is only a special case. The runtime of the presented method depends significantly on the number of considered stratifying attributes and especially on the number of attribute values that are considered for stratification. Then, suitable value partitions can be provided manually, or they can be obtained using domain knowledge. Furthermore, we describe an interactive inspection technique for analyzing the interesting subgroups in detail, such that the subgroups suspected of either confounding or effect-modification can be intuitively analyzed.

3.1 Automatic Discovery and Analysis

The automatic approach for confounding-aware subgroup discovery consists of three basic steps:

1. We need to determine the set of potential confounders concerning the given target variable. We consider both known confounders using domain knowledge, and candidate attributes that are significantly dependent with the target variable.
2. A general subgroup discovery step is performed for identifying a set of candidate subgroups. Next, we apply stratification and perform subgroup discovery on the respective strata (or partitions) of the total population: In this way, we discover a set of interesting subgroups for each of the strata. We apply domain knowledge that specifies causal relations between domain objects in order to include all known confounders for the target variable as defaults. Furthermore, considering the criteria for confounding described in Section 2.2, we eliminate objects from the search space that (causally) affect the current confounder. The unstratified set of subgroups and all subgroups obtained from the different strata are then 'normalized', i.e., a subgroup is defined in each of these sets if it occurs initially in at least one of them.
3. Finally, we compare the qualities of the non-stratified and the adjusted qualities of the stratified subgroups for detecting confounding and effect modification.

Algorithm 1 Confounding-Aware Subgroup Discovery

```
Require: Target Variable $T$, set of selectors $E$ (search space), total population $P \subseteq CB$,
    quality function $q$, minimum quality threshold $\mathcal{T}_q$
 1: Select all attributes $C \subseteq \Omega_A$ that are significantly dependent with the target variable $T$
    according to a specified confidence level
 2: If domain knowledge is available: $C = C - \{a \,|\, a \in A_C$, a is not causal for $T\}$
 3: If domain knowledge is available: $C = C \cup \{a \,|\, a \in \Omega_A$, a is known confounder for $T\}$
 4: For the search space $E$, discover a set of subgroups $U = \{u \,|\, quality(u) \geq \mathcal{T}_q\}$
 5: $S_C = \emptyset$ {Potentially confounded subgroups}
 6: $S_E = \emptyset$ {Potentially effect-modified subgroups}
 7: $S_{CE} = \emptyset$ {Subgroups potentially affected by both confounding and effect-modification}
 8: $S_N = \emptyset$ {Subgroups not affected by confounding}
 9: for all $f \in C$ : do
10:   $E_f \subseteq E, E_f = \{e \,|\, e = (f, V_e)\}, m = |E_C|$
11:   for each $e_i \in E_f$ stratify $P$ into subpopulations $P_i = \{c \,|\, c \in P, e_i(c)\}, i = 1 \ldots m$.
12:   for all strata $P_i$ do
13:     $E' = \{e \,|\, e \in E$, f is not affected by e$\}$
14:     For the search space $E'$, discover a set of subgroups $S_i = \{s \,|\, quality(s) \geq \mathcal{T}_q\}$
        restricted to the respective stratum (subpopulation) $P_i$
15:   for all $i = 1 \ldots m$ do
16:     for all $j = 1 \ldots m, j \neq i$ do
17:       for all $s \in S_i - S_j$ do
18:         $s' = restrictedTo(P_i, s)$
19:         $S_j = S_j \cup \{s'\}$ {Make $S_i$ and $S_j$ compatible}
20:     for all $s \in S_i - U$ do
21:       $s' = restrictedTo(P, s)$
22:       $U = U \cup \{s'\}$ {Make $U$ compatible to all $S_i$}
23:   for all $i = 1 \ldots m$ do
24:     for all $u \in U - S_i$ do
25:       if $sd(u)$ is not affected by $f$ then
26:         $s' = restrictedTo(P_i, u)$
27:         $S_i = S_i \cup \{s'\}$ {Make all $S_i$ compatible to $U$}
28:   for all $u \in U$ do
29:     $S = \{s \,|\, s \in S_i, s = restrictedTo(P_i, u), i = 1 \ldots m\}$
30:     if $relAdjQualityDiff(u, S) > \mathcal{T}_C$ then
31:       $S_C = S_C \cup \{u\}$
32:     else
33:       $S_N = S_N \cup \{u\}$
34:   for all $i = 1 \ldots m$ do
35:     for all $j = 1 \ldots m, j \neq i$ do
36:       for all $s_i \in S_i$ do
37:         if $relQualityDiff(s_i, s_j) > \mathcal{T}_E$, where $s_j = restrictedTo(P_j, s_i)$ then
38:           $S_E = S_E \cup \{s_i\}$
39:   $S_{CE} = S_C \cap S_E, S_C = S_C - S_{CE}, S_E = S_E \cap S_N, S_N = S_N - S_E$
40: Evaluate the potentially confounded/effect-modified sets of subgroups $S_C$, $S_E$, $S_{EC}$
```

The approach is shown in Algorithm 1 and explained in more detail below: It requires a target variable T, the search space given by a set of selectors E, the total population P, and an arbitrary quality function q with a minimum quality threshold $\mathcal{T}_q$. For certain quality functions we require an additional minimal support threshold $\mathcal{T}_{Supp}$, e.g., for the relative gain quality function (cf., Section 2.1).

Identifying a Set of Potential Confounders (lines 1-3) In a first step, we select all attributes that are significantly dependent with the target variable T considering a high user-defined confidence level, e.g., using the standard χ-square test for independence. We retrieve this set of attributes based upon the following observations: It is easy to see that according to the criteria for confounding discussed in Section 2.2, the set of confounders potentially includes all factors, that are significantly associated/dependent with the target variable, and also known confounders (risk factors), and/or combinations of these variables. Known confounders are added, because their significance level might be too low to be included in the significantly dependent confounders. Using causal domain knowledge, we can filter the statistically proposed candidates for confounding, and remove those that are not causal for the target variable.

Subgroup Discovery and Stratification (lines 4-26) For analysis, we consider four sets of subgroups: The set S_C including potentially confounded subgroups, the set S_E containing subgroups with potential effect-modification, the set S_{CE} concerning both confounding and effect-modification, and finally the set S_N containing (non-confounded) subgroups.

Initially, a general subgroup discovery step is performed in order to identify a set of candidate subgroups U that can include non-confounded and confounded subgroups. In the stratification-loop, we can first remove all selectors from the search space, that affect the current confounder, according to the criteria for detecting confounding. This is an optional step requiring causal domain knowledge; if none is available, then we start with the full set of selectors. In this case, the interactive evaluation and validation step performed by the user becomes more important.

For each stratum of the current confounder under consideration, a set of subgroups with a quality above a minimum quality threshold is obtained. If the quality function requires a minimum support threshold, then we apply this threshold relative to the respective subpopulation size when performing stratified subgroup discovery. After subgroup discovery, we consider all pairs of strata subgroups and normalize the respective sets of subgroups such that the sets are 'compatible' and contain a subgroup (restricted to the respective stratum), if any other set contains the respective subgroup. This is performed analogously for the set U and for each stratified set of subgroups.

Detecting Confounding and Effect-Modification (lines 27-38) After all sets of subgroups are compatible, we perform the tests for detecting confounding and effect-modification: If the *relative adjusted quality difference* ($relAdjQualityDiff$) value comparing the (crude) subgroup quality and the adjusted quality value of the strata subgroups is larger than a certain threshold $\mathcal{T}_C$, e.g., $\mathcal{T}_C = 0.2$ then we conclude potential confounding, and otherwise no confounding. The adjusted quality of a set of stratified subgroups S is computed as follows:

$$adjQuality(S) = \sum_{s \in S} \frac{|P_s|}{|P|} quality(s)\,,$$

where P_s specifies the respective stratum (subpopulation) of the subgroup s, and P specifies the total population. The relative adjusted difference between the (crude) quality of a subgroup u and its adjusted quality considering the corresponding stratified subgroups S, is given by:

$$relAdjQualityDiff(u, S) = \frac{\left|adjQuality(S) - quality(u)\right|}{quality(u)}$$

If the *relative quality difference* ($relQualityDiff$) within a pair of strata differs significantly according to the threshold $\mathcal{T}_E$, e.g., $\mathcal{T}_E = 0.2$, then we can infer effect modification. The relative difference between the strata subgroup s_i and another strata subgroup s_j, is then computed as:

$$relQualityDiff(s_i, s_j) = \frac{\left|quality(s_i) - quality(s_j)\right|}{\max(quality(s_i), quality(s_j))}$$

After the respective sets considering confounding/no effect-modification (S_C), effect-modification/no confounding (S_E), and confounding/effect-modification (S_{CE}) have been generated, they need to be evaluated and validated by the user using the interactive stratification method, as discussed below in Section 3.3. After a confounder has been validated, the domain knowledge can also be extended incrementally.

The proposed method described in Algorithm 1 can also be iterated for combinations of confounders and their respective value domains. However, the contribution and impact of combinations of confounding variables (considering very many factors) is often hard to estimate and to interpret by the users. In the medical domain, for example, the analysis is often restricted to combinations of typical confounders such as age, sex, or body-mass-index (BMI). Additionally, small case numbers observed in the resulting crosstables can then become a (statistical) problem.

3.2 Determining Suitable Strata for Stratification

In general, the computational complexity of the algorithm significantly depends on the number of strata (value partitions) for each considered confounder. Therefore, decreasing the number of strata can significantly increase the efficiency of the algorithm.

Furthermore, reducing the considered strata also helps to reduce a common problem for stratification, i.e., small case numbers for the subgroups contained in the different strata. In an incremental process, we first start with a reduced number of partitions for a coarse analysis, and then refine these in a detailed analysis. As a general approach we can apply common discretization methods in order to shrink the intervals of certain numeric or ordinal attributes, e.g., based on the chi-merge algorithm [Kerber, 1992]. Otherwise, if there are already defined partitions, then we apply these during stratification.

For ordinal attributes, i.e., for attributes with an ordered value domain, we can also apply domain knowledge: *Abnormality/Normality information* can then be applied for determining appropriate splits on the domain of an attribute, if available. Abnormality/Normality information is common for diagnostic domains, e.g., in the medical domain the sets of 'normal' and 'abnormal' attribute values correspond to the expected and unexpected/pathological values, respectively. Each attribute value is attached with a label specifying a normal or an abnormal state. Normality information only requires a binary label. Abnormality information defines several categories, e.g., consider the value range $\{\text{normal}, \text{marginal}, \text{high}, \text{very high}\}$ of the attribute *temperature*. The values *normal* and *marginal* denote normal states of the attribute, in contrast to the values *high* and *very high* describing abnormal states.

If normality information for ordinal attributes is available, then we split by the normal value, and obtain a range below the normal, the normal, and a range above the normal value, resulting in three different partitions for stratification. Using abnormality information we can obtain further partitions considering the 'lower' and the 'upper' partition, by grouping similar adjacent abnormality categories such that the new ranges have at least a minimum size that needs to be specified by the user.

3.3 Interactive Evaluation and Analysis

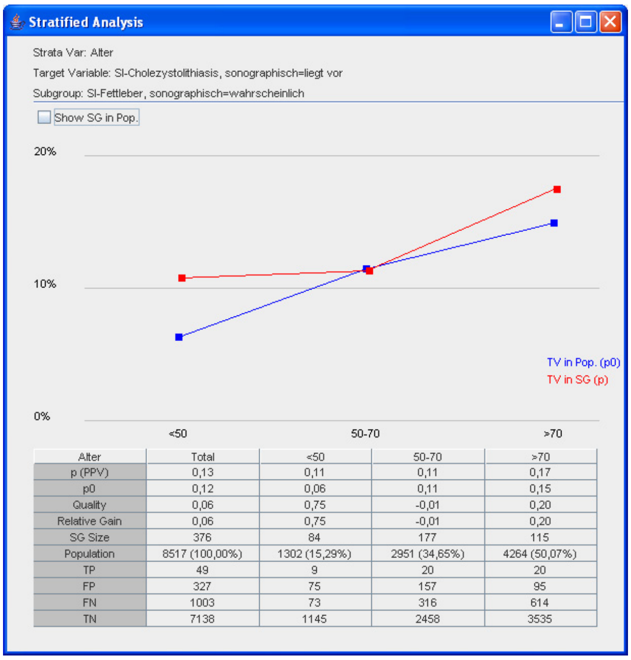
If purely automatic statistical methods are applied for detecting confounding, then such approaches may label some variables as confounders incorrectly [Pearl, 2000]: It is easy to see that confounding variables can potentially blur an association, but cannot be identified if the confounding variables themselves have not been measured. Furthermore, the contributions of different confounders can even cancel out their respective effects such that they cannot be identified using the statistical criteria [McNamee, 2003]. Thus, user interaction is rather important when testing confounding factors, and when evaluating the individual contributions of potential confounding variables.

We favor interactive approaches with short feedback-cycles for testing confounding factors: In a semi-automatic approach the presented automatic algorithm can be used to discover potential confounders that are then presented to the user for subsequent evaluation and validation.

In this semi-automatic process suitable visualization methods are essential: For potentially confounded subgroups the user can then perform the stratification step interactively, supported e.g., by line charts that show the distribution of the target variable within the respective strata and within the subgroups contained in these.

Examples are shown in Figure 2 and Figure 3, respectively. The top of the figures shows a graph displaying the positive predictive value (p), i.e., the target share or the precision, of the target variable of the subgroup (in red color) and the target share considering the respective population (in blue color) in the different strata determined by the stratification variable.

A detailed view of the stratification is given by the numbers contained in the table at the bottom of the figures. There, the stratification parameters are given that also contain the positive predictive value, the quality of the subgroup, and the true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) of the subgroups. The stratification plot gives an impression of the relations restricted to the individual strata and can be used for further analyzing confounding or effect modification. Then, a detailed analysis can be performed by the user considering the statistical parameters.



Strata Var: Alter

Target Variable: SI-Cholezystolithiasis, sonographisch=liegt vor

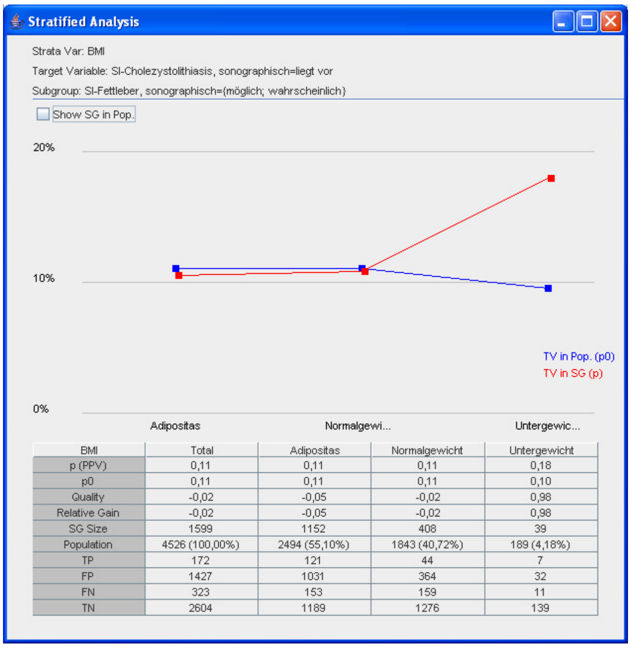
Subgroup: SI-Fettleber, sonographisch=wahrscheinlich

Alter	Total	<50	50-70	>70
p (PPV)	0,13	0,11	0,11	0,17
p0	0,12	0,06	0,11	0,15
Quality	0,06	0,75	-0,01	0,20
Relative Gain	0,06	0,75	-0,01	0,20
SG Size	376	84	177	115
Population	8517 (100,00%)	1302 (15,29%)	2951 (34,65%)	4264 (50,07%)
TP	49	9	20	20
FP	327	75	157	95
FN	1003	73	316	614
TN	7138	1145	2458	3535

Figure 2: Detailed stratification view on the subgroup *Fatty liver=probable* for the target variable *Gallstones* stratified by the attribute Age for the strata: $< 50, 50 - 70, > 70$.

4 Case Study – Examples

We use cases taken from the SonoConsult system [Huettig *et al.*, 2004] – a medical documentation and consultation system for sonography. The system is in routine use in the DRK-hospital in Berlin/Köpenick and documents an average of about 350 cases per month. These are detailed descriptions of findings of the examination(s), together with the inferred diagnoses (binary attributes). The derived diagnoses are usually correct as shown in a medical evaluation (c.f. [Huettig *et al.*, 2004]), resulting in a high-quality case base with detailed case descriptions.



Strata Var: BMI

Target Variable: SI-Cholezystolithiasis, sonographisch=liegt vor

Subgroup: SI-Fettleber, sonographisch=(möglich; wahrscheinlich)

BMI	Total	Adipositas	Normalgewicht	Untergewicht
p (PPV)	0,11	0,11	0,11	0,18
p0	0,11	0,11	0,11	0,10
Quality	-0,02	-0,05	-0,02	0,98
Relative Gain	-0,02	-0,05	-0,02	0,98
SG Size	1599	1152	408	39
Population	4526 (100,00%)	2494 (55,10%)	1843 (40,72%)	189 (4,18%)
TP	172	121	44	7
FP	1427	1031	364	32
FN	323	153	159	11
TN	2604	1189	1276	139

Figure 3: Detailed stratification view on the subgroup *Fatty liver=probable* for the target variable *Gallstones* stratified by the attribute Body-Mass-Index (BMI) for the strata: Adipositas, Normal weight, Underweight

The domain ontology contains about 400 basic attributes with about 5 nominal attribute values on average, about 70 clinicaly relevant diagnoses, which are inferred by rules and additional intermediate diagnostic concepts (i.e., finding abstractions and course diagnoses).

The experiments were performed using the VIKAMINE system [Atzmueller and Puppe, 2005] implementing the presented approach; we applied part of the SonoConsult case base containing about 8600 cases. For subgroup discovery the fast and effective *SD-Map* algorithm [Atzmueller and Puppe, 2006] was applied; we utilized the relative gain quality function (cf., Section 2.1) for estimating the quality of the subgroups.

In the following, we provide some examples shown in Table 4 for the exemplary target variable (diagnosis) *Gallstones=established*, stratified by the attribute *Sex*: While these examples are quite simple to interpret and to understand, they are nevertheless relevant for the clinical context.

Subgroup #1 is an example for confounding, but no effect modification, since the subgroup qualities within the strata are almost equal. The subgroups #2 and #3 are examples for both confounding *and* effect modification: There is a significant difference between the crude and the adjusted quality, and the selector *sex=female* significantly increases the risk for gallstones in the respective subgroups.

Next, subgroup #4 is an example for no confounding, but effect modification: The adjusted quality does not differ from the crude (unstratified) quality, but *sex=male* is much more indicative for the target variable. Finally, subgroup #5 is a subgroup for a borderline case that is affected by effect modification, but is just on the threshold-given 'border' for confounding: Since we used a threshold $\mathcal{T}_C = 0.2$ the subgroup was not marked as being potentially confounded; nevertheless, we can observe effect-modification considering the different strata.

		Quality			
No.	*Subgroup Description*	*Crude*	*Adjusted*	*Sex=male*	*Sex=female*
1	Cirrhosis of the liver=probable	0.15	0.19	0.2	0.19
2	Chronic Renal Failure=established	0.32	0.15	0.03	0.26
3	Fatty liver = possible AND Aorta sclerosis=calcifying	0.64	0.49	0.1	0.85
4	Age$\geq$ 70	0.24	0.23	0.28	0.18
5	Aorta sclerosis=calcifying	0.3	0.24	0.1	0.37

Table 1: Exemplary subgroups for the target variable *Gallstones=established* stratified by the values of the attribute *Sex*

5 Conclusion

In this paper, we have presented a semi-automatic approach for confounding-aware subgroup discovery. We favor an interactive approach since there is no purely automatic test for confounding [Pearl, 2000]. We presented an automatic algorithm providing the means for detecting both potentially confounded and/or effect-modified subgroup patterns and other unconfounded relations: The algorithm integrates a standard subgroup discovery method and extends it for the analysis of confounding and effect-modification. Additionally, we discussed interactive methods for the analysis and validation of the relations proposed by the algorithm that can easily be applied by the user. Furthermore, we have shown how to utilize (causal) domain knowledge for improving the results of the algorithm, since confounding is itself a causal concept. The applicability and benefit of the presented approach were illustrated by examples from a case-study using data from a fielded system in the medical domain of sonography.

In the future, we consider to include further domain knowledge for causal analysis. Furthermore, the integration of semi-automatic methods for extended causal analysis of subgroup patterns is also a promising direction for further improvements. Another promising option for future work is given by specialized interestingness measures for the evaluation of the confounded relations. Additionally, we are planning to apply generated data containing known (confounded) relations for estimating the efficiency and the effectiveness of the presented approach.

Acknowledgements

The authors want to thank Prof. Dr. Hans-Peter Buscher from the department of internal medicine at the DRK-Klinik Berlin-Köpenick, and PD Dr. Hardi Lührs from the department of sonography at the Würzburg University Hospital for their medical expertise and analysis while performing the case studies of this research project.

This work has been partially supported by the German Research Council (DFG) under grant Pu 129/8-1.

References

[Atzmueller and Puppe, 2005] Martin Atzmueller and Frank Puppe. Semi-Automatic Visual Subgroup Mining using VIKAMINE. *Journal of Universal Computer Science*, 11(11):1752–1765, 2005.

[Atzmueller and Puppe, 2006] Martin Atzmueller and Frank Puppe. SD-Map - A Fast Algorithm for Exhaustive Subgroup Discovery. In *Proc. 10th European Conf. on Principles and Practice of Knowledge Discovery in Databases*, pages 6–17, Berlin, 2006. Springer.

[Atzmueller *et al.*, 2005] Martin Atzmueller, Frank Puppe, and Hans-Peter Buscher. Exploiting Background Knowledge for Knowledge-Intensive Subgroup Discovery. In *Proc. 19th Intl. Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 647–652, Edinburgh, Scotland, 2005.

[Cooper, 1997] Gregory F. Cooper. A Simple Constraint-Based Algorithm for Efficiently Mining Observational Databases for Causal Relationships. *Data Mining and Knowledge Discovery*, 1(2):203–224, 1997.

[Fabris and Freitas, 1999] Carem C. Fabris and Alex A. Freitas. Discovering Surprising Patterns by Detecting Occurrences of Simpson's Paradox. In *Research and Development in Intelligent Systems XVI*, pages 148–160, Berlin, 1999. Springer.

[Huettig *et al.*, 2004] Matthias Huettig, Georg Buscher, Thomas Menzel, Wolfgang Scheppach, Frank Puppe, and Hans-Peter Buscher. A Diagnostic Expert System for Structured Reports, Quality Assessment, and Training of Residents in Sonography. *Med. Klinik*, 99(3):117–122, 2004.

[Kerber, 1992] Randy Kerber. ChiMerge: Discretization of Numeric Attributes. In *Proc. 10th National Conference on Artificial Intelligence*, pages 123–128, San Jose, California, USA, 1992. AAAI.

[Klösgen, 1996] Willi Klösgen. Explora: A Multipattern and Multistrategy Discovery Assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI Press, 1996.

[Klösgen, 2002] Willi Klösgen. *Handbook of Data Mining and Knowledge Discovery*, chapter 16.3: Subgroup Discovery. Oxford University Press, New York, 2002.

[Lavrac *et al.*, 2004] Nada Lavrac, Branko Kavsek, Peter Flach, and Ljupco Todorovski. Subgroup Discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.

[McNamee, 2003] Roseanne McNamee. Confounding and Confounders. *Occup. Environ. Med.*, 60:227–234, 2003.

[Pearl, 2000] Judea Pearl. *Causality: Models, Reasoning and Inference*, chapter 6.2 Why There is No Statistical Test For Confounding, Why Many Think There Is, and Why They Are Almost Right. Cambridge University Press, 2000.

[Simpson, 1951] Edward H. Simpson. The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society*, 18:238–241, 1951.

[Wrobel, 1997] Stefan Wrobel. An Algorithm for Multi-Relational Discovery of Subgroups. In *Proc. 1st European Symp. on Principles of Data Mining and Knowledge Discovery*, pages 78–87, Berlin, 1997. Springer.

Regularization through Multi-Objective Optimization

Ingo Mierswa
Artificial Intelligence Unit
Department of Computer Science
University of Dortmund
ingo.mierswa@uni-dortmund.de

Abstract

It has been shown that optimization schemes based on evolutionary algorithm can be successfully integrated into statistical learning methods. A Support Vector Machine (SVM) using evolution strategies for its optimization problem frequently deliver better results with respect to the optimization criterion and the prediction accuracy. Moreover, evolutionary computation allows for the efficient large margin optimization of a huge family of new kernel functions, namely non-positive semidefinite kernels as the Epanechnikov kernel. For these kernel functions, evolutionary SVM even outperform other learning methods like the Relevance Vector Machine. In this paper, we will discuss another major advantage of evolutionary large margin methods compared to traditional solutions: we can explicitly optimize the inherent trade-off between training error and model complexity by embedding multi-objective optimization into the learning method. This leads to three advantages: first, it is no longer necessary to tune the regularization parameter which weighs both conflicting criteria. This is a very time-consuming task for traditional large margin methods. Second, the shape and size of the Pareto front give interesting insights about the complexity of the learning task at hand. Finally, the user can actually see the point where overfitting occurs and can easily select a solution from the Pareto front best suiting his or her needs.

1 Introduction

Recently, several approaches were proposed where evolutionary algorithms are used to solve large margin optimization problems [Mierswa, 2006a; Jun and Oh, 2006]. Although the latter only performed evolutionary optimization on the less efficient primary optimization problem of a Support Vector Machine (SVM), both publications demonstrate the interest in this new intersection of three highly active research areas, namely machine learning, statistical learning theory, and evolutionary algorithms.

Usually, the optimization problems posed by large margin methods is solved with quadratic programming. However, there are some drawbacks with these approaches. First, no unique global optimum exists for kernel functions which are not positive semidefinite. Non-positive semidefinite kernel functions are functions which resemble a (partial) distance instead of a similarity measure. In these cases, quadratic programming is not able to find satisfying solutions at all. Moreover, most implementations do not even terminate [Haasdonk, 2005]. There exist several useful non-positive kernels [Lin and Lin, 2003], among them the sigmoid kernel which simulates a neural network [Camps-Valls *et al.*, 2004; Smola *et al.*, 2000]. Therefore, a more generic optimization scheme based on evolutionary strategies was recently proposed which allows such non-positive kernels without the need for omitting the more efficient dual optimization problem [Mierswa, 2006b]. It has been shown that the evolutionary implementation of a Support Vector Machine leads to as good results as traditional SVM on a broad variety of real-world benchmark data sets. For non-positive semidefinite kernel functions it always outperform traditional SVM and other related learning methods as the Relevance Vector Machine.

Former applications of evolutionary algorithms to SVM include the optimization of method and kernel parameters [Friedrichs and Igel, 2004; Runarsson and Sigurdsson, 2004], the selection of optimal feature subsets [Frp̈hlich *et al.*, 2004], and the creation of new kernel functions by means of genetic programming [Howley and Madden, 2005]. The latter is particularly interesting since it cannot be guaranteed that the resulting kernel functions are again positive semidefinite. In contrast to these approaches, we embed evolutionary algorithms into the learning machine itself and solve the optimization problem of large margin methods like SVM in its dual form. By doing this, we can avoid another drawback connected to traditional SVM learning. Although the statistical learning theory takes into account both the training error and the model complexity, the user still has to define a weighting factor for both conflicting criteria. The search for this parameter is usually a non-trivial and very time consuming task.

In this paper, we propose to embed multi-objective evolutionary algorithms into SVM and Kernel Logistic Regression (KLR). This allows, for a first time, to explicitely optimize the inherent trade-off which is the basic idea of statistical learning theory without applying time-consuming outer wrapper and validation approaches for optimizing the trade-off. This goal differs from the first attempts to incorporate multivariate performance measures into SVM [Joachims, 2005] which cannot be used for competing criteria and does not solve the general trade-off between training error and capacity.

The result of the proposed approach is a Pareto front in the space of training error vs. model complexity and gives interesting insights into the nature of the problem at hand. By using a hold-out data set as a test set for the resulting models we derive a second front showing the generalization error. Both, the Pareto front and the generalization error plot allows for a quick selection of the final solution from the Pareto front without the time-consuming optimization of a weighting factor.

1.1 Outline

In Section 2 we give a short introduction into the concept of regularized risk minimization and the ideas of large margin learning methods. This allows us to formalize the optimization problem of SVM and KLR for the classification of given data points. The constrained optimization problems discussed in this section will be divided into two sub problems which will be transformed into their dual forms in Section 3. Both objectives will be used in a multi-objective evolution strategies algorithm which solves the SVM and the KLR problem respectively while the trade-off between training error and model complexity is explicitely kept in the resulting Pareto fronts (see Section 4). Finally, we give some examples of results on synthetical and real-world benchmark data sets in Section 5 before we conclude this paper in Section 6.

2 Large Margin Methods

Let the instance space be defined as the Cartesian product $X = X_1 \times \ldots \times X_m$ of attributes $X_i \subseteq \mathbb{R}$. Let Y be another set of possible labels. X and Y are random variables obeying a fixed but unknown probability distribution $P(X, Y)$. *Supervised Machine Learning* tries to find a function $f(x, \gamma)$ which predict the value of Y for a given input $x \in X$. The function class f depends on a vector of parameters γ, e. g. if f is the class of all polynomials, γ might be the degree. We define a *loss function* $L(Y, f(X, \gamma))$ in order to penalize errors during prediction [Hastie *et al.*, 2001]. Every convex function with arity 2, positive range, and $L(y, y) = 0$ can be used as loss function [Smola *et al.*, 1998]. Since the mere minimization of the loss according to the training data is not appropriate to find a good generalization, we incorporate the *capacity* [Vapnik, 1998] of the used function into the optimization problem leading to the *regularized risk*:

Definition 1 *Let Ω be strictly monotonic increasing function. The* REGULARIZED RISK *is defined as*

$$R_{reg}(\gamma) = R_{emp}(\gamma) + \lambda\Omega(\gamma).$$

This risk functional is also known as *structural risk* since it takes the structural complexity into account. Ω is a function which measures the capacity of the function class f depending on the parameter vector γ (see [Schölkopf and Smola, 2002] for more details). Since the empirical risk is usually a monotonically decreasing function of Ω, both criteria are conflicting and we use λ to manage the trade-off between training error and capacity.

We need to use a class of approximation functions whose capacity can be controlled. In this section, we will discuss a special form of regularized risk minimization, namely a large margin approach. All large margin methods have one thing in common: they embed regularized risk minimization by maximizing a margin between a linear function and the nearest data points. We will discuss two of the most prominent large margin methods for classification tasks: the *Support Vector Machine* (SVM) and the *Kernel Logistic Regression* (KLR). Both can be expressed in the same way:

Problem 1 *Large margin problems are defined as*

$$\textit{minimize } \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} L\left(y_i, f(x_i)\right). \tag{1}$$

The vector w is the normal vector of the separating hyperplane. Minimizing $||w||^2$ corresponds to maximizing the margin. The second term calculates the training error and C corresponds to the trade-off parameter λ. Due to the representer theorem [Kimeldorf and Wahba, 1971], all optimal solutions of large margin methods have the form

$$f(x) = b + \sum_{i=1}^{n} \alpha_i K(x, x_i).$$

This is finite expansion in the representers $K(x, x_i)$ where K is a kernel function and the x_i are the training examples.

2.1 Support Vector Machines

Without loss of generality, we constrain the classes to $Y = \{-1, +1\}$. Furthermore, we use the so called hinge loss as a loss function:

$$L(Y, f(X)) = (1 - Yf(X))_+ .$$

The problem 1 can then be transformed into its dual form (see [Vert *et al.*, 2004] for details):

Problem 2 *The SVM problem is defined as:*

$$\begin{aligned}&\textit{maximize } \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k\left(x_i, x_j\right)\\ &\textit{subject to } 0 \le \alpha_i \le C \textit{ for all } i = 1, \ldots, n\\ &\quad \textit{and } \sum_{i=1}^{n}\alpha_i y_i = 0.\end{aligned}$$

Please note that the variables α_i are constrained by the upper bound C, i.e. by the user defined trade-off factor and that the examples x_i only occur in scalar products which was replaced by a kernel function $k = \langle\Phi\left(x_i\right), \Phi\left(x_j\right)\rangle$ for a (non-linear) mapping Φ in an arbitrary dot product space. This allows the search for a linear separating hyperplane in high-dimensional spaces after a non-linear transformation and, hence, the separating of non-linearly separable data.

2.2 Kernel Logistic Regression

Without loss of generality, we constrain the classes to $Y = \{0, 1\}$. Furthermore, we use the negative log-likelihood as a loss function:

$$L(Y, f(X)) = \log\left(1 + e^{-Yf(X)}\right)$$

The problem 1 can then be transformed into its dual form (see [Keerthi *et al.*, 2005] for details):

Problem 3 *The KLR problem is defined as:*

$$\begin{aligned}&\textit{maximize } \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k\left(x_i, x_j\right) +\\ &\qquad C\sum_{i=1}^{n} L(Y, f(X))\\ &\textit{subject to } \alpha_i \ge 0 \textit{ for all } i = 1, \ldots, n\\ &\quad \textit{and } \sum_{i=1}^{n}\alpha_i y_i = 0.\end{aligned}$$

Please note that the examples x_i again only occur in scalar products which was replaced by a kernel function $k = \langle\Phi\left(x_i\right), \Phi\left(x_j\right)\rangle$.

2.3 Comparison between SVM and KLR

Since both large margin learning methods stated above are very similar and differ basically only in the choice of the loss function we will compare both methods in this paragraph:

- the classification performance of both methods is very similar
- the hinge loss does not take the correctly classified examples into account, the logistic loss cares about all examples
- for the KLR method the probability $P(y = 1|x)$ can directly be computed as $P(y = 1|x) = e^{f(x)}/(1 + e^{f(x)})$ whereas this is not possible for SVM
- the logistic regression generalizes naturally to M-class classification through

  $$P(y = j|x) = \frac{e^{f_j(x)}}{e^{f_1(x)} + \ldots + e^{f_M(x)}}$$

  with $\sum f_m(x) = 0$
- the final example weights a_i are often zero for SVM (sparse solutions) but are often non-zero for KLR
- the kernel logistic regression has a runtime of $O(n^3)$ versus $O(n^2 N_{SV})$ for SVM if N_{SV} is the final number of support vectors

3 Explicit Trade-off between Error and Complexity

Since traditional large margin methods are, for example, not able to optimize for non-positive semidefinite kernel functions and approaches like Relevance Vector Machines are hardly feasible for real-world problems, it is a very appealing idea to replace the usual quadratic programming approaches by an *evolution strategies* (ES) optimization [Beyer and Schwefel, 2002] or by *particle swarm optimization* (PSO) [Kennedy and Eberhart, 1995]. Embedding evolutionary computation into large margin methods has the additional advantage of a straightforward application of multi-objective selection schemes in order to simultaneously optimize several conflicting criteria. In this work, we divide the criteria of the objective function stated above into two optimization targets while the weighting factor C can be omitted. This leads to the following two optimization problems:

$$\begin{aligned} &\text{minimize } \frac{1}{2}||w||^2 && (2)\\ &\text{subject to } \forall i : y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i\\ &\text{and } \forall i : \xi_i \geq 0 \end{aligned}$$

and

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{n} \xi_i && (3)\\ &\text{subject to } \forall i : y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i\\ &\text{and } \forall i : \xi_i \geq 0. \end{aligned}$$

where $g(\xi_i)$ is the loss function for an error ξ_i (slack variables). We will transform both objectives into their dual form in order to allow the efficient optimization of the problems including the usage of kernel functions.

3.1 First Objective: Maximizing the Margin

We introduce positive Lagrange multipliers into equation 2 but need multipliers α for the first set of inequality constraints and multipliers β for the second set of inequality constraints:

$$\begin{aligned} L_p^{(1)} =& \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i \left(y_i\left(\langle w, x_i\rangle + b\right) + \xi_i - 1\right) - \\ & \sum_{i=1}^{n} \beta_i \xi_i \end{aligned}$$

In order to find a solution we have to find the minimum by setting the derivatives to 0;

$$\begin{aligned} \frac{\partial L_p^{(1)}}{\partial w}(w, b, \xi, \alpha, \beta) &= w - \sum_{i=1}^{n} y_i \alpha_i x_i = 0,\\ \frac{\partial L_p^{(1)}}{\partial b}(w, b, \xi, \alpha, \beta) &= \sum_{i=1}^{n} \alpha_i y_i = 0,\\ \frac{\partial L_p^{(1)}}{\partial \xi_i}(w, b, \xi, \alpha, \beta) &= -\alpha_i - \beta_i = 0. \end{aligned}$$

Plugging the derivatives into the primal objective function $L_p^{(1)}$ delivers

$$\begin{aligned} L_p^{(1)} &= \frac{1}{2}||w||^2 - \sum_{i=1}^{n} -\alpha_i y_i \left\langle \sum_{j=1}^{n} \alpha_j y_j x_j, x_i \right\rangle + \sum_{i=1}^{n} \alpha_i\\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle \end{aligned}$$

The Wolfe dual must be maximized which leads to the formalization of the first objective of the multi-objective large margin method setting. The resulting problem is very similar to the usual dual SVM problem but without the upper bound C for the α_i (again, the dot product is replaced by a kernel function k):

Problem 4 *The first objective (maximize margin) is defined as:*

$$\begin{aligned} &\textit{maximize } \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k\left(x_i, x_j\right)\\ &\textit{subject to } \alpha_i \geq 0 \textit{ for all } i = 1, \ldots, n\\ &\textit{and } \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned}$$

3.2 Second Objective: Minimizing the Number of Training Errors

For the second objective, we discuss the solutions for both learning methods SVM and KLR on their own.

Second Objective for SVM

The second problem states that the sum of errors, i.e. the sum of the slack variables ξ_i, should be minimized. This optimization must be performed under the same inequality constraints as for the first objective. We add positive Lagrange multipliers α and β:

$$L_p^{(2)} = \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left(y_i\left(\langle w, x_i\rangle + b\right) + \xi_i - 1\right) - \sum_{i=1}^{n} \beta_i \xi_i$$

The derivatives must again be set to 0 which leads to slightly different conditions on the derivatives of $L_p^{(2)}$:

$$
\begin{aligned}
\frac{\partial L_p^{(2)}}{\partial w}(w,b,\xi,\alpha,\beta) &= -\sum_{i=1}^{n} y_i \alpha_i x_i = 0,\\
\frac{\partial L_p^{(2)}}{\partial b}(w,b,\xi,\alpha,\beta) &= \sum_{i=1}^{n} \alpha_i y_i = 0,\\
\frac{\partial L_p^{(2)}}{\partial \xi_i}(w,b,\xi,\alpha,\beta) &= 1 - \alpha_i - \beta_i = 0.
\end{aligned}
$$

Plugging the derivatives into the $L_p^{(2)}$ cancels out most terms because of the first two derivatives:

$$L_p^{(2)} = \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \xi_i + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \beta_i \xi_i$$

Together with the third derivative we can replace the β_i by $1-\alpha_i$ leading to

$$
\begin{aligned}
L_p^{(2)} &= \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} \alpha_i \xi_i + \sum_{i=1}^{n} \alpha_i\\
&= \sum_{i=1}^{n} \alpha_i
\end{aligned}
$$

The Wolfe dual must again be maximized which leads to the second objective of the multi-objective SVM setting. Maximizing the sum of α_i corresponds to transforming each example into a support vector. In the limit, this means that the training set is merely memorized instead of generalized which is an indication of overfitting or training error minimization respectively.

Problem 5 *The second objective (minimize error) for multi-objective SVM is defined as:*

$$
\begin{aligned}
&\textit{maximize } \sum_{i=1}^{n} \alpha_i\\
&\textit{subject to } \alpha_i \geq 0 \textit{ for all } i = 1,\ldots,n\\
&\textit{and } \sum_{i=1}^{n} \alpha_i y_i = 0.
\end{aligned}
$$

Second Objective for KLR

The second problem states that the sum of errors, i.e. the sum of the slack variables ξ_i, should be minimized. In the case of the Kernel Logistic Regression, this can be stated as

$$
\begin{aligned}
&\text{minimize } \sum_{i=1}^{n} g(\xi_i) \qquad (4)\\
&\text{subject to } \forall i : \xi_i = -y_i f(x_i)
\end{aligned}
$$

for the loss function $g(\xi_i) = \big(1 + e^{\xi_i}\big)$. We add positive Lagrange multipliers α in order to embed the constraints into the objective function:

$$L_p^{(2)} = \sum_{i=1}^{n} g(\xi_i) - \sum_{i=1}^{n} \alpha_i \left(\xi_i - y_i \left(\langle w, x_i\rangle + b\right)\right)$$

The derivatives must again be set to 0 which leads to the conditions:

$$
\begin{aligned}
\frac{\partial L_p^{(2)}}{\partial w}(w,b,\xi,\alpha) &= -\sum_{i=1}^{n} y_i \alpha_i x_i = 0,\\
\frac{\partial L_p^{(2)}}{\partial b}(w,b,\xi,\alpha) &= \sum_{i=1}^{n} \alpha_i y_i = 0,\\
\frac{\partial L_p^{(2)}}{\partial \xi_i}(w,b,\xi,\alpha) &= g'(\xi_i) - \alpha_i = 0 \quad \text{for all i.}
\end{aligned}
$$

From these conditions we can follow that

$$\xi_i(\alpha_i) = g'^{-1}(\alpha_i).$$

We define the function

$$G(\alpha_i) = g(\xi_i) - \alpha_i \xi_i$$

and calculate the derivative

$$
\begin{aligned}
G'(\alpha_i) &= \frac{\partial G(\alpha_i)}{\partial \alpha_i}\\
&= \frac{\partial g(\xi_i)}{\partial \alpha_i} \cdot \frac{\partial \xi_i(\alpha_i)}{\partial \alpha_i} - \alpha_i \cdot \frac{\partial \xi_i(\alpha_i)}{\partial \alpha_i} + \xi_i\\
&= (\alpha_i - g'(\xi_i)) \cdot \frac{\partial \xi_i(\alpha_i)}{\partial \alpha_i} + \xi_i(\alpha_i)\\
&= \xi_(\alpha_i)\\
&= g'^{-1}(\alpha_i)
\end{aligned}
$$

Plugging the derivatives into the $L_p^{(2)}$ again cancels out most terms:

$$
\begin{aligned}
L_p^{(2)} &= \sum_{i=1}^{n} g(\xi_i) - \alpha_i \xi_i\\
&= \sum_{i=1}^{n} G(\alpha_i)
\end{aligned}
$$

For the logistic loss function g the function G can be calculated as [Keerthi *et al.*, 2005]:

$$G(\alpha_i) = \alpha_i \log(\alpha_i) + (1-\alpha_i)\log(1-\alpha_i).$$

This Wolfe dual must again be maximized which leads to the second objective of the multi-objective KLR setting:

Problem 6 *The second objective (minimize error) for multi-objective KLR is defined as:*

$$
\begin{aligned}
&\textit{maximize } \sum_{i=1}^{n} G(\alpha_i)\\
&\textit{subject to } 0 \leq \alpha_i \leq 1 \textit{ for all } i = 1,\ldots,n\\
&\textit{and } \sum_{i=1}^{n} \alpha_i y_i = 0.
\end{aligned}
$$

4 MOEA for Large Margin Learning

After all objectives and constraints for the multi-objective setting of large margin learning are defined, we will discuss some details in this section. In the following, we will concentrate on the details for Support Vector Machines but everything can also be applied for Kernel Logistic Regression as well.

4.1 Definition of the Objectives

The Problems 4 and 5 can be used as objectives for the MOEA. Both objectives share a common term, the sum $\sum_{i=1}^{n} \alpha_i$. Since this sum as part of the first objective is not conflicting with the second objective as a whole, we can simply omit the calculation of the sum of α_i for the first objective.

Another efficiency improvement can be achieved by formulating the problem with $b = 0$. All solution hyperplanes must then contain the origin and the equality constraints $\sum_{i=1}^{n} \alpha_i y_i = 0$ will vanish [Burges, 1998]. This is a mild restriction for high-dimensional spaces since the number of degrees of freedom is only decreased by one. However, during optimization we do not have to cope with this equality constraint and do not need to calculate it each generation anew.

If the equality constraint should be fulfilled (e.g. for small numbers of dimensions where omitting the constraint would make a difference), it can simply be defined as a third objective by maximizing $-\left|\sum_{i=1}^{n} \alpha_i y_i\right|$. The whole set of objectives is then given as a maximization of the terms (in the SVM setting):

$$-\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k\left(x_i, x_j\right),$$

$$\sum_{i=1}^{n} \alpha_i,$$

$$\text{and } -\left|\sum_{i=1}^{n} \alpha_i y_i\right|$$

subject to $\alpha_i \geq 0$ for all $i = 1, \ldots, n$.

4.2 Implementation: evoSVM

We developed a support vector machine based on evolution strategies optimization. Individuals are the real-valued vectors $\alpha = (\alpha_i, \ldots, \alpha_n)$. For mutation, we used the hybrid mutation proposed by [Mierswa, 2006a] to get sparser solutions, i.e. solutions where many α_i are zero. Crossover probability is high (0.9). The individuals are initialized with 0 to further support sparsity. The maximum number of generations is 1000. The population size is 100. We use NSGA-II as the multi-objective selection scheme [Deb *et al.*, 2002]. NSGA-II employs a selection technique which first sorts all individuals into levels of non-domination. Individuals from the first levels are added to the next generation until the desired population size is reached. Before adding individuals from the last possible level this level is sorted with respect to the crowding distance in order to preserve diversity in the population.

4.3 Selecting a Solution from the Pareto Set

The first idea of supporting the user in selecting a final solution from the Pareto front might be to just calculate the first objective in its original form and check which individual provides the highest value for

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k\left(x_i, x_j\right).$$

The corresponding model is the maximum margin model for the given data set without respecting the training errors since the values α_i were not bounded during the optimization. Although this solution is interesting in its own, this model is often not the desired one.

Alternatively, one could use another pointer to where in the Pareto front one should search for the final solution. We suggest to keep a small hold-out set of the data points of size k. These k data points were part of the input training set and are not used by the learner during the multi-objective optimization. After the optimization has finished and the Pareto front for all objectives is derived, the learner is applied to all k data points of the hold-out set. The prediction error for each individual is calculated with the binary loss

$$l(y, f(x)) = \begin{cases} 1 \text{ if } y \neq f(x) \\ 0 \text{ otherwise} \end{cases}$$

which leads to the error Err_p for the learned decision function f_p of the p-th individual:

$$Err_p = \sum_{q=1}^{k} l\left(y_q, f_p(x_q)\right).$$

Plotting all errors Err_p together with the training set errors results in another front which can be compared to the original Pareto front. The user should examine places where the training error and the generalization error are close together and should avoid areas where the generalization performance is much worse then the achieved objectives. The plots of both fronts together are a powerful tool to control overfitting: displaying the effect of overfitting in the generalization performance plot for all possible models ease the selection of an optimal model without getting in danger of too much overfitting.

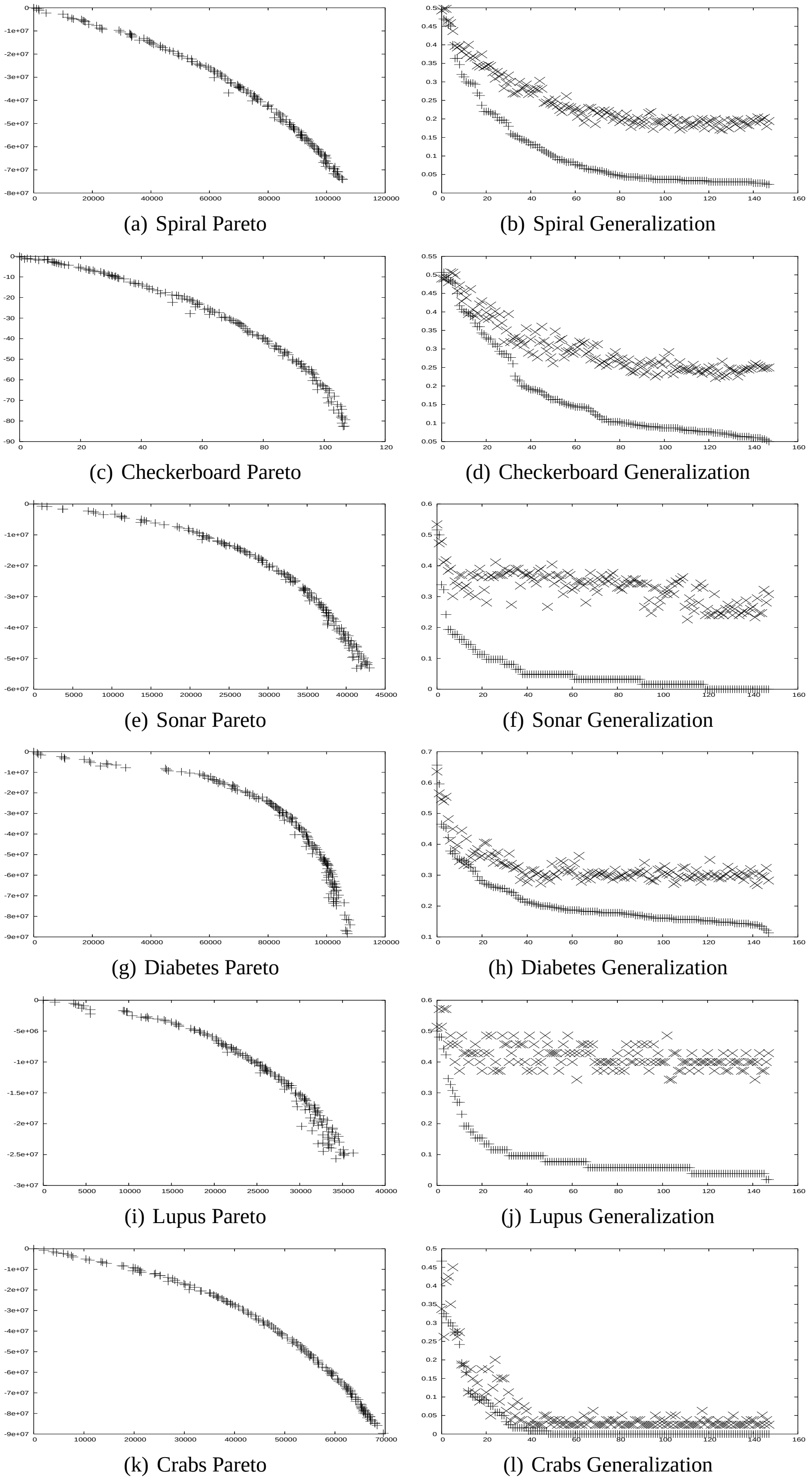
5 Examples

The experiments in this section do not prove the ability to solve the SVM problem with evolutionary algorithms which was already done by previous work [Mierswa, 2006a], [Mierswa, 2006b]. It has been shown that the proposed evolutionary optimization SVM frequently outperform the quadratic programming counterparts, especially for non-positive semidefinite kernels.

In this section, we show the benefit of the transformation of the original SVM problem into an efficient multi-objective formalization by showing the Pareto fronts for several benchmark data sets. We use a RBF kernel for all SVM and determine the best parameter value for σ with a grid search parameter optimization. Possible parameters were 0.001, 0.01, 0.1, 1 and 10. A description of all data sets together with the optimal kernel parameter value σ for each data set is given in Table 1. All experiments were performed with the machine learning environment YALE [Mierswa *et al.*, 2006][1], the new SVM implementation is called *evoSVM* within this framework.

Figure 1 shows all results. *The left plot* for each data set shows the resulting Pareto front delivered by the multi-objective evolutionary SVM proposed in this paper. The y-axis denotes the first optimization objective from Section 4.1 (margin size) and the x-axis shows the second objective (training error). The third objective is omitted in the plots for the sake of simplicity. *The right plot* shows the prediction errors for the training set and a hold-out test set (cf. Section 4.3). The x-Axis simply denotes a counter over all Pareto-optimal solutions found during the optimization ordered by their training errors. The y-axis denotes the prediction error for the training ($+$) and testing ($\times$) data, i.e.

[1] `http://yale.sf.net/`



(a) Spiral Pareto (b) Spiral Generalization

(c) Checkerboard Pareto (d) Checkerboard Generalization

(e) Sonar Pareto (f) Sonar Generalization

(g) Diabetes Pareto (h) Diabetes Generalization

(i) Lupus Pareto (j) Lupus Generalization

(k) Crabs Pareto (l) Crabs Generalization

Figure 1: The left plot for each dataset shows the Pareto front delivered by the SVM proposed in this paper (x: training error, y: margin size). The right plot shows the training (+) and testing (×) errors (on a hold-out set of 20%) for all individuals of the resulting Pareto fronts (x: Pareto solution counter, y: errors).

Data set	n	m	**Source**	σ	**Default**
Spiral	1000	2	Synthetical	1.000	50.00
Checkerboard	1000	2	Synthetical	1.000	50.00
Sonar	208	60	UCI	1.000	46.62
Diabetes	768	8	UCI	0.001	34.89
Lupus	87	3	StatLib	0.001	40.00
Crabs	200	7	StatLib	0.100	50.00

Table 1: The evaluation data sets. n is the number of data points, m is the dimension of the input space. The kernel parameter σ was optimized with a grid parameter search. The last column contains the default error, i. e. the error for always predicting the major class.

on the hold-out set. The hold-out test set was a randomly sampled subset of size 20% of the given training set.

The generalization ability plotted on the right side clearly shows the location where overfitting occurs and the training error is still minimized while the test error remains or get worse. You can detect this area in the right plots at places where the training error $(+)$ and the testing error $(\times)$ diverge. Since the x-axis in the right plots correspond to a counter of solutions in the Pareto front, ordered by its training errors which corresponds to the x-axis in the left plot, you can find the interesting solutions in the Pareto front in the same area as on the right side.

Please note that these types of plots could also be achieved for other learning schemes (e.g. usual SVM or Logistic Regression) by iteratively applying the learner for different parameter settings and produce the set of models in this way. The approach proposed in this paper has the advantage that all models are calculated in one single run which is far less time-consuming.

6 Conclusion

Recently, evolutionary computation was connected with statistical learning theory. The idea of large margin methods was very successful in many applications from machine learning and data mining. Embedding evolution strategies as the optimization scheme of a Support Vector Machine results in even better learning methods which frequently outperform traditional SVM. This is especially true in the case of learning with non-positive semidefinite kernel functions where traditional SVM implementations are not able to find an optimum in feasible time.

In this paper, we demonstrated how the trade-off between training error and model complexity can be made explicit for general large margin methods. We demonstrated this explicit form of regularization for two different learning methods, namely Support Vector Machines and Kernel Logistic Regression. We divided the optimization problem of these problems in two parts and transformed both parts into its dual form of its own. These transformations reduce the runtime for fitness evaluation and provide space for other well-known improvements like incorporating arbitrary kernel functions for non-linear classification tasks.

We exploited the new objectives by employing a multi-objective evolutionary algorithm after some consequences of the explicit regularization were discussed. These include the possibility of further reducing the runtime by using only parts of the objectives and the optional usage of a hold-out set in order to produce a hint which areas of the resulting Pareto front should be inspected by the user. This turns the Pareto front of all solutions between minimal training error and minimal model complexity into a powerful tool for controlling the overfitting of machine learning methods. Please note that all information about these plots are collected in one single run of the algorithm in contrast to wrapper approaches where the learner must be performed once for each point of such an overfitting plot.

The idea of statistical learning theory, i.e. taking the model complexity into account, is simple and appealing. Although the idea of regularization is nowadays used in almost all new learning methods, current approaches, however, did not make use of the inherent trade-off but demanded the definition of a weighting factor of the conflicting criteria from the user. The multi-objective evolutionary SVM and KLR proposed in this paper are the first solutions explicitely solving the basic problem of statistical learning theory.

7 Acknowledgments

This work was supported by the *Deutsche Forschungsgemeinschaft (DFG)* within the *Collaborative Research Center “Reduction of Complexity for Multivariate Data Structures”*.

References

[Beyer and Schwefel, 2002] H.-G. Beyer and H.-P. Schwefel. Evolution strategies: A comprehensive introduction. *Journal Natural Computing*, 1(1):2–52, 2002.

[Burges, 1998] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[Camps-Valls *et al.*, 2004] G. Camps-Valls, J.D. Martin-Guerrero, J.L. Rojo-Alvarez, and E. Soria-Olivas. Fuzzy sigmoid kernel for support vector classifiers. *Neurocomputing*, 62:501–506, 2004.

[Deb *et al.*, 2002] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: Nsga-ii. Technical report, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology, 2002.

[Friedrichs and Igel, 2004] F. Friedrichs and C. Igel. Evolutionary tuning of multiple svm parameters. In *Proc. of the 12th European Symposium on Artificial Neural Networks (ESANN 2004)*, pages 519–524, 2004.

[Frp̈hlich *et al.*, 2004] H. Frp̈hlich, O. Chapelle, and B. Schölkopf. Feature selection for support vector machines using genetic algorithms. *International Journal on Artificial Intelligence Tools*, 13(4):791–800, 2004.

[Haasdonk, 2005] B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, 2005.

[Hastie *et al.*, 2001] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2001.

[Howley and Madden, 2005] T. Howley and M.G. Madden. The genetic kernel support vector machine: Description and evaluation. *Artificial Intelligence Review*, 2005.

[Joachims, 2005] T. Joachims. A support vector method for multivariate performance measures. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 377–384, 2005.

[Jun and Oh, 2006] Sung-Hae Jun and Kyung-Whan Oh. An evolutionary statistical learning theory. *International Journal of Computational Intelligence*, 3(3):249–256, 2006.

[Keerthi *et al.*, 2005] S. S. Keerthi, K. B. Duan, S. K. Shevade, and A. N. Poo. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 61(1–3):151–165, 2005.

[Kennedy and Eberhart, 1995] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proc. of the International Conference on Neural Networks*, pages 1942–1948, 1995.

[Kimeldorf and Wahba, 1971] G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

[Lin and Lin, 2003] H.-T. Lin and C.-J. Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods, March 2003.

[Mierswa *et al.*, 2006] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. YALE: Rapid prototyping for complex data mining tasks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, 2006.

[Mierswa, 2006a] I. Mierswa. Evolutionary learning with kernels: A generic solution for large margin problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*, 2006.

[Mierswa, 2006b] Ingo Mierswa. Making indefinite kernel learning practical. Technical report, Collaborative Research Center 475, University of Dortmund, 2006.

[Runarsson and Sigurdsson, 2004] T.P. Runarsson and S. Sigurdsson. Asynchronous parallel evolutionary model selection for support vector machines. *Neural Information Processing*, 3(3):59–67, 2004.

[Schölkopf and Smola, 2002] B. Schölkopf and A. J. Smola. *Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[Smola *et al.*, 1998] A. Smola, B. Schölkopf, and K.-R. Müller. General cost functions for support vector regression. In *Proceedings of the 8th International Conference on Artificial Neural Networks*, pages 79–83, 1998.

[Smola *et al.*, 2000] A. J. Smola, Z. L. Ovari, and R. C. Williamson. Regularization with dot-product kernels. In *Proc. of the Neural Information Processing Systems (NIPS)*, pages 308–314, 2000.

[Vapnik, 1998] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[Vert *et al.*, 2004] J.-P. Vert, K. Tsuda, and B. Schölkopf. *Kernel Methods in Computational Biology*, chapter A primer on kernel methods, pages 35–70. MIT Press, 2004.

TheMoT - A Theme Monitoring Tool for Text Streams

Rene Schult
Otto-von-Guericke-University Magdeburg
D-39116, Magdeburg, Germany
schult@iti.cs.uni-magdeburg.de

Abstract

In this paper we will show our ThemeFinder. This algorithm is a method to monitor cluster labels of different clusterings, specially clusterings of a data stream at different time periods. This algorithm is the basis for our new developed ”Theme Monitoring Tool”. This Theme Monitoring Tool is an intuitive graphical user interface to use the ThemeFinder and present the results of the monitoring process in two different ways. One way is a short result list to get a fast overview of the monitoring results and the second way is a long result list to get a detailed view of the mapped results and the labels of the mapping labels at both clusterings.

1 Introduction

Stream clustering considers two different perspectives: On the one hand, one is focused on grouping multiple independent streams of signals (e.g. signals from multiple sensors) into clusters and adapting or monitoring these clusters while the senders continue delivering signals. On the other hand, data of complex structures or texts can arrive as a single stream, the contents of which are grouped into clusters according to similarity; cluster monitoring is then devoted to identify changes in these clusters, as the data flow continues. In this paper, we take the second approach on stream clustering and cluster monitoring into account and study the impact of different clustering algorithms on cluster evolution. For monitoring clusters evolution we focus on monitor labels of clusters.

As a case study, we consider the ACM archive[1], which uses the ACM taxonomy for keyword assignment, categorization and browsing as well. A stream of documents is added to the archive and is ranged to the existing taxonomy. This taxonomy could be considered as the labels/topic of the clusters of the documents under this taxonomy point. The ACM taxonomy has been expanded with subjects like “data mining” and “image databases” under the existing subject “database applications” to assign the document to the appropriate taxonomy. All existing documents in “database applications” are not automaticially assigned to the new created subcategories like “data mining” or “image databases”. If a knowledge seeker is interested on early advances on data mining, he or she has to go through the whole subarchive on database applications. Although keyword-based search is available, the appropriate keywords for search on data mining in the early nineties are likely to be different from those of today. A retrospective *re-categorization* of the documents or at least the discovery of the keywords characterizing them is needed.

We propose the monitoring of cluster evolution by focusing on monitor cluster labels. We are aiming to find persistent labels of clusters or at least some changes at the labels in order to find the real cluster evolution. We have set up our experiments under the assumption that the terminology of the document archive changes over long time. Thus, we should be able to detect cluster changes over time. In our case study of the ACM archive, we have looked for evidence on the emergence of themes like “data mining” and for words that correspond to this theme like subtopics of this from the time it emerged till today. To detect temporal trends, we have also used a non accumulated archive in [Schult and Spiliopoulou, 2006b].

We have applied text clustering in an evolving feature space. It must be stressed that a classification is not appropriate for this problem: Classification (even adaptive classification) requires a labeled dataset. Here, the challenge lays more in identifying documents that adhere to a yet unknown subject, with these documents having been assigned to some more generic class label. Moreover, themes consists of words in a feature space that must be adapted as the language of the documents’ authors evolves. In [Schult and Spiliopoulou, 2006b] and in [Schult and Spiliopoulou, 2006a] we present an algorithm, the ThemeFinder , to detect such “themes” at cummulated and non-cummulated datasets and show it with experiments on clustering the data with a bisecting k-means algorithm. We have also shown the influence of the used clustering algorithm to the functionality and the quality of the ThemeFinder and to the quality of the detected cluster evolutions.

We have shown in different papers that the algorithm is robust against different algorithms for clustering and labeling methods. So we decide to produce a complete tool to the ThemeFinder, because the usage of a algorithm is much easier and so the goal of the ThemeFinder will be followed. Via our tool it is much easier to use the ThemeFinderalso by management persons to get a fast overview of the results. Here we will present a Theme Monitoring Tool , TheMoT, as the developed tool to the ThemeFinder.

In the next section, we have discussed relevant research, specially to the nature of the ThemeFinder, to monitor labels or patterns over time. In section 3, we first have short present the ThemeFinder. Section 4 we have shown the TheMoT, the Theme Monitoring Tool and shown the usage of the TheMoT. The last section concludes our work.

[1] http://portal.acm.org/ccs.cfm

2 Related Work

Relevant research on text mining can be organised into the following categories: Discovery of emerging concepts in accumulating document collections, assignment of labels to clusters and comparison of clusters build over possibly distinct populations.

The subjects of Topic Detection and Topic Tracking are defined in [Allan, 2002], where the five tasks of TDT are enlisted. As stated in that book, TDT concentrates on the detection and tracking of *stories* (a "topic" is a story) and encompasses the tasks of (1) story segmentation, (2) first story detection, (3) cluster detection, (4) tracking and (5) story link detection. However in this paper we address the influence of different clustering algorithms on the quality of label monitoring that is not the focus of TDT works.

Moringa et al. [Moringa and Yamanishi, 2004] and Wang and McCallum [Wang and Andrew McCallum, 2006] present different methods for detecting topics over time, but in [Moringa and Yamanishi, 2004] only persistent topics over the whole time are detected and in [Wang and Andrew McCallum, 2006] they have the assumption that topics never change over time, which differ from our point of view.

For the identification of systematic changes in the data, Ganti et al. propose the DEMON framework for data evolution and monitoring across the temporal dimension [Ganti *et al.*, 2000]. DEMON focuses on detecting systematic vs. non-systematic changes in the data and on identifying the data blocks (along the time dimension) which have to be processed by the miner in order to extract new patterns. However, the emphasis is on updating the knowledge base by detecting changes in the data, rather than detecting changes in the patterns. The closely related framework FOCUS of the same group is designed to compare two datasets and compute an interpretable, qualifiable deviation measure between them [Ganti *et al.*, 1999a]. Bartolini et al also propose a generic framework for the comparison of patterns in general and clusters in particular [Bartolini *et al.*, 2004], allowing for the incorporation of application-specific comparison measures. Both this and the FOCUS framework though are designed for arbitrary clusters, not taking the prominent role of cluster labels into account for the comparison.

Close to the nature of a cluster label is the notion of "summary" used by the CACTUS algorithm [Ganti *et al.*, 1999b], which complements the DEMON+FOCUS suite: CACTUS exploits summaries upon datasets as the basis of "well-defined" clusters, which can then be discovered by only two passes over each of the datasets under consideration [Ganti *et al.*, 1999b].

The basic idea of mechanism of concept indexing [Karypis and Han, 2000] or latent semantic indexing [Deerwester *et al.*, 1990] is that the importance of a component can be derived from the weight it receives by an analysis, like clustering [Hotho *et al.*, 2003]. Here we rank the importance of words for label the cluster results based on the number of documents of the cluster which contains the word.

Relevant results about evaluating cluster results are the reasearch of [Banerjee and Langford, 2004]. Also the comparision of different evalution measures for clustering at [Stein *et al.*, 2003] and [Halkidi *et al.*, 2001; Halkidi and Vazirgiannis, 2001] are very interesting for our research and future work.

3 ThemeFinder on an Accummulated Document Collection

Our algorithm, denoted as ThemeFinder hereafter, takes as input an accumulating document collection, typically a bibliographical archive, thematically categorized on subject according to a slowly evolving taxonomy [2]. It monitors this archive over a series of time periods and tries to discover *persistent thematic subcollections* and assign labels to them. These labels are meant as themes that should extend the original taxonomy, so that the discovered subcollections become represented in it as new classes. Since the subcollections are already present as clusters with associated labels, no explicit labelling and no classifier training are needed.

A document is described as a vector of words derived from a feature space. We do not observe the documents in their entirety, but concentrate on title, keywords and a limited number of sentences (e.g. from the document's abstract), assuming that this part of the document is particularly designed to disseminate the content to the reader in a compact way [3].

In the first period, ThemeFinder clusters all documents in the collection and builds thematic clusters. In subsequent periods, only the new documents are considered. Simply adding them to the clusters would be counterproductive though, since the sheer size of the original collection would suppress emerging trends. Hence, ThemeFinder *re-clusters* the new documents with the original feature space and then juxtaposes the clusters thus found with the ones found in the previous period. If the clusters of two adjacent periods are thematically similar *and* if the quality of the clustering is not degenerating, then the original feature space is still representative of the collection. Otherwise, a new feature space is generated for the documents of the current period and the juxtaposition process is restarted at the next period. By the end of the observation time interval, thematic clusters that have survived over several periods despite the re-clustering and despite the change of the feature space are considered stable enough to become part of the classification scheme.

In the following, we first specify the formal model describing the archive as an accummulating collection of documents over time periods, as well as the notions of *(cluster) label* and *collection theme*. We then describe how TheMeFinder monitors the accummulating documents, detects label changes and initiates the recomputation of the feature space.

3.1 Modeling Documents and Labels over Time

Let $\mathcal{A}$ denote a subset of the document archive, which encompasses the documents assigned to the same theme of the original thesaurus. For simplicity of notation, we refer to $\mathcal{A}$ as "the archive" hereafter, although it corresponds to a well-defined subset of the actual archive. Further, let $\mathcal{W}$ be the set of all words in $\mathcal{A}$.

We observe the archive $\mathcal{A}$ over a series of T time periods $t_1, \ldots, t_T$, whereupon each period t_i encompasses a subset of documents D_i such that $\cup_{i=1}^{n} D_i = \mathcal{A}$ and $D_i \cap D_j = \emptyset \forall i \neq j$. Hence, in each period t_i, the documentset D_i contains the documents that have been inserted

[2] The ACM taxonomy is a typical example. We used it in our experiments

[3] PageRank of Google also checks only a small part of the document, including header, preample etc.

in the archive during this period.

Definition 1 (Feature Space): *L*et D be a collection of documents and W be the set of words in these documents. Then, the "feature space" over D, $FS(D)$ is the set of the n "dominant" words in W, which we define as the words with the highest TFxIDF values in D.

Then, for each time period $t_i, i = 1 \ldots, T$ we define the "period-specific feature space" $FS_i \equiv FS(D_i)$ as the set of dominant words over D_i.

By this definition, the size of the feature space remains constant across the time spectrum of observations but the contents of the period-specific feature space may change from period to period.

Definition 2 (Document Vector): *L*et D be a set of documents and $fs = \{w_1, \ldots, w_n\}$ be a feature space. Then, for each document $d \in D$, its "document vector" in fs consists of the TFxIDF values of the words in fs over D:

$$v(d, fs) =< tfidf(w_1), \ldots, tfidf(w_n) >$$

By this definition, a document can be associated with several vectors, one per feature space. In particular, for each document d in the documentset D_i of period t_i, we can define the document vector of d over the period-specific feature space FS_i or over another feature space $FS_j, j \neq i$. In fact, ThemeFinder first attempts to cluster the documents of a period t_i thematically using the feature space of the previous periods and changes the feature space only if the thematic clusters thus built are not satisfactory.

Definition 3 (Thematic Cluster): *L*et D be a collection of documents and fs be a feature space. Let $\zeta(D, fs) = \{C_1, \ldots, C_k\}$ be a clustering, i.e. a set of clusters that partition D into k non-overlapping groups of similar document vectors over the feature space fs.

A cluster $C \in \zeta(D, fs)$ is a "thematic cluster" if the following set is not empty:

$$L_C = \{w \in fs | support(w, C) \geq \tau_{wordsupport}\} \tag{1}$$

In this definition, $support(w, C)$ returns the fraction of documents in cluster C that contain the word w, divided by the number of documents in C, $card(C)$. The threshold $\tau_{wordsupport}$ restricts the set L_C to the words of the feature space that are frequent in C, i.e. those that characterize the documents in C.

Definition 4 (Cluster Label): *L*et D be a collection of documents, fs be a feature space and $\zeta(D, fs) = \{C_1, \ldots, C_k\}$ be a clustering over D as in Def.3 . Let $C \in \zeta(D, fs)$ be a cluster. The "label" of C, denoted as $label(C)$ is the set of words in $L_C \cup \{e\}$, where L_C is defined in Eq. 1 and e is the empty word.

Thus, a thematic cluster consists of documents characterized by a set of words from the feature space. These words constitute the cluster's label, which is a candidate as theme of the collection. If this set of words is empty, then $label(C) = \{e\}$, i.e. C has an "empty label".

Definition 5 (Collection Theme): *L*et $t_1, \ldots, t_T$ be the series of T periods of observation over the accumulating archive. Let D_i denote the set of documents in period t_i, X_i be the feature space used in this period and $\zeta(D_i, X_i)$ be the clustering of D_i over X_i.

A set of words $TP \subseteq \mathcal{W}$, chosen among the words of the archive $\mathcal{A}$, is a *collection theme* iff:

$$\forall i = 1, \ldots, T \exists C^i \in \zeta(D_i, X_i) : label(C^i) = TP$$

This definition says that a cluster label that persists over all periods of the collection is appropriate for addition to the thesaurus. It must be stressed that the feature space of t_i is not necessarily the period-specific one. This conforms to our approach of retaining a feature space once specified, unless labels disappear. On the other hand, we expect that a collection theme should survive changes in the feature space.

According to Def.5, a label becomes a "collection theme" iff it appears in all periods of observation. This is a very restrictive definition: First, we expect that a set of words would make a good label if it appears in an adequately large number m of periods for some threshold value m. Second, the terminology associated with a theme is not static: Especially during a peak of activity on a new theme, terminology may change rapidly as authors are looking for representative terms *and* as the borders between the new theme and other subject areas are being redefined. For example, the subject area now known as "data mining" used a slightly different terminology (and dominant terms) in 1995 than now, ten years later. This indicates that the label of clusters that refer to the same theme may undergo changes. Therefore, we relax some of the requirements of Def.5 as follows:

Definition 6 (Theme): *L*et $t_1, \ldots, t_T$ be the series of T periods of observation over the accumulating archive. Let D_i denote the set of documents in period t_i, X_i be the feature space used in this period and $\zeta(D_i, X_i)$ be the clustering of D_i over X_i.

As before, let $TP \subseteq \mathcal{W}$ be a set of words chosen among the words of the archive $\mathcal{A}$ and let $m \geq T$ be a threshold value. The set TP is a "theme" iff there are m periods $t_{i_1}, t_{i_2}, \ldots, t_{i_m}$ such that: $\forall j = i_1, i_2, \ldots, i_m \exists C^j \in \zeta(D_j, X_j) : card(TP - label(C^j) \cap TP) \leq \tau_{deviation}$

This definition classifies a label as a theme if some of its words appear in at least m arbitrary, not necessarily consecutive periods. The threshold $\tau_{deviation}$ determines how many of the words may deviate. By setting $\tau_{deviation} := 0$ and $m := T$, we come to the original definition of a collection theme.

Different implementations and deviations from this heuristic-based definition are possible: For example, we may require that a minimum number among the m periods are consecutive or that m refers to the *last* m periods $t_{T-m}, t_{T-m+1}, \ldots, t_T$. We may also require that a cluster C^j may be considered against TP only if the most frequent word in $label(C^j)$ is in TP, thus restricting the candidate clusters considered for each TP at each period. The function $extract\_themes()$ of ThemeFinder contains a heuristic implementation of Def.6.

3.2 Tracing Labels over Time

ThemeFinder for theme-tracing is presented in Table 1 and explained in the rest of this section.

The Main Procedure of the ThemeFinder

ThemeFinder starts with the establishment of a clustering at period t_1 for the documentset D_1. It is assumed that clustering at this period is performed on the basis of the period-specific feature space $FS_1 \equiv FS_{D_1}$. Then, the first set

```
Step  Action
1     $fs \leftarrow FS_1$; $fs_{orig} \leftarrow fs$
2     $\zeta_1 \leftarrow \zeta(D_1, fs)$
3     $L_1 \leftarrow \{label(c)|c \in \zeta_i\} - \{e\}$
4     $Collectionthemes \leftarrow L_1$; $subL_1 \leftarrow L_1$
5     for $i = 2, \ldots, T$ do
6       $\zeta_i \leftarrow \zeta(D_i, fs)$; $subL_i \leftarrow \emptyset$; $matches = 0$
7       if $thematic\_clusters(\zeta_i) \geq \tau_{thematic}$
8         then
9          $\xi \leftarrow \zeta_i$
10         for each $c \in \zeta_{i-1}$ do
11          if $label(c) == \{e\}$ then continue
12          $c' \leftarrow best\_match(c, \xi)$
13          if $c' \neq \emptyset$ then
14           $\xi \leftarrow \xi - \{c'\}$; $matches++$;
15           $l \leftarrow label(c) \cap label(c')$
16           if $card(label(c)) - card(l) < \tau_{deviation}$ then
               $subL_i \leftarrow subL_i \cup \{(c, c', l\}$
17          endif
18         endfor
19      endif
20      if $matches < \tau_{matches}$
21        then
22         $fs_{orig} \leftarrow fs$; $fs \leftarrow FS_i$
23         $\zeta_i \leftarrow \zeta(D_i, fs)$
24      endif
25      $L_i \leftarrow \{label(c)|c \in \zeta_i\} - \{e\}$
26      $Collectionthemes \leftarrow Collectionthemes \cap L_i$
27    endfor
28    $themes \leftarrow extract\_themes(\cup subL_i, m)$
```

Table 1: ThemeFinder for label tracing

of labels is derived over $\zeta(D_1, FS_1)$. For each subsequent period $t_i, i \geq 2$, the algorithm first builds the clustering ζ_i over D_i using the feature space of the previous period(s) (originally: FS_1).

At step 7, ThemeFinder checks whether the clustering ζ_i contains an adequate number of thematic clusters according to Def.3, subject to a threshold $\tau_{thematic}$. If this is not the case, then the original feature space is not describing the collection adequately: The period-specific feature space becomes the current feature space (Step 10) and D_i is re-clustered with it (Step 11).

If ζ_i does contain sufficiently many thematic clusters, then ThemeFinder juxtaposes ζ_i to the clustering of the previous period ζ_{i-1} and identifies pairs of similar clusters (Steps 9-18). In step 12, the function $best\_match(\cdot)$ described below returns for each cluster in ζ_{i-1} the cluster of ζ_i with the most similar label, or the emptyset if no such cluster exist.

If two clusters match, we retain the common part of their labels in step 16. The threshold $\tau_{deviation}$ determines the number of words in the old cluster label that may disappear from the new label, according to Def.6. This information is used in step 28 to extract themes.

If there is a sufficient number of good matches among the thematic clusters[4], subject to a threshold $\tau_{matches}$, then the feature space is still good for the collection. Otherwise, the feature space is replaced and re-clustering is performed in steps 22, 23.

In step 25, ThemeFinder stores the labels of the clusters in the final clustering ζ_i. In step 26, it records labels that persist in the sense of Def.5 and reports them to the expert by the end of period t_T: The set $Collectionthemes$ contains the collection themes. The function $extract\_themes$ in step 28 returns the set of labels adhering to the weaker definition of theme in Def.6.

Cluster Matching and Evaluation

For a good clustering ζ_i at timepoint t_i, ThemeFinder juxtaposes the clusters in it with those of the previous clustering ζ_{i-1} and tries to match thematic clusters of ζ_{i-1} to those in ζ_i. Thematic clusters thus matched in all (or most) periods represent stable partitions/classes in the archive.

Since thematic clusters are clusters with a non-empty label by definition (cf. Def.3), the simplest case of a match for a cluster $C \in \zeta_{i-1}$ would be a cluster $C' \in \zeta_i$ such that $label(C) = label(C')$. In many cases, though, this criterion is too restrictive. For example, if $label(C)$ consists of 8 words and 7 among them constitute $label(C')$, then it is likely that the two clusters contain similar documents, although their labels are not identical. The heuristic $best\_match(\cdot)$ in step 12 of Table 1 returns for a cluster C the cluster of ζ' that is most similar to it in content and label, as shown in Table 2.

```
Step  Action in $best\_match(C, \xi)$
1     $candidates \leftarrow \emptyset$
2     for each $X \in \xi$ do
3       if $label(X) == label(C)$ then
4         return $X$
5       endif
6       if $label(X) \cap label(C) \neq \{e\}$ then
7         $candidates \leftarrow candidates \cup \{X\}$
8       endif
9     endfor
10    if $candidates == \emptyset$ then
11      return $\emptyset$
12    endif
13    $L \leftarrow ordering(label(C), MFWF)$
14    for each $w \in L$ do
15      $wL \leftarrow \{X \in candidates | w \in label(X) \&$
            $support(w, X) \approx support(w, C)\}$
16      if $wL \neq \emptyset$ then
17        $candidates \leftarrow wL$
18      endif
19    endfor
20    $L \leftarrow ordering(candidates, MCWF)$
21    return $firstOf(L)$
      $MFWF = Most\_Frequent\_Word\_First$
      $MCWF = Most\_Common\_Words\_First$
```

Table 2: The heuristic *best_match* of ThemeFinder

The heuristic $best\_match(C, \xi)$ in Table 2 is actually a series of heuristics applied upon the set of clusters ξ. In step 3 it is checked whether there is a cluster with the same label as C. If this is the case, the procedure returns this cluster and ends. Otherwise, a list of candidates is built, consisting of the thematic clusters having at least one common word with the label of C (Steps 6, 7). If there are no such candidates, the empty set is returned in step 11. If there are candidates, then they are filtered on the basis of the frequency of the words in their labels (Steps 13-19).

The motivation of ordering the words in the label of C by frequency (cf. step 13) is that frequent words inside the

[4] Clusters not adhering to Def.3 do not count here.

cluster are likely to be more important. Then, starting with the most frequent word in step 14, a subset of candidates is identified in step 15: These are the clusters, in which the word appears in the label *and* has a similar support as in the cluster C. If this set is not empty, it replaces the original set of candidates (steps 16-17). In any case, the next most frequent word is processed in the next iteration (step 14).

Steps 10 and 16 guarantee that the set of candidates considered in step 20 is not empty. In this step, the candidates are ordered by number of common words between their label and the label of C. Then, in step 21, the cluster with the most common words is returned as best match.

Identifying themes

The last step of ThemeFinder in Table 1 extracts the themes of the collection, using the sets of labels of the thematic clusters identified for each period (Step 13 of Table 1). The function $extract\_Themes$ takes as input a set of triplets and a threshold m. Each triplet (c, c', l) consists of a cluster c from period t_i, its best match in period t_{i+1}, say c' and the intersection of their labels (Step 16 of Table 1). Here are some possible implementations:

- A set of words l is a theme if it appears in m periods.
- A set of words l is a theme if it appears in the periods $t_{T-m}, \ldots, t_T$.
- A set of words l is a theme if it appears in m (arbitrary or consecutive) periods *and* for each period t_i: If c' is the best match of c in period t_{i-1}, then there exists a best match c'' of c' in t_i, with $l = label(c') \cap label(c'')$.

4 TheMoT - the ThemeMonitoring Tool

At the following we describe the developed Theme Monitoring Tool. We decided to use the programming language Python (www.python.org) because it is very easy and fast to implement a prototyp with this language. The development was done at a Intel machine with 1 GHz and 1GB RAM under Ubuntu linux.

We decided to produce two access points to the implementation of our algorithm, the graphical user interface and one console based access for use it in other scripts or tools. So the tool has two main classes. One class is the algorithm class with the code for the ThemeFinder. This class is also executable from a console and has to be called with the three parameters, the two cluster label files of the different clusterings and the mapping parameter. The second main class is the class for the graphical user interface, which present a window for selection of the label files and selection of the mapping parameter and than call the algorithm class via ”start” button and present the results in the text area of the main window.

Here we concentrate on the representation of the graphical user interface.

After start the TheMoT the user find the interface shown at figure 1. There the user see to file input lines including a button for a file selection dialog. After that it can be seen a slider. The user can set the percentage of the mapping value. It means how many percentage of a label at clustering 1 must be refind at a label at clustering 2 so that we say the label of clustering 1 exist also in clustering 2. In bottom of the slider, a blank text box is presented and underlying that the user find the button ”start”, the button ”exit” and the button ”show details” for start TheMoT, exit the tool and to show details of the short results, especially hide the details at the long results list. The button ”show details” is disabled at the start of TheMoT and will be enabled after the first run of ThemeFinder.

Format for the Input Files

Two label files are needed as input for TheMoT. This files must be text files, which has as content the labels of a clustering. Each line of such input file must be one label of one cluster of the clustering. The number of the lines is also the number of clusters at this clustering. At each label, the words, which are members of a label, has a word statistic in parenthesis at the end and are separated by commas. The following is an example of a label for a cluster: *”secur(0.54),statist(0.47),queri(0.31)”*.

Start the ThemeFinder

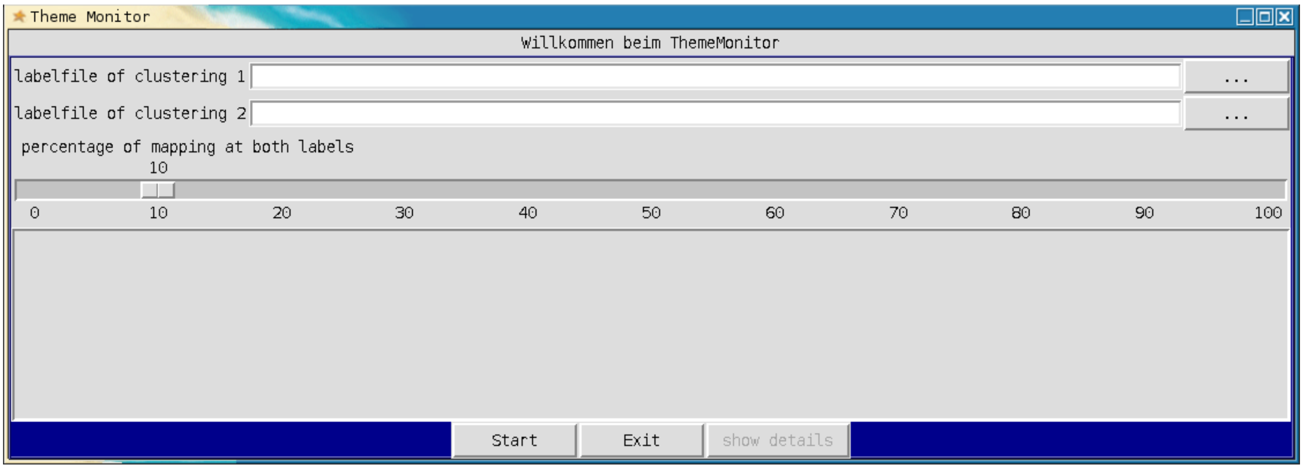
To start the ThemeFinder the user select the both labeling files of the different clusterings, he will monitor. Than the user select the percentage of the mapping of the labels and press the start button. After starting the ThemeFinder the result of the monitoring process will be shown in the text box under the slider for the percentage of label mapping. First the user see the short result list as an overview. It will be shown the cluster numbers of the first clustering grouped by the label which was refound at the second clustering. Also the user see, which clusters of the first cluster are not found again, shown at section ”died” at the result window and which clusters of the second clustering are not a mapping part to the first clustering and so we called this clusters as ”born” at the result window. You see this at figure 2. After the first run of ThemeFinder, through press button ”start” after select two cluster label files, the button ”show details” is enabled.

If the user will see more details of the mapping result, he press the button ”show details”. The text window with the results will be cleaned and the long result list will be shown. The long result list is a list of the labels and not only of the cluster numbers. It is shown which label of clustering one map to which label of clustering two. The user see the labels of the cluster in the three categories as in the short result list, mapped labels, died labels, born labels. At the mapped labels category, the words of the labels, which are equal at both labels, are colored green, the rest red. So the user can fast deteced the mapping parts of the labels. The button ”show details” has changed to a button ”hide details”, which switch back from the long result list to the short result list. A long result list can be seen at figure 3.

5 Conclusion

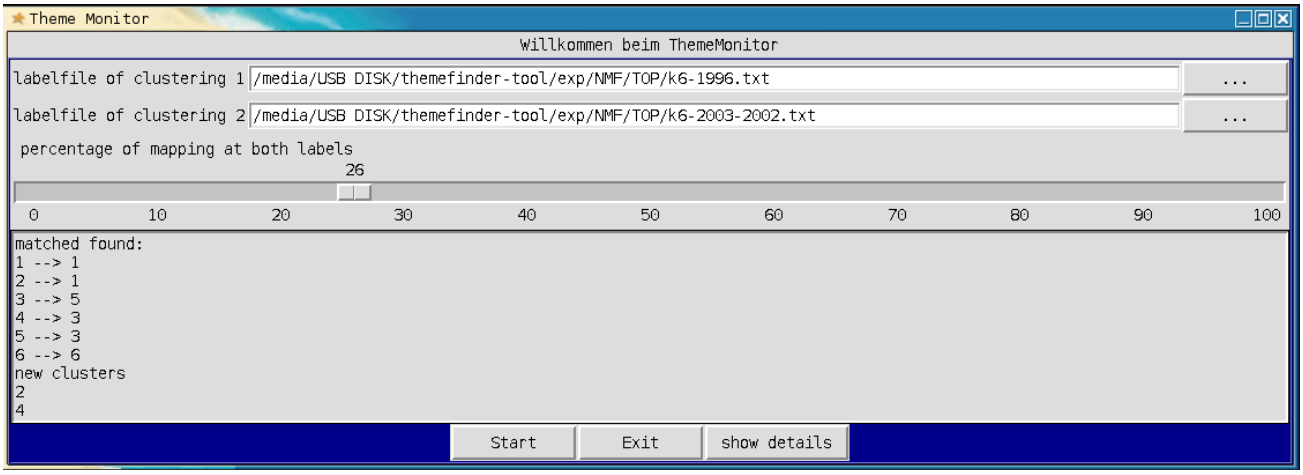
At this paper we present a algorithm for monitoring cluster labels over time, the ThemeFinder. Secondly the present the first prototyp of the Theme Monitoring Tool as a graphical user interface and a console interface to ThemeFinder and to present the results in a short and a detailed view. in the short future, we will build the possiblility for the user to select a value for the word statistic of a word at a label. This should be give the possibility to say only the words with a word statistic above this value should be used as member of a label.

Secondly in the future we will build a connection to the ”Text Clustering Toolkit (TCT)” from Derek Greene and the machine learning group of the University College of Dublin. This is a framework for text clustering, including different algorithms for clustering, different preprocessing tasks and different labeling methods.



labelfile of clustering 1
labelfile of clustering 2
percentage of mapping at both labels
10
0 10 20 30 40 50 60 70 80 90 100

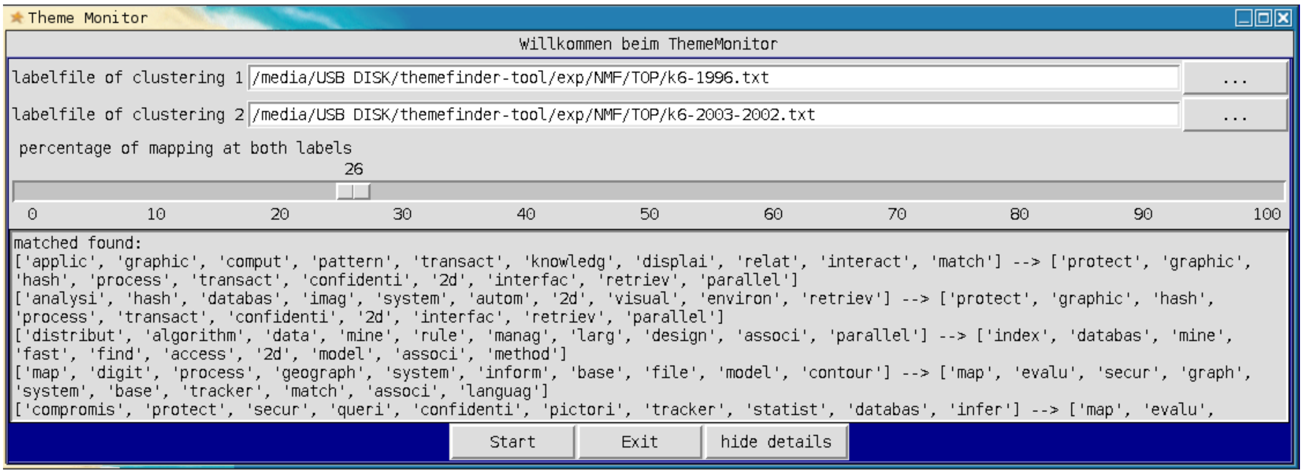
Figure 1: TheMoT at start



labelfile of clustering 1 /media/USB DISK/themefinder-tool/exp/NMF/TOP/k6-1996.txt
labelfile of clustering 2 /media/USB DISK/themefinder-tool/exp/NMF/TOP/k6-2003-2002.txt
percentage of mapping at both labels
26
0 10 20 30 40 50 60 70 80 90 100

```
matched found:
1 --> 1
2 --> 1
3 --> 5
4 --> 3
5 --> 3
6 --> 6
new clusters
2
4
```

Figure 2: TheMoT with short results list



labelfile of clustering 1 /media/USB DISK/themefinder-tool/exp/NMF/TOP/k6-1996.txt
labelfile of clustering 2 /media/USB DISK/themefinder-tool/exp/NMF/TOP/k6-2003-2002.txt
percentage of mapping at both labels
26
0 10 20 30 40 50 60 70 80 90 100

```
matched found:
['applic', 'graphic', 'comput', 'pattern', 'transact', 'knowledg', 'displai', 'relat', 'interact', 'match'] --> ['protect', 'graphic',
'hash', 'process', 'transact', 'confidenti', '2d', 'interfac', 'retriev', 'parallel']
['analysi', 'hash', 'databas', 'imag', 'system', 'autom', '2d', 'visual', 'environ', 'retriev'] --> ['protect', 'graphic', 'hash',
'process', 'transact', 'confidenti', '2d', 'interfac', 'retriev', 'parallel']
['distribut', 'algorithm', 'data', 'mine', 'rule', 'manag', 'larg', 'design', 'associ', 'parallel'] --> ['index', 'databas', 'mine',
'fast', 'find', 'access', '2d', 'model', 'associ', 'method']
['map', 'digit', 'process', 'geograph', 'system', 'inform', 'base', 'file', 'model', 'contour'] --> ['map', 'evalu', 'secur', 'graph',
'system', 'base', 'tracker', 'match', 'associ', 'languag']
['compromis', 'protect', 'secur', 'queri', 'confidenti', 'pictori', 'tracker', 'statist', 'databas', 'infer'] --> ['map', 'evalu',
```

Figure 3: TheMoT with details results list

Acknowledgments

I would like to say a great thanks to my professor Myra Spiliopoulou for so much help with my research. Also thanks to Gamal Kassam and Stefan Kiltz for many hours of discussing different problems of my experiments.

References

[Allan, 2002] J. Allan. *Introduction to Topic Detection and Tracking*. Kluwer Academic Publishers, 2002.

[Banerjee and Langford, 2004] Arindam Banerjee and John Langford. An Objective Evaluation Criterion for Clustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 515–520, July 2004.

[Bartolini *et al.*, 2004] Ilaria Bartolini, Paolo Ciaccia, Irene Ntoutsi, Marco Patella, and Yannis Theodoridis. A unified and flexible framework for comparing simple and complex patterns. In *Proc. of ECML/PKDD 2004*, Pisa, Italy, Sept. 2004.

[Deerwester *et al.*, 1990] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 44(6):391–407, 1990.

[Ganti *et al.*, 1999a] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. A Framework for Measuring Changes in Data Characteristics. In *Proceedings of the 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 126–137, Philadelphia, Pennsylvania, May 1999. ACM Press.

[Ganti *et al.*, 1999b] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. CACTUS: Clustering categorical data using summaries. In *Proc. of 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '99)*, pages 73–83, San Diego, CA, Aug. 1999. ACM Press.

[Ganti *et al.*, 2000] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. DEMON: Mining and Monitoring Evolving Data. In *Proc. of the 15th Int. Conf. on Data Engineering (ICDE'2000)*, pages 439–448, San Diego, CA, USA, Feb. 2000. IEEE Computer Society.

[Halkidi and Vazirgiannis, 2001] Maria Halkidi and Michalis Vazirgiannis. Clustering Validity Assessment: Finding the optimal partitioning of a data set. In *Proceedings of IEEE - Internationa Conference on Data Mining (ICDM) Conference*, November 2001.

[Halkidi *et al.*, 2001] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Cluster Validity Methods: Part1. *SIGMOD Records*, June 2001.

[Hotho *et al.*, 2003] Andreas Hotho, Steffen Staab, and Gerd Stumme. Explaining Text Clustering Results using Semantic Structures. In *Proceedings PKDD2003*, 2003.

[Karypis and Han, 2000] George Karypis and Eui-Hong (Sam) Han. Fast Supervised Dimensionality Reduction Algorithm with Apllications to Document Categorization & Retrieval. In *Proceedings of CIKM-00*, pages 12–19. ACM Press, New York, US, 2000.

[Moringa and Yamanishi, 2004] Satoshi Moringa and Kenji Yamanishi. Tracking Dynamics of Topic Trends Using a Finite Mixture Model. In Ronny Kohavi, Johannes Gehrke, William DuMouchel, and Joydeep Ghosh, editors, *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 811–816. ACM Press New York, NY, USA, August 2004.

[Schult and Spiliopoulou, 2006a] Rene Schult and Myra Spiliopoulou. Discovering emerging topics in unlabelled text collections. In *Advances in Databases and Information Systems, 10th East-European Conference,(ADBIS'2006)*, pages 353–366. Springer Verlag, September 2006.

[Schult and Spiliopoulou, 2006b] Rene Schult and Myra Spiliopoulou. Expanding the Taxonomies of Bibliographic Archives with Persistent Long-Term Themes. In *Procedings of the 21th Annual ACM Symposium on Applied Computing (SAC'06)*. ACM, ACM Press, April 2006.

[Stein *et al.*, 2003] Benno Stein, Sven Meyer zu Eissen, and Frank Wißbrock. On Cluster Validity and the Information Need of Users. In M.H. Hanza, editor, *3rd IASTED Int. Conference on Artificial Intelligence and Applications (AIA03)*, pages 216–221, Benalmadena, Spain, September 2003. ACTA Press.

[Wang and Andrew McCallum, 2006] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Procedings of KDD06*, Philadelphia, Pennsylvania, USA, August 2006. ACM.

Position Paper: Ontology Learning from Folksonomies

Dominik Benz, Andreas Hotho
Knowledge & Data Engineering Group (KDE), University of Kassel,
Wilhelmshöher Allee 73, 34121 Kassel, Germany
http://www.kde.cs.uni-kassel.de
{benz,hotho}@cs.uni-kassel.de

Abstract

The emergence of collaborative tagging systems with their underlying flat and uncontrolled resource organization paradigm has led to a large number of research activities focussing on a formal description and analysis of the resulting "folksonomies". An interesting outcome is that the characteristic qualities of these systems seem to be inverse to more traditional knowledge structuring approaches like taxonomies or ontologies: The latter provide rich and precise semantics, but suffer - amongst others - from a knowledge acquisition bottleneck. An important step towards exploiting the possible synergies by bridging the gap between both paradigms is the automatic extraction of relations between tags in a folksonomy. This position paper presents preliminary results of ongoing work to induce hierarchical relationships among tags by analyzing the aggregated data of collaborative tagging systems as a basis for an ontology learning procedure.

1 Introduction

A fundamental aspect of knowledge management is often the establishment of structure within a set of information resources, e.g., PDF documents, bookmarks or photographs. Most traditional approaches address this issue by decomposing the domain under consideration into interrelated classes or categories, which are intended to model exhaustively the underlying knowledge structure. Each available information resource is then assigned to one or more classes. Ontologies are a well-known formalism for this purpose. The hierarchical topic category structure of, e.g., a web directory like the Open Directory Project[1] can be seen as an example of a taxonomy, which constitute a core component of ontologies [Staab and Studer, 2004]. Their widespread use is however hindered by the expertise and cost required for their creation and maintenance.

Collaborative tagging systems feature another structuring paradigm: Each user can assign one or more arbitrary keywords (or *tags*) to each of his resources, facilitating a flat "by-keyword" access to personal or public resources. The resulting structure of users, tags and resources became known as *folksonomies* [Mathes, 2004]; refer to [Hotho *et al.*, 2006] for a formal definition. Due to their inherent simplicity and immediate usefulness, these systems are able to overcome the previously described knowledge acquisition bottleneck. However, this comes at the cost of a lack of precision (see [Golder and Huberman, 2006]), which is exactly the strength of ontological approaches.

As a first step towards unleashing synergies by automatically learning ontologies from folksonomies, this position paper proposes an algorithm to induce hierarchical relationships among tags. The algorithm has been tested with real-world user data from the social music sharing platform *Last.fm*[2], and the outcome has been evaluated against a gold-standard music style hierarchy taken from the comprehensive online music directory *MusicMoz*[3].

2 Inducing Hierarchical Relations among Tags

The goal of this work is to automatically induce a concept hierarchy, i.e., a tree structure, whose nodes (representing concepts) each consist of one or more tags from a folksonomy. Concept specificity increases with increasing depth in the tree, and there exists only a single type of relation, whose semantics resembles closely the one of the taxonomic relation [Bozsak *et al.*, 2002].

Data foundation The most often used information source is based on two types of so-called *tag-tag-cooccurrence networks*, which can be extracted from a folksonomy. Each existing tag corresponds to a node, and there exists a undirected edge with weight w_{ij} between two tags t_i and t_j if

- there were w_{ij} users who have used both t_i and t_j to annotate any of their resources (*user-based tag-tag-cooccurence, UTC*)
- there were w_{ij} resources both annotated with t_i and t_j by any user (*resource-based tag-tag-cooccurence, RTC*)

Classes of approaches Existing approaches based on tag cooccurrence information can be assigned to one of the following three classes:

- *Social Network Analysis:* [Mika, 2005] pioneered in applying centrality and other measures like the clustering coefficient coming from social network analysis to the UTC and RTC networks in order to identify broader and narrower terms. [Heymann and Garcia-Molina, 2006] proposed betweeness centrality as tag generality measure. The latter approach will serve as a basis for the proposed algorithm.

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

[2] http://www.last.fm

[3] http://www.musicmoz.org

- *Statistical approaches:* The work of [Schmitz, 2006] and [Schmitz *et al.*, 2006] is based on statistical models of tag subsumption, the latter is corroborated with the theory of association rule mining.
- *Clustering approaches:* Starting from a similarity measure between tags, clustering approaches like [Begelman *et al.*, 2006] identify groups of highly related tags. Depending on the chosen clustering algorithm, a hierarchical relationship between the tag clusters is established.

Proposed Algorithm The proposed algorithm is an extension of the work of [Heymann and Garcia-Molina, 2006]. It comprises the following steps:

1. Filter the tags by an occurrence threshold τ_{occ}.
2. Order the tags in descending order by generality (measured by degree centrality [Hoser *et al.*, 2006] in the UTC network).
3. Starting from the most general tag, add all tags t_i subsequently to an evolving tree structure:
   (a) Identify the most similar existing tag t_{sim} (using the weights w_{ij} in the UTC network as similarity measure).
   (b) Decide whether t_{sim} and t_i are synonyms or form a compound expression (using an adapted statistical model of subsumption from [Schmitz, 2006] based on the RTC network).
   (c) If yes → merge t_{sim} and t_i, otherwise append t_i as a less general term underneath t_{sim}.

Compared to the original algorithm, the first extensions consists of applying a computationally much less complex centrality measure (namely degree centrality) as tag generality measure (step 2). The original measure is based on betweenness centrality, whose computation requires $O(nm + n^2 \log n)$ time [Brandes, 2001], whereby n is the number of tags and m is the number of edges in the weighted cooccurrence network. This dimension becomes problematic when applied to real-world large scale folksonomy systems. Degree centrality in contrast can be computed in linear time $O(n)$, as it mainly comprises counting the edges for each node in the cooccurrence network.

As a further extension, tag synonymy and compound expressions (e.g., *"open"* and *"source"*) are considered (step 3.b) instead of single-tag concepts. The model applied hereby does not increase the overall algorithm complexity.

3 Assessing the Quality of Learned Relations

Choosing a gold-standard based evaluation paradigm, it is a non-trivial task to judge the similarity between a learned concept hierarchy and a reference hierarchy, especially regarding the absence of well-established and universally accepted evaluation measures. As a detailed description of the similarity measures used is beyond the scope of this paper, the reader is referred to [Dellschaft and Staab, 2006] for an overview. Two of the described measures, namely taxonomic precision / recall / F_1-measure and the OntoRand-Index were adapted to compare two hierarchies on an instance-based level: The underlying idea is that two concept hierarchies are very similar if they structure the resources in question in a similar manner.

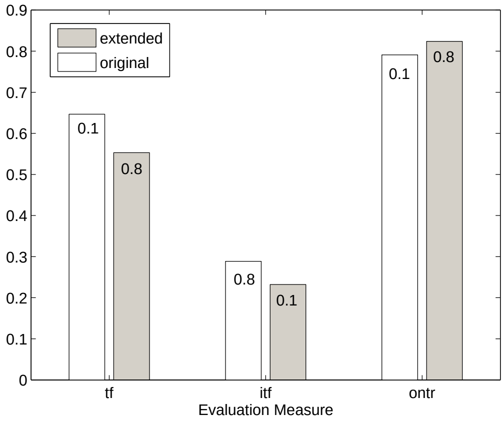


Figure 1: Experimental Results: Comparison of the performance of the proposed algorithm with the original version. The numbers in the bars correspond the optimal parameters for each algorithm as found in the first test phase.

4 Preliminary Experimental Results

In order to validate the proposed algorithm, experiments were conducted with a dataset crawled from the social music sharing website *Last.fm*[4]. It consists of 978 resources (i.e., music artists), 3585 users and 7283 tags, connected by 162406 tag assignments. As a gold standard, a music style hierarchy (built by volunteer music fans) consisting of 548 styles was downloaded from *MusicMoz*[5]. In this dataset, each artist from the Last.fm dataset is assigned to 1-3 MusicMoz style categories.

The experimental setup consisted of two phases:

1. Parameter optimization for both the original and the proposed algorithm;
2. Comparison of the performance of both algorithms with the obtained optimal parameters, compared by the taxonomic F_1-measure (tf), the instance-based taxonomic F_1-measure (itf) as well as the extended OntoRand-Index $ontr$.

Figure 1 displays the results. For none of the given measures, there is a clear winner. An important issue when interpreting the differing assessments of the measures is their respective basis: The taxonomic F_1-measure (tf) compares two hierarchies based on matching concept names, while the instance-based taxonomic F_1-measure (itf) and the extended OntoRand-index are based on the assignment of information resources to each concept. It is obvious that the two latter measures are strongly influenced by the chosen resource assignment strategy.

Considering the fact that the proposed algorithm is computationally much less complex (see Section 2) compared to its original version, the results are acceptable. To get a better impression of the capabilities of the proposed algorithm, Figure 2 illustrates its outcome. Following paths from the hierarchy root towards the leafs, the styles become more and more specific. Starting from the *ROOT* node in the center of the image, one nice example is the path *rock* → *metal* → *death metal* → *progressive death metal* towards the lower left corner.

[4] http://www.last.fm

[5] http://www.musicmoz.org

5 Conclusions and Further Work

This paper presented preliminary results of ongoing work on inducing hierarchical relationships among tags in a folksonomy as basis for an ontology learning procedure. Experiments with real-world data suggest that the proposed algorithm is able to produce a consistent hierarchical category scheme, which comes close to a handcrafted scheme. An open issue for future research is how to assess the quality of the gold-standard the outcome of the learning procedure is compared with. A deeper theoretical understanding of the interaction of the algorithm's building blocks (i.e., tag generality measure, tag similarity measure and tag subsumption measure) is needed in order to further improve the results. Another aspect that needs consideration is how the resources of the folksonomy are assigned to the resulting hierarchical structure.

References

[Begelman *et al.*, 2006] Grigory Begelman, Philipp Keller, and Frank Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006*, Edinburgh, Scotland, May 2006.

[Bozsak *et al.*, 2002] E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, and V. Zacharias. Kaon - towards a large scale semantic web. In K. Bauknecht, A. Min Tjoa, and G. Quirchmayr, editors, *E-Commerce and Web Technologies, Third International Conference, EC-Web 2002, Proceedings*, volume 2455 of *LNCS*, pages 304–313, Berlin, 2002. Springer.

[Brandes, 2001] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[Dellschaft and Staab, 2006] Klaas Dellschaft and Steffen Staab. On how to perform a gold standard based evaluation of ontology learning. In *Proceedings of ISWC-2006 International Semantic Web Conference*, Athens, GA, USA, November 2006. Springer, LNCS.

[Golder and Huberman, 2006] Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *Journal of Information Sciences*, 32(2):198–208, April 2006.

[Heymann and Garcia-Molina, 2006] Paul Heymann and Hector Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical report, Computer Science Department, Standford University, April 2006.

[Hoser *et al.*, 2006] Bettina Hoser, Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Semantic network analysis of ontologies. In *European Semantic Web Conference, Budva, Montenegro*, June 2006.

[Hotho *et al.*, 2006] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, June 2006. Springer.

[Mathes, 2004] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata, December 2004.

[Mika, 2005] Peter Mika. Ontologies are us: A unified model of social networks and semantics. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *The Semantic Web - ISWC 2005, Proceedings of the 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10*, volume 3729 of *Lecture Notes in Computer Science*, pages 522–536. Springer, 2005.

[Schmitz *et al.*, 2006] Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. iberna, editors, *Data Science and Classification. Proceedings of the 10th IFCS Conf.*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Heidelberg, July 2006. Springer.

[Schmitz, 2006] Patrick Schmitz. Inducing ontology from flickr tags. In *Proceedings of the Workshop on Collaborative Tagging at WWW2006*, Edinburgh, Scotland, May 2006.

[Staab and Studer, 2004] Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.

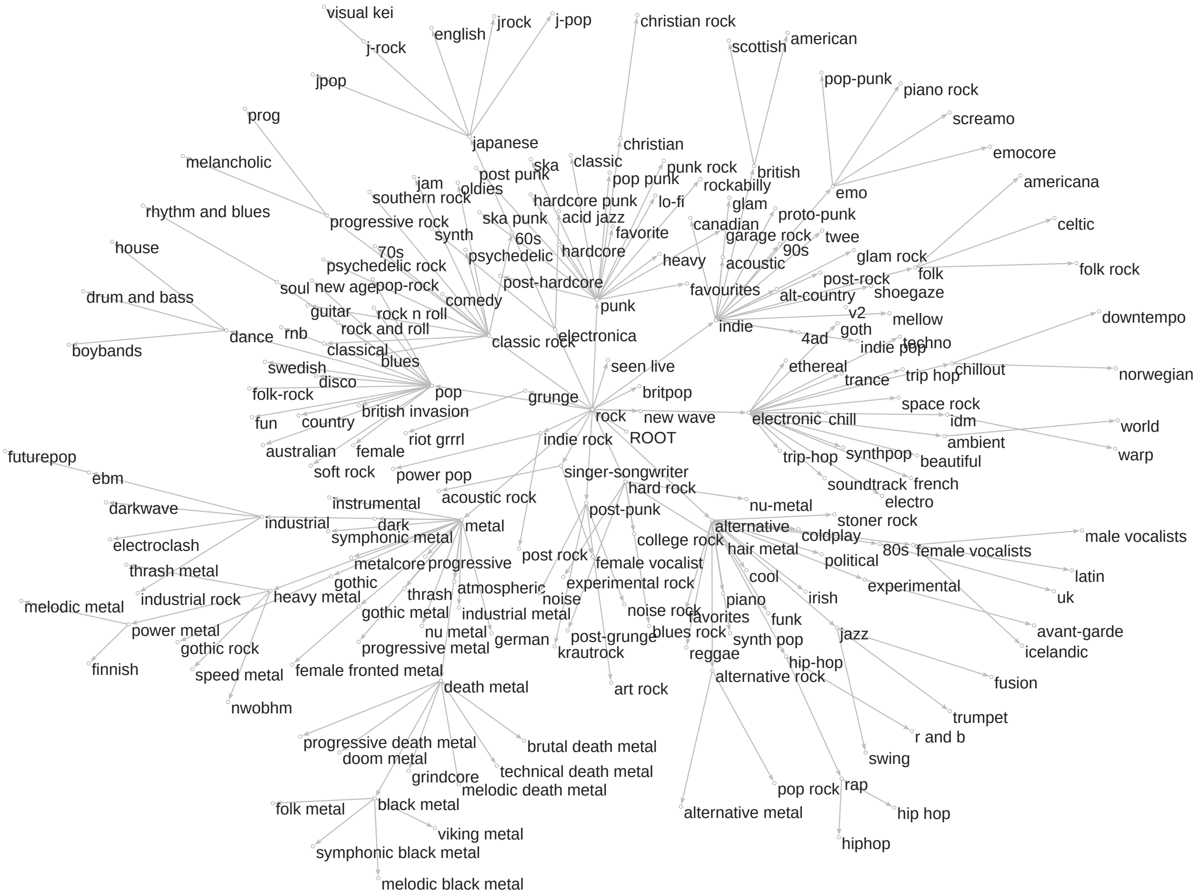


Figure 2: Music style hierarchy extracted from Last.fm dataset by the proposed algorithm.

Using Neural Nets for Semi-Automatic Semantic Annotation of Images

Suzanne Little
Institute of Computer Vision and
Applied Computer Sciences,
Leipzig, Germany
Suzanne.Little@ibai-institut.de

Ovidio Salvetti
ISTI, CNR,
Pisa, Italy
Ovidio.Salvetti@isti.cnr.it

Petra Perner
Institute of Computer Vision and
Applied Computer Sciences,
Leipzig, Germany
pperner@ibai-institut.de

Abstract

Detailed, consistent semantic annotation of large collections of multimedia data is difficult and time-consuming. In domains such as eScience, digital curation and industrial monitoring, fine-grained high-quality labeling of regions enables advanced semantic querying, analysis and aggregation and supports collaborative research. Manual annotation is inefficient and too subjective to be a viable solution. Automatic solutions are often highly domain or application specific, require large volumes of annotated training corpi and, if using a 'black box' approach, add little to the overall scientific knowledge. This article evaluates the use of simple artificial neural networks to semantically annotate micrographs and discusses the generic process chain necessary for semi-automatic semantic annotation of images.

1 Introduction

Semantic annotation of media is recording high-level descriptive terms about the content of the media. It may be coarse-grained (descriptions at the image level) or fine-grained (descriptions at the segment or region level). Manual annotation (i.e., annotation by a human expert) is expensive, time-consuming, inconsistent and subjective. Tools and algorithms are available that can automatically extract low-level feature data from media objects such as color, shape, size, trajectory etc. However, these features are insufficient to support the queries required by domain experts who prefer to access data using higher-level terms such as catalyst, gaseous microemboli or mitochondria. Bridging the distance (often called the "multimedia semantic gap") between automatically extracted low-level features and high-level semantic terms is the focus of a great deal of research.

The concept of the "semantic gap" was initially discussed in psychology [Finke, 1989] and refers to the distance between information that can be extracted from the visual data and the interpretation that different users have for this same data. The difficulty is that humans cannot always describe what they see or explain why they interpret it in a certain way. Also there is rarely complete agreement between a group of users about the interpretation of media content. Because of this, choosing the algorithms to extract low-level information from the media is also more complicated as we do not know what the key features are and therefore what procedure to apply. Overcoming or mitigating this semantic gap to enable rapid and accurate semantic annotation of images is of particular importance in domains such as biology, geology, astronomy and industrial monitoring. Fields like these use high-resolution, high-throughput sensors and analytical machines to produce very large volumes of media content. Researchers need to be able to analyze and label objects of interest within these images and manage the quantity of data they provide.

This article discusses an approach to addressing the semantic gap using artificial neural networks in the context of the requirements for semi-automatic semantic annotation of scientific media. It summarizes of some of the main approaches to bridging the semantic gap (section 2) and outlines the motivations and characteristics of a sample application (section 3). We present a generic processing chain for classifying images (section 4) and discuss the key challenges. Section 5 describes the use of artificial neural networks to classify image regions of a 3D slice sequence and presents the results of an initial evaluation.

2 Related Work

A variety of research efforts have investigated the use of techniques to extract semantic labels from different low-level visual or audio features. These approaches range from interfaces to facilitate user-driven annotation to systems integrating formalized knowledge structures, prototype based applications and automatic classification using machine-learning technologies. This section presents some examples of the more common approaches.

Interfaces that assist users to annotate or to link semantic terms to examples or sets of visual features are one approach to addressing the semantic gap. M-Ontomat-Annotizer [Petridis et al., 2006] from the aceMedia project provides a graphical user interface for experts to link ontologies with low-level media features. The Rules-By-Example interface [Little and Hunter, 2004; Hunter and Little, 2005] also enables expert users to define mappings from low-level MPEG-7 features to high-level semantic terms using semantic web technologies such as OWL and RuleML/SWRL.

More traditional approaches have used machine-learning techniques such as statistical analysis, hidden markov models or artificial neural networks to determine semantic terms based on sets of low-level features. Chang et al. [1998] applied a library of examples approach, which they call semantic visual templates. Zhao and Grosky [2002] employ a latent semantic indexing technique which integrates a variety of automatically extracted visual features (global and sub-image color histograms and anglograms for shape-based and color-based representations) to enable semantic indexing and retrieval of images. Adams et al. [2003] manually annotate atomic audio and video features in a set of training videos and from these develop explicit statistical models to automatically label the video with high-level semantic concepts. Work by Naphade et al. [Naphade *et al.*, 2000; Naphade and Huang 2001] proposed a statistical factor graph framework to bridge the gap between low-level features and semantic concepts. IBM alphaWorks have developed a tool, MARVEL [IBM alphaWorks, 2006; Natsev *et al.*, 2005], for "Multimedia Analysis and Retrieval" that applies heuristic techniques to automatically label image and video repositories based upon semantic models derived from sets of training examples.

Another example of a statistical approach to semantic labeling of images is found in Kyrgyzov et al. [2007] who apply a criterion based on Minimum Description Length (MDL) using low-level texture descriptions to cluster and thus classify images. Like other statistically-based methods, this approach requires a significant training set and is unable to incrementally learn as new data is generated and added to the dataset.

Colantonio et al. [2006] and Di Bona et al. [2003] use neural networks to segment and characterize medical images. Previous work by Perner et al. [2001] has shown that neural networks to tend generalize better than other methods and can model non-linear decision surfaces. However, neural networks require a labeled training set of suitably significant size and variation. In contrast, a decision tree is a method that can easily be trained but they do not generalize as well as neural nets. If the variation in the data is very high then the preferred method would be case based reasoning [Perner, 2007a; Perner, 2007b]. This method does not necessarily generalize; it relies on samples and can incrementally learn. Unlike neural networks, both decision trees and case-based reasoning have the capability to explain their classifications.

The use of taxonomies or ontologies, either in combination with a user interface or with machine-learning approaches, enables reasoning using the relationships defined in the ontology and query expansion for better searching over the annotated dataset. Benitez and Chang [2003] exploit structural semantics to label media objects by using WordNet [Fellbaum, 1998] as a source of keywords in combination with Bayesian networks to provide media classification. Hollink et al. [2003] use multiple ontologies to support the annotation of art images and applied a similar approach [Hollink *et al.*, 2005b] to the annotation of news videos by combining links between visual features in a multimedia ontology (MPEG-7) and general semantic concepts defined in Wordnet. Bloehdorn et al. [2005] have also used an MPEG-7 based ontology to formalize the relationships between high- and low-level visual features and semantic terms by recording 'prototype' instances that define the visual feature values.

The commonalities between these different approaches include the requirement for appropriate extraction of sets of low-level media features, the need for high-level, preferably well-defined, semantic terms to label and classify the media. The approaches that employ machine-learning techniques usually require a significant set of examples for training. The next section of this article presents an example application that needs support for automatic or semi-automatic annotation of image regions but is not able to initially supply a training set of statistically significant size.

3 Application

At the Institute for Molecular Bioscience of the University of Queensland, the Visible Cell project [Hunter *et al.*, 2004; Marsh *et al.*, 2001] aims to increase understanding of the mammalian cell via the synthesis of physical data, models, mathematical and statistical simulations, and bioinformatics data. The objective of the project is to provide a visualization environment that seamlessly embeds macro-molecular structures, networks and quantitative simulations based on mathematical and complex-system models into a 3D mammalian cell reconstructed from high resolution tomograms and electron micrographs.

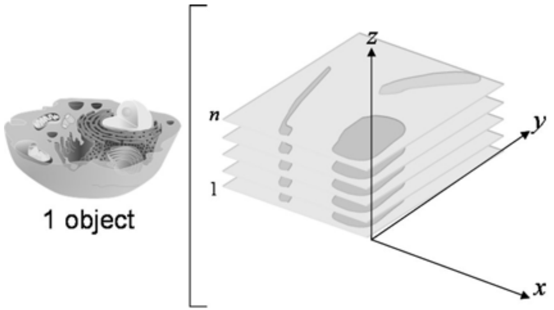


Figure 1: 2D slices of a 3D object[1]

Figure 1 illustrates the data in this application which consists of 2D micrographic digital images of thin slices taken sequentially through a cell from a pancreas (a 3D object). We had 31 digital images from one cell. These 2D slices are used to create a 3D model of the sub-cellular structure. To build this 3D model and support the type of integrated, semantic queries desired by users, we need to label the single sub-cellular objects within the 2D slices. The objects of interest in this project are: endoplasmic reticulum (ER), golgi (Go), mature granules (MG), mitochondria (Mi), ribosome (Ri) and tubular vesicles (TV).

A sample image is shown in Section 4.2, Figure 2. Finding automatic image segmentation algorithms for images like this one is not easy since the cell is highly structured. Parallel research at the IMB [IMB] is working on algorithms for automated segmentation. However, until efficient and accurate automatic segmentation algorithms are able to be developed the method of choice is to manually label the objects. This means that a human user is sitting in front of a computer and circles objects of interest they can detect. The correct label for each object is not always obvious to the user and can only be determined after examining other images in the 3D stack to find adjacent regions that could be more easily identified. The user was able to label 7580 regions in the set of 31 slices. This is only the regions that the user was able to manually circle and identify – we cannot be certain that the user was able to identify all possible objects correctly.

To correctly label all these regions is time-consuming and difficult for a human user. Therefore an approach to automate part or all of the procedure is necessary. Initially, work using semantic inferencing rules was extended to evaluate the Rules-By-Example interface [Little and Hunter, 2004; Hollink *et al.*, 2005a]. In this article we

[1] 3D cell structure adapted from http://commons.wikimedia.org/wiki/Image:Biological_cell.svg

use artificial neural networks (ANN) since they are widely used in image classification [Perner *et al.*, 2001] to discuss the generic process for semantic annotation and classification of images.

Our dataset consists of 5548 entries. Of the 7580 regions labeled by the expert, 2032 were identified using class labels that were not used in this evaluation being of minor objects that were of less interest. Each entry represents a region with the set of extracted low-level features and the user-assigned label. The distribution of the classes is shown in Table 1. The evaluation of the network, in terms of the accuracy rate, was done by test and train. The dataset was divided into sets of 70% for training and 30% for testing. The neural network architecture used was a simple feedforward network trained using a backpropogation algorithm.

Table 1. Overall Sample Distribution (test and training sets)

	ER	Go	Mi	MG	Ri	TV
Number of Objects	2486 (45%)	1463 (26%)	221 (4%)	105 (2%)	1125 (20%)	148 (3%)

The features in the dataset are: Area, ConvexArea, FilledArea, MajorAxisLength, MinorAxisLength, Eccentricity, Solidity, Extent and DominantColor. Excluding DominantColor, which was calculated separately, these terms are extracted using MATLAB's regionprops function from the Image Processing Toolbox. Through discussion with the expert users, general features such as size and shape were noted as being of particular importance and usefulness in distinguishing the different objects in the cell micrograph. This process of knowledge acquisition is not an easy procedure and requires experience to be able to extract relevant and useful knowledge from expert users. A methodology for doing so is presented in [Perner, 2002].

Future work includes converting the shape features used here to the more general MPEG-7 Shape Descriptors based on moments. MPEG-7 Region-base Shape Descriptors [Manjunath *et al.*, 2003] would have been useful in this instance but extractors for generating them were not available at the time. The MPEG-7 standard itself does not bridge the gap between the low-level features and the higher semantic terms. Using standard MPEG-7 features helps with interoperability and provides an abstraction hierarchy or taxonomy of the different features for viewing or reasoning. The standard does not provide this hierarchy, however work on ontologies that incorporate the MPEG-7 feature terms include [Hunter, 2005; Tsinaraki *et al.*, 2005; Garcia and Celma, 2005] and aims to achieve this level of structure. These proposed ontologies also add more intermediate terms to the media description vocabulary.

4 (Semi-)Automatic Annotation

The previous section presented an example application that would benefit from the ability to semi-automatically semantically annotate images. This section discusses the different levels of "semantic" labels and then describes the generic semantic annotation process for images.

4.1 Semantic Labels

Figure 2 shows the different types of semantic labels that can be applied to an image and gives some examples of the features and some possible values. The automatically or semi-automatically extracted *low-level* features have numerical values and, as such, are not easily understood or interpreted by a human user. The descriptors defined in the MPEG-7 standard [Manjunath *et al.*, 2003] are examples of this type of feature. However, MPEG-7 does not give us a sufficient level of semantic terms for the visual features. It concentrates on descriptors such as region-based shape, scalable color or homogeneous texture, etc.

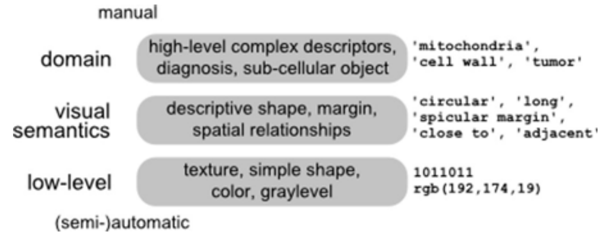

manual
domain
high-level complex descriptors, diagnosis, sub-cellular object
'mitochondria', 'cell wall', 'tumor'
visual semantics
descriptive shape, margin, spatial relationships
'circular', 'long', 'spicular margin', 'close to', 'adjacent'
low-level
texture, simple shape, color, graylevel
1011011
rgb(192,174,19)
(semi-)automatic

Figure 2: Overview of different levels of semantic annotations

Of more use for human interpretation are *visually descriptive* or *semantic features* which describe characteristics in more usable, often symbolic vocabularies and may be drawn from standards such as BIRADs [BIRADS]. The highest-level of semantic labels are *domain specific* descriptors. These may be defined in a domain ontology, taxonomy or standard such as MeSH [MeSH] or the Gene Ontology [Smith *et al.*, 2003] and are rich, descriptive terms about the content of the image.

The visual semantic terms can be used to define mappings to the domain level terms. For example, "if object is long and thin and close to an object that is identified as 'Golgi' then the object is a 'Golgi' ".

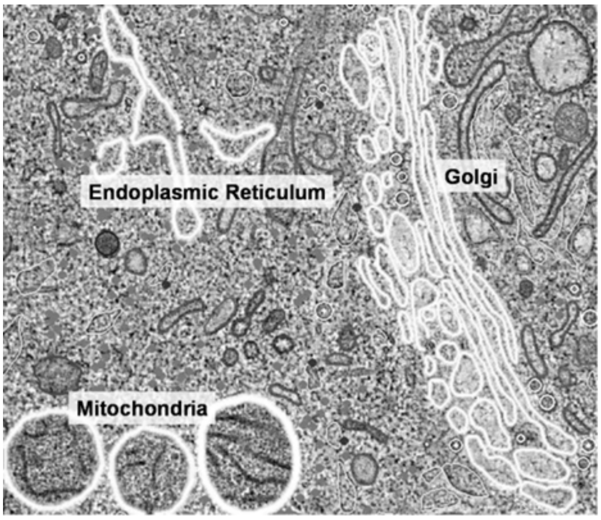


Figure 3: Sample segmented pancreas cell micrograph, example regions for Endoplasmic Reticulum, Golgi and Mitochondria have been highlighted.

In the example application described in section 3, the 2D slices contain objects that are described by human experts using both visual semantic terms and labeled or classified using domain terms. For example, the sections labeled 'Mitochondria' in Figure 3 have an uneven texture with distinct internal striations; they are circular in shape and generally large in area. In contrast the regions which make up the structure labeled 'Golgi' have less distinctive visual features in common; they are generally smooth in texture, often long and thin in shape. Their most distinctive visual characteristic is their spatial relationships as the regions tend to lie in long, parallel alignments close to

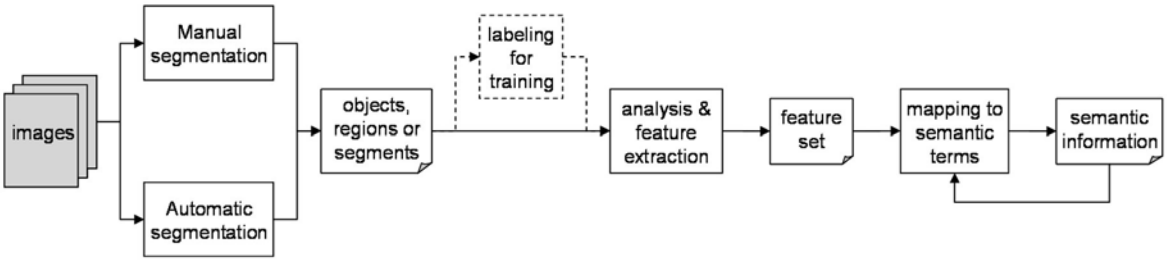


Figure 4: Generic process chain for image annotation or classification

each other. A further example of the importance of spatial relationships is the 'Endoplasmic Reticulum' regions which are only distinguishable by the presence of the small ribosome region touching it. As the ribosome generally only touches the endoplasmic reticulum object at one point in 3D space, it is often only able to be identified in a small subset of the 2D slices. Its presence in other slices is inferred through the adjacency of regions along the z-plane.

As you can see these descriptions are not numerical values but rather intermediate descriptors used to describe the objects of interest in terms of their shape, texture and location within the complete scene. In order to generate the high-level domain terms, useful for querying and to assist in generating the final 3D models, semantic mappings need to be developed from the low-level to the intermediate terms and then from the intermediate to the domain level descriptions. The next section describes a generic process chain for image annotation and classification.

4.2 The Generic Process Chain Necessary for Semi-Automatic Semantic Annotation

The process of semantically annotating images based on low-level features usually follows a common abstract procedure. The generic processing chain for image understanding is shown in Figure 4.

The first step is segmentation where the image is segmented into background and objects of interest. Ideally algorithms for automatic segmentation should be used to analyze and divide the images. However, devising a general procedure for segmenting images is not always possible and sometimes manual or semi-automatic processes are required.

Once we have determined what image pixels belong to which object, we need to label the regions representing the objects. Using these regions, we can then extract the object features (e.g., low-level features such as graylevel, simple shape features, texture and color) to produce a feature set for each object.

These low-level features need to be mapped to intermediate semantic terms such as 'circular', 'long', 'angular margin', 'spicular margin', etc. These semantic terms, more familiar to an expert user, can be used to describe the class or category of an object in the image. This is the first phase of mapping to semantic terms.

To achieve fine-grained, high-level descriptions of objects is sometimes only possible by taking into account intermediate descriptors such as spatial information about location or relation to other objects within the image. Coarse-grained descriptions of the complete scene require the grouping of objects and describing their spatial relation to each other. The intermediate level semantic terms need to be mapped to domain level terms which describe the content depicted in the image. This is the second phase of mapping to semantic terms.

There is no universal algorithm available that can automatically process the semantic information for all kinds of images. This means that specific images need special processing functions in order to implement the processing chain described in this section.

5 Results and Discussion

Table 2 shows the evaluation results of the neural network using the test data set of 1627 objects. User Labeled (User) is the number of objects in the test set that were labeled as a specific 'class' by the expert user. Objects Classified (ANN) is the number of objects in the test set that the network classified as 'class' while Objects Correctly Classified (OCC) is the number of the Objects Classified whose class label corresponds with that given by the expert user. Precision, Recall and f-measure are terms from the information retrieval domain [van Rijsbergen, 1979]. Precision and recall are both measurements of classification quality. f-measure provides a more useful measure of the overall performance since it takes into account the generally opposing qualities of precision and recall (i.e. high recall generally results in lower precision and vica versa). They are calculated as follows:

- Precision (P) is Objects Correctly Classified / User Labeled
- Recall (R) is Objects Correctly Classified / Objects Classified
- f-measure (f-m) is (2*Recall*Prec.)/(Recall + Prec.)

Table 2. Results of simulating the trained neural network using the test data set.

Class	User	ANN	OCC	P	R	f-m
ER	435	482	226	0.469	0.520	0.493
Go	696	928	474	0.511	0.681	0.584
Mi	38	30	30	1.000	0.789	0.882
MG	63	67	47	0.701	0.746	0.723
Ri	354	305	291	0.954	0.822	0.883
TV	41	28	26	0.929	0.634	0.754

The accuracy of the system is calculated as the total number of correctly classified objects (1094) divided by the total number of objects (1627). This gives an accuracy rate of 0.627.

The high precision for Mitochondria and to a lesser extent for Ribosome and Tubular Vesicles indicates the better visual distinction based around the shape of these objects – Mitochondria are large and tend to be more circular (higher Eccentricity values); Ribosome are small and more irregular in shape while Tubular Vesicles are circular (very high Eccentricity values) and consistently very small in size.

The poor performance in identifying Endoplasmic Reticulum and Golgi objects is possibly due to their low visual distinction when only considering basic shape features. As section 4.1 discussed, they are much more easily described using texture and more intermediate descriptors such as spatial relations.

Neural networks are generally better at discriminating between classes, as is shown in [Perner *et al.*, 2001]. Therefore the difficulty this network has in distinguishing between Golgi and Endoplasmic Reticulum is interesting. However, we feel this is attributable to the lack of spatial relations in the input features.

We didn't achieve the accuracy that we were hoping for since we mapped directly from the low-level features to the domain class label. In addition we did not have a large enough set of input features such as spatial relationship to other objects. This meant that information about spatial relations and intermediate shape and texture descriptors were not incorporated into the classification process.

Also the testing and training data all came from a single example cell. This means that this dataset might not represent adequate statistical variation among the data from different cells. It is not clear that this network would perform as well using data from another cell. When new data is added, the ANN needs to be retrained in order to achieve good performance. Neural nets do not support incremental learning. Other methods such as decision trees or case-based reasoning are preferred for this reason.

Overall, the small sample set and the limited variation in the source object (one cell) restricts the statistical significance of these results and the conclusions that can be made from this evaluation. However, the results from this network provide support for the view that mapping from low-level features to intermediate terms and then from intermediate terms to domain descriptions is likely to be a more successful approach.

6 Conclusion and Future Work

Using artificial neural networks to map from low-level media features directly to high-level semantic terms for image regions does not demonstrate a particularly high level of accuracy. While previous applications have shown that neural networks can be effective in image classification tasks [Colantonio *et al.*, 2006; Di Bona *et al.*, 2003; Perner *et al.*, 2001], we believe that a multi-stage process, as proposed in section 4.2, is likely to be more effective for semantic annotation of image regions. Building semantic annotations by mapping from low-level to visual semantic descriptors and then to domain semantic terms rather than mapping directly from low-level features (for example, as is the case in [Kyrgyzov *et al.*, 2007]) will enable richer and more accurate semantic annotation.

However, until efficient and accurate automatic segmentation algorithms are able to be developed, techniques are needed for semi-automatic semantic annotation that can handle small input data sets, evolving models and rapidly increasing data. Therefore, we aim to develop a system that can operate on an initial, small dataset but incrementally adapt and improve with the addition of further data as it becomes available. The classification system will eventually become more generalized and have improved accuracy as new cell slices are incorporated into the data set. This situation is common in many medical and scientific research fields where the available experimental data may initially be relatively small but which will increase as further experiments and analysis are conducted.

We believe that to generate a semantic description of an image you cannot use the low-level features directly, you have to first map them to intermediate symbolic or semantic terms that make sense for a domain expert. Therefore we intend to focus on a multi-step classification procedure where visual semantic terms (such as 'circular', 'fine speckled margin', 'adjacent' etc.) are created from the automatically extracted low-level features. These terms can then be used to build better classification systems using techniques such as inferencing rules [Little and Hunter, 2004; Hunter and Little, 2005], case-based reasoning [Perner, 2007a; Perner, 2007b] or decision trees [Perner, 2002].

Acknowledgments

This project has been sponsored by the European Commission within the project "Multimedia Understanding through Semantics, Computation, and Learning, MUSCLE" No. 507752.

References

[Adams *et al.*, 2003] B. Adams, G. Iyengar, C. Lin, M. Naphade, C. Neti, H. Nock and J. Smith. "Semantic Indexing of Multimedia Content Using Visual, Audio and Text Cues." In *EURASIP Journal on Applied Signal Processing*, 2003.

[Benitez and Chang, 2003] A. Benitez, A. and S.-F. Chang. "Image classification using multimedia knowledge networks." In *Proceedings of International Conference on Image Processing (ICIP 2003),* 2003, volume 3, pp. III–613–16 vol.2.

[Bloehdorn *et al.*, 2005] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y. Avrithis, S. Handschuh, I. Kompatsiaris, S. Staab, and M. G. Strintzis. "Semantic Annotation of Images and Videos for Multimedia Analysis." In *Proc. 2nd European Semantic Web Conference, ESWC 2005.* Heraklion, Greece, 2005.

[BIRADS] American College of Radiology, Breast Imaging Reporting and Data System (BI-RADS®)

[Chang *et al.*, 1998] S. F. Chang, W. Chen and H. Sundaram. "Semantic Visual Templates: linking visual features to semantics." In *IEEE International Conference on Image Processing (ICIP '98).* Chicago, Illinois, 1998.

[Colantonio *et al.*, 2006] S. Colantonio, I. B. Gurevich and O. Salvetti. "Automatic Fuzzy-Neural based Segmentation of Microscopic Cell Images." *Industrial Conference on Data Mining* - Workshops 2006: 34-45

[Di Bona *et al.*, 2003] S. Di Bona, H. Niemann, G. Pieri and O. Salvetti. "Brain volumes characterisation using hierarchical neural networks." *Artificial Intelligence in Medicine* 28(3): 307-322 (2003)

[Fellbaum, 1998] C. Fellbaum. *Wordnet, An Electronic Lexical Database*. MIT press, 1998.

[Finke, 1989] R. Finke. *Principles of Mental Imagery*, Cambridge MIT Press, 1989, pp89-90

[Garcia and Celma, 2005] R. Garcia and O. Celma. "Semantic integration and retrieval of multimedia metadata." In *Proc. of the 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2005)*, Galway, Ireland, November 2005.

[Holink *et al*., 2003] L. Hollink, A. Schreiber, J. Wielemaker and B. Wielinga. "Semantic Annotation of Image Collections." In *Workshop on Knowledge Capture and Semantic Annotation (KCAP'03)*. Florida, USA, 2003.

[Hollink *et al*., 2005a] L. Hollink, S. Little and J. Hunter. "Evaluating the Application of Semantic Inferencing Rules to Image Annotation." In *Proceedings of the Third International Conference on Knowledge Capture*, KCAP05. Banff, Canada, 2005.

[Hollink *et al*., 2005b] L. Hollink, M. Worring and A. Schreiber. "Building a Visual Ontology for Video Retrieval." In *ACM Multimedia*. Singapore, November 2005.

[Hunter *et al*., 2004] J. Hunter, M. Regan and S. Little. "Position Paper for Semantic Web Life Sciences Workshop – The Visible Cell." *In W3C's Semantic Web Life Sciences Workshop*. Cambridge Mass., 2004.

[Hunter and Little, 2005] J. Hunter and S. Little. "A Framework to enable the Semantic Inferencing and Querying of Multimedia Content" *International Journal of Web Engineering and Technology (IJWET) Special Issue on the Semantic Web*. vol. 2. December 2005.

[Hunter, 2005] J. Hunter, "Adding Multimedia to the Semantic Web - Building and Applying MPEG-7 Ontology." *Multimedia Content and the Semantic Web: Standards, and Tools*, Giorgos Stamou and Stefanos Kollias (Editors), Wiley (2005).

[IBM alphaWorks, 2006] IBM alphaWorks. "Multimedia Analysis and Retrieval Engine (MARVEL)." http://www.alphaworks.ibm.com/tech/marvel, Last accessed: August 2006.

[IMB] Institute for Molecular Biology, "Visible Cell Project", University of Queensland, Australia, http://www.visiblecell.com

[Kyrgyzov *et al*., 2007] I. O. Kyrgyzov, O. O. Kyrgyzov, H. Maitre and M. Campedel. "Kernel MDL to Determin the Number of Clusters", In *Proceedings of the 5th Conference in Machine Learning and Data Mining in Pattern Recognition*, Leipzig, Germany, 2007. pp203-217.

[Little and Hunter, 2004] S. Little and J. Hunter. "Rules-By-Example – a Novel Approach to Semantic Indexing and Querying of Images" *Proceedings of the Third International Semantic Web Conference, ISWC2004*. Hiroshima, Japan. November 2004.

[Manjunath *et al*., 2003] B. S. Manjunath, P Salembier, T. Sikora (Eds). *Introduction to MPEG-7: Multimedia Content Description Interface*, Wiley, 2003.

[Marsh et al., 2001] B. Marsh, D. Mastronarde, K. Buttle, K. Howell, and J. McIntosh. "Organellar relationships in the Golgi region of the pancreatic beta cell line, HIT-T15, visualized by high resolution electron tomography." In *Proceedings of the National Academy of Sciences of the United States of America*, volume 98(5), pp. 2399–2406, 2001.

[MeSH] National Library of Medicine. "Medical Subject Headings (MeSH)." http://www.nlm.nih.gov/mesh/, Last accessed: July 2007.

[Naphade *et al*., 2000] M. Naphade, I. Kozintsev, T. Huang and K. Ramchandran. "A Factor Graph Framework for Semantic Indexing and Retrieval in Video." In *IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, 2000.

[Naphade and Huang, 2001] M. Naphade and T. Huang. "Detecting semantic concepts using context and audiovisual features." In *Proceedings of IEEE Workshop on Detection and Recognition of Events in Video*, 2001, pp. 92–98.

[Natsev *et al*., 2005] A. Natsev, M. Naphade and J. Tesic. "Learning the Semantics of Multimedia Queries and Concepts from a Small Number of Examples." In *ACM Multimedia*, 2005.

[Perner *et al*., 2001] P. Perner, U. Zscherpel and C. Jacobsen. "A comparison between neural networks and decision trees based on data from industrial radiographic testing" *Pattern Recognition Letters*, vol 22, pp47-54, 2001.

[Perner, 2002] P. Perner. *Data Mining on Multimedia Data*, Lecture Notes in Computer Science, Springer, 2002.

[Perner, 2007a] P. Perner (Ed.), *Case-Based Reasoning on Images and Signals*, Springer, 2007 (in print)

[Perner, 2007b] P. Perner. "Prototype-based classification", *Journal of Applied Intelligence*, 2007 (in print)

[Petridis et al., 2006] K. Petridis, D. Anastasopoulos, C. Saathoff, N. Timmermann, I. Kompatsiaris and S. Staab. "M-OntoMat-Annotizer: Image Annotation. Linking Ontologies and Multimedia Low-Level Features", *Engineered Applications of Semantic Web Session (SWEA) at the 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES 2006)*, Bournemouth, U.K., Oct. 2006.

[Smith *et al*., 2003] B. Smith, J. Williams and S. Schulze-Kremer. "The Ontology of the Gene Ontology." In *Proceedings of AMIA Symposium*, 2003.

[Tsinaraki *et al*., 2005] C. Tsinaraki, P. Polydoros, F. Kazasis, and S. Christodoulakis. "Ontology-based Semantic Indexing for MPEG-7 and TV-Anytime Audiovisual Content", *Special issue of Multimedia Tools and Application Journal on Video Segmentation for Semantic Annotation and Transcoding*, Vol. 26, pp. 299-325, 2005.

[van Rijsbergen, 1979] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, second edition edition, 1979.

[Zhao and Grosky, 2002] R. Zhao and W. Grosky. "Negotiating The Semantic Gap: From Feature Maps to Semantic Landscapes." In *Pattern Recognition*, volume 35(3), pp. 51–58, 2002.

Discovering Linguistic Dependencies with Graphical Models

Ernesto William De Luca and **Frank Rügheimer**
University of Magdeburg
Universitätsplatz 2
39106 Magdeburg, Germany
{deluca, ruegheim}@iws.cs.uni-magdeburg.de

Abstract

Graphical models provide a compact approach to analysing and modeling the interaction between attributes. By exploiting marginal and conditional independence relations, high-dimensional distributions are factorized into a set of distributions over lower dimensional subdomains, allowing for a compact representation and efficient reasoning. In this paper, we motivate the choice of linguistic parameters from different language resources as attributes and investigate their interaction. Following that, we extract linguistic dependencies from the structural component of Bayesian Networks induced from data characterized by those attributes. We discuss these preliminary results with respect to their applicability for natural language processing and information retrieval tasks.

1 Introduction

Several types of information sources are available for natural language processing e.g., written and spoken language corpora, lexicons, terminology, etc. [Cole *et al.*, 1997]. These information sources complement one another by describing different, yet related, aspects of language. For this reason a number of successful applications rely on a combination of two or more resources. Examples include word sense disambiguation [Voorhees, 1993], enrichment of resources [Widdows *et al.*, 2002], information retrieval [Littman *et al.*, 1998], and particularly a variety of content-based tasks, such as semantic query expansion [Voorhees, 1994] and conceptual indexing [Gonzalo *et al.*, 1998] for boosting retrieval performance [Vintar *et al.*, 2003]. Whereas some of these methods rely on analysing relational interaction, statistical approaches, such as Full Bayes Classifiers [Duda and Hart, 1973], focus on differences between marginal and conditional probability distributions of attribute values. Information is added by modifying distributions based on the known attribute instantiations. However, because of the size of distributions involved, they are often substituted by Naïve Bayes Classifiers [Good, 1965; Langley *et al.*, 1992; Escudero *et al.*, 2000] thus introducing strong assumptions about attribute independence. In contrast to that, Graphical Models allow a more refined, yet efficient representation of attribute interactions by employing only (conditional) independence relations which are given from background knowledge or observed in training data.

In the following section we recapitulate the fundamentals of Graphical Models with directed and undirected underlying structure graphs. Afterwards (section 3), we discuss several language resources to elaborate how their content can be combined and used in supporting Word Sense Disambiguation tasks. We deal with characteristics frequently found in linguistic data such as large attribute domains or the presence of functional and relational dependencies as well as with properties specific to the corpora serving as data sources. The results of this work are used for selecting suitable parameters (attributes) and compiling training data from those resources.

That data is used in section 4, where we measure attribute interaction and the extract of linguistic dependencies in the form of the structural component of Bayesian Networks. We interpret these preliminary results with respect to their applicability for natural language processing and information retrieval tasks. Section 5 summarizes the work and points to planned supplementary studies.

2 Graphical Models

In principle interaction within a group of attributes is fully described by the joint distribution of its members. However, a direct representation is usually infeasible due to the huge size of the combined domain. Graphical models provide a more compact approach by exploiting marginal and conditional independence relations to factorize high-dimensional distributions. Additionally, efficient reasoning is possible, because the operation on the original domain is reduced to operations on significantly smaller lower-dimensional subspaces and a small overhead for the exchange of information between these local distributions. Underlying independence relations are represented by a graph.

The name *Graphical Models* originates from an analogy between the properties of node separation in graphs and independence relations between attributes. More specifically independence relations are expressed using a graph $G = (X, E)$. The set of nodes in G are corresponds to the set X of considered attributes.

In addition to the structural information provided by the graph G, a graphical model relies a quantitative component in the form of local distributions or factor potentials. Reasoning is performed by conditioning those local distributions and propagating changes along the edges in the graph. In addition to the significantly smaller representation, this provides an efficient method of reasoning. Instead of setting or excluding specific attribute values, an attribute can be instantiated with a probability distribution over the respective attribute domain. That way, uncertain information can be expressed and used in the Graphical Model (see e.g. [Gebhardt *et al.*, 2004]).

If G is a directed acyclic graph (DAG) two attributes A_i and $A_j; i \neq j; A_i, A_j \in X$ are d-separated [Pearl, 1988; Verma and Pearl, 1992] by a set of attributes $S \subseteq X$ in G, this expresses conditional independence of A_i and A_j given the instantiation for the attributes in S. Probabilistic Bayesian Networks are based on the idea that the common probability distribution over a set of attributes can be written as the product of marginal and conditional distributions. Independence relations then allow to simplify the product. Any independence map that is also a DAG provides such a factorization. The factor distributions are the conditional distributions of the attributes A_i in X given their respective set of predecessors $\mathrm{pred}(A_i) = \{A_j \in X \mid (A_j, A_i) \in E\}$ in the graph G. If there are no predecessors the marginal distribution is used instead [Borgelt and Kruse, 2002; Castillo *et al.*, 1997].

$$p_X(t_X) = \prod_{A_i \in X} p_{\{A_i\}|\mathrm{pred}(A_i)}(t_{X|\{A_i\}} \mid t_{X|\mathrm{pred}(A_i)}). \tag{1}$$

If the graph G uses undirected edges to represent the independence structure, it provides the structural component of a Markov Network. For this type of model the clique-graph of G is required to have hypertree structure, which can always be enforced by triangulating G. The distribution is represented as the product of factor potentials assigned to the maximal cliques $C = \{C_1, \ldots, C_k\}$ of G [Lauritzen and Spiegelhalter, 1988].

$$p_X(t_X) = \prod_{C_i \in C} \phi_{C_i}(t_{X|C_i}) \tag{2}$$

In principle, these factor potentials can be computed from the marginal distributions w.r.t. the attributes in each clique, but one must take care, that probability factors for separator sets are considered only once. Their contribution is divided between the cliques sharing these separator sets so several equivalent factor potential assignments my exist for a given distribution and decomposition structure (see [Borgelt and Kruse, 2002] for details on finding factor potentials factors).

The latter representation is closely related to the joint tree propagation method, the main difference being that marginal distributions for separator sets are explicitely represented in the join tree representation, thus allowing for the use of more intuitive marginal distributions instead of factor potentials. Directed graphs can always be converted to undirected graphs with hypertree structure to resolve cyclic dependencies, though at the cost of loosing some independence relations.

3 Information Sources

The goal in building learning models that can support users in a retrieval process, is to provide characteristic information about the query and the corresponding linguistic domain. For instance, when given a query consisting of a single word, we can select concepts based on the linguistic relations of the lexical resource that define its different word senses. Such disambiguating relations are intuitively used by humans. However, if we want to automate this process, we have to use resources – such as probabilistic language models or ontologies – that define appropriate relations.

3.1 Disambiguation of Word Senses

One of the most challenging tasks in information retrieval consists in disambiguating words. As we said before, humans can understand words in context, given their experience, world knowledge and language understanding. For automatic disambiguating of word a variety of association methods (knowledge-driven, data-driven or corpus-based Word Sense Disambiguation) can be used [Ide and Véronis, 1998]. These approaches can be distinguished into supervised and unsupervised approaches.

Supervised approaches normally base on a disambiguated training corpus. It means that there is a training set having word occurrences already annotated with a semantic label (contextually appropriated). The task of such approaches is to build a classifier that can correctly classify new cases (test set) basing on the training set This setting is part of statistical classification [Manning and Schütze, 1999].

Unsupervised approaches are explained as not having/using any information available (e.g. without any training sets or other resources). Strictly speaking, completely unsupervised disambiguation is not possible. If we want to semantically tag (*sense tagging*) some occurrences (and assign them to a sense), some characterization of the possible senses, has to be provided. Otherwise, sense discrimination can be done in a completely unsupervised way (e.g. clustering contexts of an ambiguous word into groups and discriminating them without labeling them).

An important unsupervised learning algorithm for sense disambiguation that can accurately disambiguate word senses in a large, completely untagged corpus has been presented by [Yarowsky, 1995]. He observed and quantified the strong tendency for words to have only one sense in a given collocation (*one sense per collocation* [Yarowsky, 1993]). In other work with Gale and Church [Gale *et al.*, 1992], he describes that words strongly tend to exhibit only one sense in a given discourse or document (*"one sense per discourse"* or *"one sense per document"*). A deeper and complete description of the Yarowsky algorithms for Word Sense Disambiguation is given in [Yarowsky, 1995].

3.2 MultiWordNet

Because we agree with the *one sense per collocation* assumption of [Yarowsky, 1993], recognizing the strong tendency for words to have only one sense in a given collocation, we decided to first enrich the description of every searched word with his meaning and related relevant information (part-of-speech, semantic domain and hyperonyms). In order to access such word-dependent information, we needed to use a lexical resource that could satisfy our requirements.

Fellbaum [Fellbaum, 1998] described WordNet as an electronic lexical database designed based on psycholinguistic and computational theories of the human lexical memory. It provides a list of word senses for each word, organized into synonym sets (SynSets), each representing one constitutional lexicalized concept. Every element of a SynSet is uniquely identified by its SynSet identifier (SynSetID). It is unambiguous and carrier of exactly one meaning. Furthermore, different relations link these elements of synonym sets to semantically related terms (e.g. hyperonyms, hyponyms, etc.). All related terms are also represented as SynSet entries. It also contains descriptions of nouns, verbs, adjectives, and adverbs. It was developed for the English language.

Because of these features, we decided to use the current version of MultiWordNet [Pianta *et al.*, 2002] that is an expansion of (version 1.6) of WordNet contains English,

Italian, and Spanish terms.
The structure of this resource is always organized into synonym sets (SynSets), each representing one constitutional lexicalized concept (like WordNet). But this expansion includes also semantic domain descriptions (called *WordNet Domains*) that are not present in the Princeton version. Semantic domains are considered as a list of related words belonging to the same subject or area of interest and domains in general are common areas of human discussion (e.g. economics, politics, law, science etc.) and are at the basis of lexical coherence [Gliozzo *et al.*, 2004]. Unfortunately set valuedness in the WordNet Domain attribute does not have a consistent interpretation and as it sometimes appears to express alternatives like in the sentence "My usual breakfeast consists of either cereal products or fresh fruit", distributed membership "Sheila's dog is part Collie, part Dalmatian and a little bit of dachshund" or conjunction like in "Fred holds both Argentine and Italian Citizenship". In the first two cases probabilistic modeling is still feasible but a distribution or degrees of membership would have to be provided. In the latter case probabilistic modeling is not sufficient to resolve set-valuedness. For that reason we decided to treat such set-valued specifications as separate values of the domian attribute.

3.3 MultiSemCor

As we already said before, we wanted to learn the joint distribution of the attributes. This can be done using corpora to be then represented by graphical models. For this reason, we decided to also use linguistic information contained in SemCor (annotated with WordNet SynSetIDs) and MultiSemCor (annotated with MultiWordNet SynSetIDs).

SemCor [Miller *et al.*, 1993] is the best-known publicly available corpus hand-tagged with WordNet senses. It is a manually sense tagged subset of the Brown Corpus consisting of 352 Documents (with more than 200,000 content words) subdivided into three data sets, having for every lemma a unique sense identifier (SynSetID). The collection is rather heterogeneous, covering different topics like politics, sports, music, cinema, philosophy, parts of fiction novels, scientific texts. Because one of our aims in the future is also to support multilingual retrieval, we decided to use also MultiSemCor. This is an English/Italian parallel corpus, aligned at the word level and annotated with POS, lemma and word sense. For creating it, they first exploited the SemCor corpus, in which content words are lemmatized and annotated with WordNet senses. After this, they created the appropriate parallel translation, done from human translators. At present MultiSemCor is composed by 116 English/Italian aligned parallel texts [Bentivogli *et al.*, 2005]. The authors explain their approach based on the assumption that if a text in one language has been annotated and its translation has not, annotations can be transferred from the source text to the target using word alignment as a bridge [Bentivogli *et al.*, 2004].

We used the MultiSemCor data (the English data) as a training data.

3.4 Related Work

Testing disambiguators is a very difficult task, because there are few 'pre-disambiguated' test corpora publicly available. Researcher are often confronted with the time consuming task of manual disambiguation of the words. Each project (e.g. WordNet or LDOCE) has often different definitions of word sense [Sanderson, 2000]. Retrieval based on WordNet SynSets was found to produce consistently high effectiveness results (62% of known items retrieved at rank one) better than words (48%) or senses alone (53.2%).
But in contrast to the work based on a predefined set of word sense definitions, there has been another disambiguation approach for IR: the creation of disambiguators based on corpora. [Mihalcea and Moldovan, 2000] combined a word-based and sense-based approach with indexing and retrieval components. They built a semantic representation of open text, at word and collocation level. They call their technique "semantic indexing", showing that effectiveness is improved over the classic word based indexing techniques. The word context is the base of this disambiguation process. Here the meaning of the words is identified basing on the WordNet senses. The lexical and semantic tags (retrieved from WordNet) are added to the words and the documents are ready to be indexed. The index creation is based on words as lexical strings (for the word-based retrieval), and on semantic tags (for the sense-based retrieval). They performed tests on 6 randomly selected files from SemCor, to evaluate the accuracy and the recall of the disambiguation method. Only 55% of the words could be disambiguated so far.

3.5 Choosing Linguistic Parameters

Because our first idea was to do multilingual text retrieval [De Luca and Nürnberger, 2006], we looked for multilingual parallel corpora annotated with multilingual resources. We licensed the *EUROVOC Thesaurus* [Steinberger and Pouliquen, 2003], to use it in combination with the Multilingual Parallel Corpus *JRC-Acquis* [Steinberger *et al.*, 2006], annotated with its categories.
Although we did not yet obtain access to the thesaurus, we included other information sources that are directly available, delaying the use of *JRC-Acquis* for future work.
Another important resource available to researchers for this purpose is the mentioned WordNet [Miller *et al.*, 1990] and its variations like MultiWordNet [Pianta *et al.*, 2002] and EuroWordNet [Vossen, 1999]. The SemCor Corpus (and its variant MultiSemCor), is annotated with WordNet SynSetIDs and can be used in a similar way as the EUROVOC and Acquis resources, though different features are provided.
The data we used for network induction had to be preprocessed in order to extract the most relevant attributes. Different information could have been taken into account. The MultiSemCor data set is subdivided in paragraph, sentence and token level. Whereas this structure can be used to determine collocations, annotations are only given on the token level so all of the SemCor attributes used by us refer to this last level. The attributes taken into account are: token, POS, lemma and lexsn of every word contained in the whole collection. It means that for every annotated word, we extracted the word itself (token), the correspondent Part-Of-Speech (POS), the canonical form of the word (lemma) and the lexsn (unique identifier containing information about the sense and the annotator).

Because we also wanted to retrieve additional information about the word (token), we needed to obtain the related word-dependent WordNet SynSetID. Using the pattern "lemma%lexsn" we constructed a key using these two attributes already contained in the corpus. As WordNet provides a mapping index from that key to its internal SynSetID we were able to recover the values for the

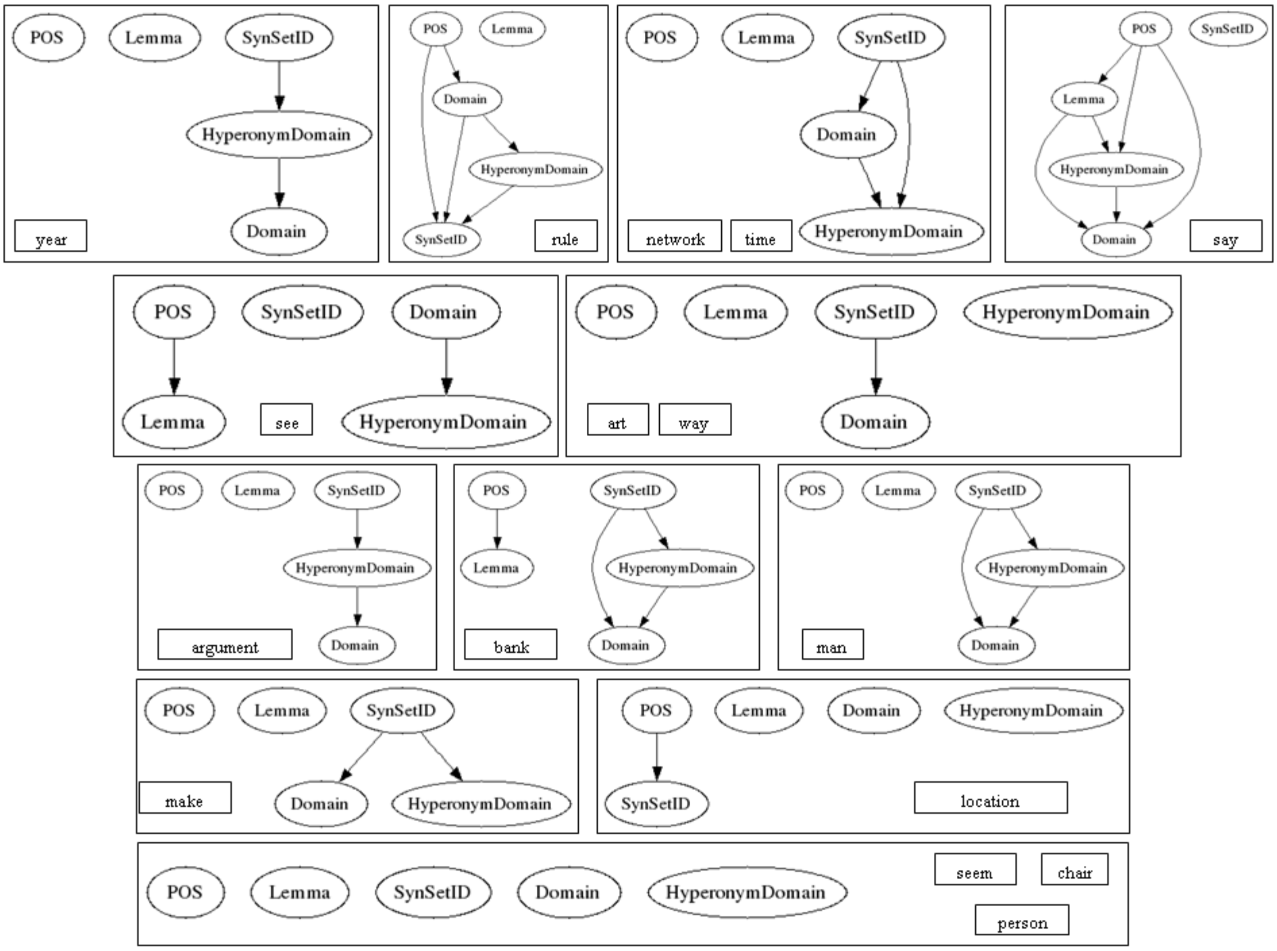


Figure 1: Network structures obtained for selected lemmas

SynSetID attribute. With this identifier, we retrieved the WordNet domain, hyperonyms and hyperonym domains of every token.

Since we wanted to use semantic domains for learning, we had to take care about the word distributions. Several meanings are assigned to the domain "Factotum" that could be described as the class "other domain, generic". This assignment is explained from the restriction that the WordNet authors have to assign a domain to each SynSet.
Therefore, if we want to use this information, we have to choose which senses are relevant and which are not.. However, if we maintain all senses that are labeled with "Factotum", in many cases, we have to distinguish between only slightly different contexts defined by different SynSets. Because this domain distorts our distribution and does not give any additional information about the token or about the lemma, we decided include the frequency of SynSets with the "Factotum" domain in our discussion. Our original dataset contained 110888 tuples. Because missing values are problematic for subsequent processing, tuples that lacked instantiations for any of the considered attributes were removed in a cleaning step. The resulting dataset consisted of 65774 Tuples with a total of 16267 different value combinations.

4 Experimental Settings and Evaluation

In our experiments we investigated the statistical relations between the Part-Of-Speech (POS), Lemma, SynSetID features as well as the assumed semantic domain of the considered word and of its hyperonym (see also section 3.2). We extracted the directed graphical model using the conditional independence test approach [Spirtes *et al.*, 1993] with symmetric information gain ratio. The INeS toolset (http://borgelt.net/ines.html) was employed for learning Graphical Models based on directed graphs. A detailed description of the INeS toolset and network training methods is found in [Borgelt and Kruse, 2002].

In a preparation step we measured pairwise relational attribute interaction in the data set using Hartley information gain. The reasoning behind this procedure was that strong relational or even functional dependence between linguistic attributes frequently expresses a-priori background knowledge, but may mask genuinely interesting probabilistic interaction patterns.

The Hartley information [Hartley, 1928] of a single attribute is defined as the binary logarithm of the cardinality of that attributes domain. An attribute with a larger domain is considered to be more informative than an attribute with a smaller domain, because instantiating it excludes more alternative values. Hartley information gain compares the number of attribute value combinations actually observed to the total size of the joint domain (i.e. the expected number of combinations if the attribute values were freely combined) and thus measures the amount of information gained considering the attribute interaction instead of treating them as independent. It is computed according to the following formula:

$$\begin{aligned} I_{gain}^{(Hartley)}(A,B) &= \log_2(\text{valsobs}(\{A\})) \\ &+ \log_2(\text{valsobs}(\{B\})) \\ &- \log_2(\text{valsobs}(\{A,B\}), \quad (3) \end{aligned}$$

where $\text{valsobs}(X)$ denotes the number of observed values in the combined domain of the attributes in X. Note that the quality of the results depends on the size of the distribution over the combined attribute domains. Specifically the sample size may be insufficient if the joint domain is too large. Nevertheless it allows a good detection of strong relational interactions. which are taken into consideration when discussing the results of the probabilistic network training.

Tables 1 and 2 show the functional dependence of the attributes 'Domain' and 'HyperonymDomain' w.r.t. 'SynSetID'. This result was expected, because both attributes have been obtained from WordNet using queries based on 'SynSetID'. Moreover we observed a comparatively low number of combinations for {POS, Lemma} and {POS, SynSetID} which reveal a relational dependency. We expected this result as well, because many words and meanings may only appear with a limited set of functions in the sentence. The number of combinations for {Lemma, SynSetID} is very low, even if the limited size of the sample is taken into account (also seen in the high value of the measure in Table 4). A very strong connection between the attributes may at first appear unusual given difficulties of word sense disambiguation in applications. But the observation is in line with the "one sense per document" hypothesis. The connection is understood if one considers that the effective sample size is reduced to the number of documents if all occurrences in the same document are assigned identical SynSetIDs, thus explaining the low number of value combinations.

Due to the strong relational dependencies and large attribute domains for the 'Lemma' and 'SynSetID' attributes the probability distribution over the joint variable domain is very large, 0 for almost every input tuple. Even if only projections are considered the presence of either attribute would have significantly reduced the effectiveness of a probabilistic approach based on marginal distributions. Instead we opted for considering conditional distributions given attributes which exhibit a strong relational interaction with either 'Lemma' or 'SynSetID'. This approach effectively eliminated the conditioning attribute and reduced the number of lemmas and SynSets to be considered at a given time leading to much smaller and managable distributions.

For the first run of experiments we subsequently fixed the value of the 'lemma' attribute to its 10 most frequent values. An additional six lemmas were selected for their high ambiguity. From the statistical point of view we considered estimated conditional probabilities of value combinations given the selected values for 'Lemma'. The obtained network structures are shown in Figure 1.

attribute	POS	Lemma	SynSet-ID	Domain	Hypero-nym-Domain
#values	7	10318	13106	532	330
$\log_2$	2.80	13.33	13.68	9.06	8.37

Table 1: Number of distinct attribute values and its binary logarithm for each attribute

attribute pair	Lemma	SynSetID	Domain	Hypero-nym-Domain
POS	11107	13119	644	406
Lemma		16273	12833	12493
SynSetID			13106	13106
Domain				1265

Table 2: Number of distinct observed values from the combined domain for each attribute pair

attribute pair	Lemma	SynSetID	Domain	Hypero-nym-Domain
POS	2.7	2.81	2.53	2.51
Lemma		13.02	8.74	8.09
SynSetID			4.43	3.74
Domain				6.43

Table 3: Hartley information gain computed for each attribute pair

The graph for "seem","chair" and "person" demonstrate marginal and conditional independence of all attributes. This result is explained with the observation that occurrences either exhibit the same combination of attribute values, or only vary in one of the attributes, so no attribute interaction is possible. The latter is also the case for "say", though the high number of values for SynSetID leads the network training algorithm to retain edges between the remaining attributes. Nevertheless no conditional distributions would have to be stored in the associated graphical model, because the respective attribute domains are singletons. Among the tuples referring to "location" only three different combinations exist, two of them very rare compared to the dominant one. All semantic domains are given as "Factotum" and therefore useless to predict other attributes. The edge from POS to Lemma is reflected by the data, but may well be an artifact from the sampling process. In general we observed stronger interactions within the subset of SynSetID, Domain, HyperonymDomain attribute. Only in one case ("rule") the POS attribute was linked to the SynSetID distinguishing between two different meanings.

In a second batch of experiments we employed the same approach, but fixed the Domain attribute to collapse the domains for both the SynSetID and Lemma attribute. Again the 10 most frequent attribute values were used. Notably the most frequent value "Factotum" alone made up for 32105 tuples (48.8% of the cleaned database).

As with the previous experiment the network structures are consistent with low interaction between POS and the remaining attributes. The majority of the induced networks (7) considered POS as an independent attribute though in 6 of these cases POS had more than one value. In one of the three remaining cases (Domain="Biology Per-

attribute pair	Lemma	SynSetID	Domain	Hypero-nym-Domain
POS	0.2	0.21	0.27	0.29
Lemma		0.93	0.64	0.59
SynSetID			0.32	0.27
Domain				0.62

Table 4: Symmetric Hartley information gain ratio computed for each attribute pair

son") only five different value combinations occurred. And the resulting structure (a directed chain) is not representative of the full distribution. For Domain="Economy" the POS attribute contributed to the distinction of the between senses of "pay" and a similar observation could be made for the "Psychology Domain". Most networks considered full interaction between the Lemma, SynSetID and HyperonymDomain. Two networks where HyperonymDomain appeared as an independent attribute can be explained by the attribute domain either being a singleton or being dominated by a single value thus justifying an approximate independence assumption.

5 Summary

We investigated the interaction of attributes from different linguistic information sources and motivated the choice of parameters from different language resources. Following that, we extracted linguistic dependencies in the form of the structural component of Bayesian Networks from data characterized by those attributes. We discussed these preliminary results with respect to their applicability for natural language processing and information retrieval tasks. We combined syntactic (POS) and semantic information (semantic domains extracted from WordNet) for model construction. In our experiments, we found indication for the necessity to employ more than one information source, because the two classes of features did not strongly interact. Only in few cases the POS attribute could be used for Word Sense Disambiguation though the discriminate power appeared to be high if prediction via POS was applicable. However the size of the considered data set and the number of features available at the given time suggest further experiments with extended data. The results of our analysis are consistent with the "one sense per document" assumption. A better test of that assumption could be performed by using richer data sets (e.g. the EUROVOC thesaurus and Acquis Corpus), providing features from other sources such as document categories or information provided at paragraph, sentence and document level.

References

[Bentivogli *et al.*, 2004] Luisa Bentivogli, Pamela Forner, and Emanuele Pianta. Evaluating cross-language annotation transfer in the multisemcor corpus. In *Proceedings of COLING 2004*, pages 364–370, Geneva, Switzerland, 2004.

[Bentivogli *et al.*, 2005] Luisa Bentivogli, Emanuele Pianta, and Marcello Ranieri. Multisemcor: an english italian aligned corpus with a shared inventory of senses. In *Proceedings of the Meaning Workshop 2005*, Trento, Italy, 2005.

[Borgelt and Kruse, 2002] Christian Borgelt and Rudolf Kruse. *Graphical Models—Methods for Data Analysis and Mining*. J. Wiley & Sons, Chichester, 2002.

[Castillo *et al.*, 1997] E. Castillo, J. M. Guitérrez, and A. S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer-Verlag, New York, 1997.

[Cole *et al.*, 1997] R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue. Survey of the state of the art in human language technology, 1997.

[De Luca and Nürnberger, 2006] Ernesto William De Luca and Andreas Nürnberger. A word sense-oriented user interface for interactive multilingual text retrieval. In *Proceedings of the Workshop Information Retrieval In conjunction with the LWA 2006, GI joint workshop event 'Learning, Knowledge and Adaptivity'*, Hildesheim, Germany, 2006.

[Duda and Hart, 1973] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. J. Wiley & Sons, New York, NY, USA, 1973.

[Escudero *et al.*, 2000] G. Escudero, L. Arquez, and G. Rigau. Naive bayes and exemplar-based approaches to word sense disambiguation revisited. In *Proceedings of the 14th European Conference on Artificial Intelligence, ECAI*, Berlin, Germany, 2000.

[Fellbaum, 1998] Christiane Fellbaum. *WordNet, an electronic lexical database.* MIT Press, 1998.

[Gale *et al.*, 1992] William A. Gale, Kenneth W. Church, and David Yarowsky. One sense per discourse. In *Proceedings of the 4th DARPA Speech and Natrual Language Workshop*, pages 233–237, 1992.

[Gebhardt *et al.*, 2004] Jörg Gebhardt, Frank Rügheimer, Heinz Detmer, and Rudolf Kruse. Adaptable markov models in industrial planning. In *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems (Budapest)*, Piscataway, NJ, USA, 2004. IEEE Press.

[Gliozzo *et al.*, 2004] Alfio Gliozzo, Carlo Strapparava, and Ido Dagan. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech and Language*, 18(3):275–299, 2004.

[Gonzalo *et al.*, 1998] Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan Cigarran. Indexing with wordnet synsets can improve text retrieval. In *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP*, pages 38–44, Montreal, Canada, 1998.

[Good, 1965] Irving John Good. *The Estimation of Probabilities: An Essay on Modern Byesian Methods*. MIT Press, Cambridge, Mass., USA, 1965.

[Hartley, 1928] R.V.L. Hartley. Transmission of information. *The Bell System Technical Journal*, 7:535–563, 1928.

[Ide and Véronis, 1998] Nancy Ide and Jean Véronis. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24:1:1–40, 1998.

[Langley *et al.*, 1992] Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI'92, San Jose, CA, USA)*, pages 223–228, Menlo Park and Cambridge, USA, 1992. MIT Press.

[Lauritzen and Spiegelhalter, 1988] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 2(50):157–224, 1988.

[Littman *et al.*, 1998] M. Littman, S. Dumais, and T. Landauer. Automatic cross-language information retrieval using latent semantic indexing, 1998.

[Manning and Schütze, 1999] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Boston, USA, 1999.

[Mihalcea and Moldovan, 2000] Rada Mihalcea and Dan Moldovan. Semantic indexing using wordnet senses, 2000.

[Miller *et al.*, 1990] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Five papers on wordnet. *International Journal of Lexicology*, 3(4), 1990.

[Miller *et al.*, 1993] George Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. A semantic concordance. In *Proceedings of DARPA Speech and Natural Language Workshop*, 1993.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Francisco, CA, USA, 1988.

[Pianta *et al.*, 2002] Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. Multiwordnet: developing an aligned multilingual database. In *First International Conference on Global WordNet*, Mysore, India, 2002.

[Sanderson, 2000] Mark Sanderson. Retrieving with good sense. *Information Retrieval*, 2(1):49–69, 2000.

[Spirtes *et al.*, 1993] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Number 81 in Lecture Notes in Statistics. Springer Verlag, New York, NY, USA, 1993.

[Steinberger and Pouliquen, 2003] Ralf Steinberger and Bruno Pouliquen. Automating the assignment of eurovoc descriptors to text. In *Eurovoc Conference 2003*, Brussels, Belgium, 2003.

[Steinberger *et al.*, 2006] Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufis, and Dániel Varga. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, 2006.

[Verma and Pearl, 1992] T. Verma and J. Pearl. An algorithm for deciding whether a set of observed independencies has a causual explanation. In *Proc. of the 8th UAI Conference*, pages 323–330, 1992.

[Vintar *et al.*, 2003] S. Vintar, P. Buitelaar, and M. Volk. Semantic relations in concept-based cross-language medical information retrieval. In *Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining*, Croatia, 2003.

[Voorhees, 1993] E. Voorhees. Using wordnet to disambiguate word sense for text retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–180, 1993.

[Voorhees, 1994] E. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61–69, 1994.

[Vossen, 1999] Piek Vossen. Eurowordnet general document, version 3, final. In *www.illc.uva.nl/EuroWordNet/docs/GeneralDocPS.zip*, 1999.

[Widdows *et al.*, 2002] Dominic Widdows, Beate Dorow, and Chiu-Ki Chan. Using parallel corpora to enrich multilingual lexical resources. In *Third International Conference on Language Resources and Evaluation*, pages 240–245, Las Palmas, Spain, May 2002.

[Yarowsky, 1993] David Yarowsky. One sense per collocation. In *Proceedings of ARPA Human Language Technology Workshop*, 1993.

[Yarowsky, 1995] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

An Evaluation of Efficient Multilabel Classification Algorithms for Large-Scale Problems in the Legal Domain

Eneldo Loza Mencía and Johannes Fürnkranz
Knowledge Engineering Group
Technische Universität Darmstadt
[loza,juffi]@ke.informatik.tu-darmstadt.de

Abstract

In this paper we evaluate the performance of multilabel classification algorithms on two classification tasks related to documents of the EUR-Lex database of legal documents of the European Union. It permits different settings of large-scale multilabel problems with up to 4000 classes with the same underlying documents. We compared the well known one-against-all approach (OAA) and its recently proposed improvement, the multiclass multilabel perceptron algorithm (MMP), which modifies the OAA ensemble by respecting dependencies between the base classifiers in the training protocol of the classifier ensemble. Both use the simple but very efficient perceptron algorithm as underlying classifier. This makes them very suitable for large-scale multilabel classification problems, in particular when the number of classes is high. Our results on the EUR-Lex database confirm that the MMP algorithm has a better response to an increasing number of classes than the one-against-all approach. We also show that it is principally possible to efficiently and effectively handle very large multilabel problems.

1 Introduction

Recently, *multilabel classification* problems, where the task is to associate an object with an unrestricted set of classes instead of exactly one, have received increased attention in the literature. With the increased attention in recent times in this type of setting, new algorithms have been developed or adapted to automatically solve the task of multilabel classification. But simultaneously an increased number of new scenarios have been identified and higher demands are continuously made to the existing algorithms. This concerns not only challenges due to large scale instance spaces, large numbers of instances and numbers of features, but particularly due to the number of possible classes.

In particular in text classification, these type of problems are very common. The number of possible categories that can typically be assigned to each document varies from a few dozen to several hundred. In this paper, we study a challenging new domain, namely assigning documents of the EUR-Lex database to a few of $\approx 4,000$ possible labels. The EUR-Lex database is a freely accessible document management system for legal documents of the European Union. We chose this database for several reasons:

- it contains multiple classifications of the same documents, making it possible to analyze the effects of different classification properties using the same underlying reference data without resorting to artificial or manipulated classifications,
- the overwhelming number of produced documents make the legal domain a very attractive field for employing supportive automated solutions and therefore a machine learning scenario in step with actual practice,
- and the data is freely accessible.

The simplest strategy to tackle the multilabel problem with existing techniques is to use the *one-against-all* binarization, in the multilabel setting also referred to as the *binary relevance* method. It decomposes the original problem into less complex, binary problems, by learning one classifier for each class, using all objects of this class as positive examples and all other objects as negative examples. At query time, each binary classifier predicts whether its class is relevant for the query example or not, resulting in a set of relevant labels. While this technique can potentially be used to transform any binary classifier into a multilabel classifier and it is often used in practical applications, the question remains, whether this general approach can fully adapt to the particular needs of multilabel classification, because it trains each class independently of all other classes.

A recently proposed alternative that tries to tackle this problem is the *multilabel multiclass perceptron algorithm (MMP)* developed by Crammer and Singer [2003], which adapts the one-against-all approach to the multilabel case. Instead of learning the relevance of each class individually and independently, MMP incrementally trains the entire classifier ensemble as a whole so that it predicts a real-valued relevance value for each class. This is done by always evaluating the performance of the entire ensemble, and only producing training examples for the individual classifiers when their corresponding classes are misplaced in the ranking. It uses perceptrons as base classifiers, because they are simple and efficient, and because for high-dimensional problems like text classification, linear discriminants are sufficiently expressive.

A shortcoming of the proposed method is that the resulting prediction is not any more a set of classes as expected for a multilabel task, but a ranking of class relevance scores. However, it is possible to obtain the desired output in an additional step that selects classes which exceed a determined relevance value. Different methods exist for determining the threshold, a good overview can be found in Sebastiani [2002]. Recently, Brinker et al. [2006] introduced the idea of using an artificial label that encodes the boundary between relevant and irrelevant labels for each example.

In this paper, we will concentrate on the topic ranking task, which also enables a more detailed evaluation of the classification performance.

The MMP algorithm has already been used on large scale data sets such as the Reuters Corpus Volume 1 with its over 800,000 examples and approx. 100 classes [Crammer and Singer, 2003]. In the experiments the MMP algorithm was able to substantially improve the performance of the one-against-all method with perceptrons as base classifiers. In this paper we will analyze if these results can be repeated on two different settings of the EUR-Lex database, one with approx. 200 and the other with 4000 possible classes. Note that the latter problem has one order of magnitude more classes than other known applications of this algorithm.

2 Preliminaries

We represent an instance or object as a vector $\bar{\mathrm{x}} = (x_1, \ldots, x_N)$ in a feature space $\mathcal{X} \subseteq \mathbb{R}^N$. Each instance $\bar{\mathrm{x}}_i$ is assigned to a set of relevant labels y_i, a subset of the K possible classes $\mathcal{Y} = \{c_1, \ldots, c_K\}$. For multilabel problems, the cardinality $|y_i|$ of the label sets is not restricted, whereas for binary problems $|y_i| = 1$. For the sake of simplicity we use the following notation for the binary case: we define $\mathcal{Y} = \{1, -1\}$ as the set of classes so that each object $\bar{\mathrm{x}}_i$ is assigned to a $y_i \in \{1, -1\}$, $y_i = \{y_i\}$.

2.1 Ranking Loss Functions

In order to evaluate the predicted ranking we use different *ranking losses*. The losses are computed comparing the ranking with the true set of relevant classes, each of them focusing on different aspects. For a given instance $\bar{\mathrm{x}}$, a relevant label set y, a negative label set $\overline{y} = \mathcal{Y} \backslash y$ and a given predicted ranking $r(\bar{\mathrm{x}})$ the different loss functions are computed as follows:

ISERR The is-error loss determines whether $r(c) < r(c')$ for all relevant classes $c \in y$ and all irrelevant classes $c' \in \overline{y}$. It returns 0 for a completely correct, *perfect ranking*, and 1 for an incorrect ranking, irrespective of 'how wrong' the ranking is.

ONEERR The one error loss is 1 if the top class in the ranking is not a relevant class, otherwise 0 if the top class is relevant, independently of the positions of the remaining relevant classes.

ERRSETSIZE The error set size loss returns the number of pairs of labels which are not correctly ordered. Like ISERR, it is 0 for a perfect ranking, but it additionally differentiates between different degrees of errors.

$$E \stackrel{\text{def}}{=} \{(c, c') \mid r(c) > r(c')\} \subseteq y \times \overline{y} \tag{1}$$

$$\delta_{\text{ErrSetSize}} \stackrel{\text{def}}{=} |E| \tag{2}$$

MARGIN The margin loss returns the number of positions between the worst ranked positive and the best ranked negative classes. This is directly related to the number of wrongly ranked classes, i.e. the positive classes that are ordered below a negative class, or vice versa. We denote this set by F.

$$\begin{aligned} F \stackrel{\text{def}}{=} &\{c \in y \mid r(c) > r(c'), c' \in \overline{y}\} \\ &\cup \{c' \in \overline{y} \mid r(c) > r(c'), c \in y\} \end{aligned} \tag{3}$$

$$\begin{aligned} \delta_{\text{Margin}} \stackrel{\text{def}}{=} &\max(0, \max\{r(c) \mid c \in y\} \\ &- \min\{r(c') \mid c' \notin y\}) \end{aligned} \tag{4}$$

AVGP Average Precision is commonly used in Information Retrieval and computes for each relevant label the percentage of relevant labels among all labels that are ranked before it, and averages these percentages over all relevant labels. In order to bring this loss in line with the others so that an optimal ranking is 0, we revert the measure.

$$\delta_{\text{AvgP}} \stackrel{\text{def}}{=} 1 - \frac{1}{y} \sum_{c \in y} \frac{|\{c^* \in y \mid r(c^*) \leq r(c)\}|}{r(c)} \tag{5}$$

2.2 Perceptrons

A perceptron is a binary classifier initially developed as a model of the biological neuron [Rosenblatt, 1958]. Internally, it computes a linear combination of a real-valued input vector and predicts the positive class if the result is positive, and the negative class otherwise. More precisely, given an input vector $\bar{\mathrm{x}}$, the predicted class of a perceptron is computed as

$$o'(\bar{\mathrm{x}}) = sgn(\bar{\mathrm{x}} \cdot \bar{\mathrm{w}} + \omega) \tag{6}$$

with the weight vector $\bar{\mathrm{w}}$, threshold ω and $sgn(t) = 1$ for $t \geq 0$ and -1 otherwise. We can interpret a perceptron as a hyperplane with the formula $\bar{\mathrm{x}} \cdot \bar{\mathrm{w}} = -\omega$ that divides the N-dimensional space into two halves. An instance is a point in this space and its position determines its class membership. If the two sets of positive and negative points, respectively, can be separated by a hyperplane, they are called *linearly separable*. As a consequence, irrespective of the training algorithm used, linear classifiers like the perceptron cannot arrive at correct predictions for all potential instances unless the negative and positive instances are linearly separable. In order to find a possibly existing *separating hyperplane*, the weights are adapted according to the following perceptron training rule:

$$\begin{aligned} \theta_i &= (y_i - o'(\bar{\mathrm{x}}_i)) \\ \bar{\mathrm{w}}_{i+1} &= \bar{\mathrm{w}}_i + \eta\theta_i\bar{\mathrm{x}}_i \\ \omega_{i+1} &= \omega_i + \eta\theta_i\delta \end{aligned} \tag{7}$$

with δ usually being set to 1 and the initial weights set to zero without loss of generality. The learning rate η can be ignored if set to be constant [Bishop, 1995], as it will be the case in this work. When a N-dimensional point is misclassified, the hyperplane is moved towards this point (indicated by θ). If the training examples can be seen iteratively and the data is linearly separable, the algorithm provably finds a dividing hyperplane. This is called the perceptron convergence criterion (see, e.g., [Bishop, 1995]). Irrespective of training until convergence not always being desirable, this property does not reveal anything about the performance on unseen data.

Note that the number of errors until convergence depends on the margin between the positive and negative points. The hyperplane that maximizes the margin to the closest positive and negative point is called the *optimal hyperplane*. Contrary to support vector machines, perceptrons will not necessarily find an optimal hyperplane. However, the size of the margin is an indicator for the hardness of the learning problem: the smaller the margin the harder it is for the perceptron algorithm to find a good solution. On the other hand, perceptrons can be trained efficiently in an incremental setting, which makes them particularly well-suited for large-scale classification problems such as the Reuters 2000 (RCV1) benchmark [Lewis et al., 2004]. For this reason, the perceptron has recently received increased attention [Freund and Schapire, 1999, Li et al., 2002, Shalev-Shwartz and Singer, 2005, Dekel et al., 2005].

Certainly, the δ value becomes important when the perceptron is trained in only one epoch: it is easily shown that $|\bar{w}| \leq |\mathcal{W}| \cdot \max_{\bar{x} \in \mathcal{W}} |\bar{x}|$ and $|\omega| \leq |\mathcal{W}| \cdot \delta$ holds for misclassified training examples $\mathcal{W} = \{\bar{x} \mid o'(\bar{x}) \neq \mathcal{y}\}$. A disproportion between $\max_{\bar{x}} |\bar{x}|$ and δ can obviously lead to an excessive predominance of the threshold and make the scalar product even superfluous. To circumvent the problem of determining the right value for δ, we can set it to zero sacrificing one dimension in the hypothesis space (thus ω=0). Graphically this means that only separating hyperplanes through the origin are considered, reducing the number of potentially solvable problems. In practice, especially in high dimensional spaces as for text documents, this is usually not a very significant restriction, and it additionally renders online learning possible.

2.3 Binary Relevance Ranking

In the binary relevance or one-against-all (OAA) method, a multilabel training set with K possible classes is decomposed into K binary training sets of the same size that are then used to train K binary classifiers. So for each pair ($\bar{x}_i$,$\mathcal{y}_i$) in the original training set K different pairs of instances and binary class assignments $(\bar{x}_i, y_{i_j})$ with $j = 1 \ldots K$ are generated as follows:

$$y_{i_j} = \begin{cases} 1 & c_j \in \mathcal{y}_i \\ -1 & otherwise \end{cases} \tag{8}$$

Supposing we use perceptrons as base learners, K different o'_j classifier are trained in order to determine the relevance of c_j. In consequence, the combined prediction of the one-against-all classifier for an instance $\bar{x}$ would be the set $\{c_j \mid o'_j(\bar{x}) = 1\}$. If, in contrast, we want to obtain a ranking of classes according to their relevance, we can simply use the result of the internal computation of the perceptrons as a measure of relevance. According to Equation 6 the desired linear combination is the inner product $o_j(\bar{x}) = \bar{x} \cdot \bar{w}_j$ (ignoring ω as mentioned above). So the result of the prediction is a vector $\bar{o}(\bar{x}) = (\bar{x}\bar{w}_1, \ldots, \bar{x}\bar{w}_K)$ where component j corresponds to the relevance of class c_j. We will denote the ranking function that returns the position of class c in the ranking with $r(c) \in \{1 \ldots K\}$. Ties are broken randomly to not favor any particular class.

2.4 Multiclass Multilabel Perceptrons

MMPs were proposed as an extension of the one-against-all algorithm with perceptrons as base learners [Crammer and Singer, 2003]. Just as in one-against-all, one perceptron is trained for each class, and the prediction is calculated via the inner products. The difference lies in the update method: while in the one-against-all method all perceptrons are trained independently to return a value greater or smaller than zero, depending on the relevance of the classes for a certain instance, MMPs are trained to produce a good ranking so that the relevant classes are all ranked above the irrelevant classes. The perceptrons therefore cannot be trained independently, considering that the target value for each perceptron depends strongly on the values returned by the other perceptrons.

```
Require: Training example pair $(\bar{x}, \mathcal{y})$, perceptrons $\bar{w}_1, \ldots, \bar{w}_K$
calculate $\bar{x}\bar{w}_1, \ldots, \bar{x}\bar{w}_K$, loss $\delta$
if $\delta > 0$ then                                  ▷ only if ranking is not perfect
    calculate error sets $E$, $F$
    for each $c \in F$ do $\tau_c \leftarrow 0$                    ▷ initialize $\tau$'s
    for each $(c, c') \in E$ do
        $p \leftarrow$ PENALTY$(\bar{x}\bar{w}_1, \ldots, \bar{x}\bar{w}_K)$
        $\tau_c \leftarrow \tau_c + p$                            ▷ push up pos. classes
        $\tau_{c'} \leftarrow \tau_{c'} - p$                      ▷ push down neg. classes
        $\sigma \leftarrow \sigma + p$                            ▷ for normalization
    normalize $\tau$'s
    for each $c \in F$ do
        $\bar{w}_c \leftarrow \bar{w}_c + \delta \frac{\tau_c}{\sigma} \cdot \bar{x}$             ▷ update perceptrons
return $\bar{w}_1 \ldots \bar{w}_K$                        ▷ return updated perceptrons
```

Figure 1: Pseudocode of the training method of the MMP algorithm

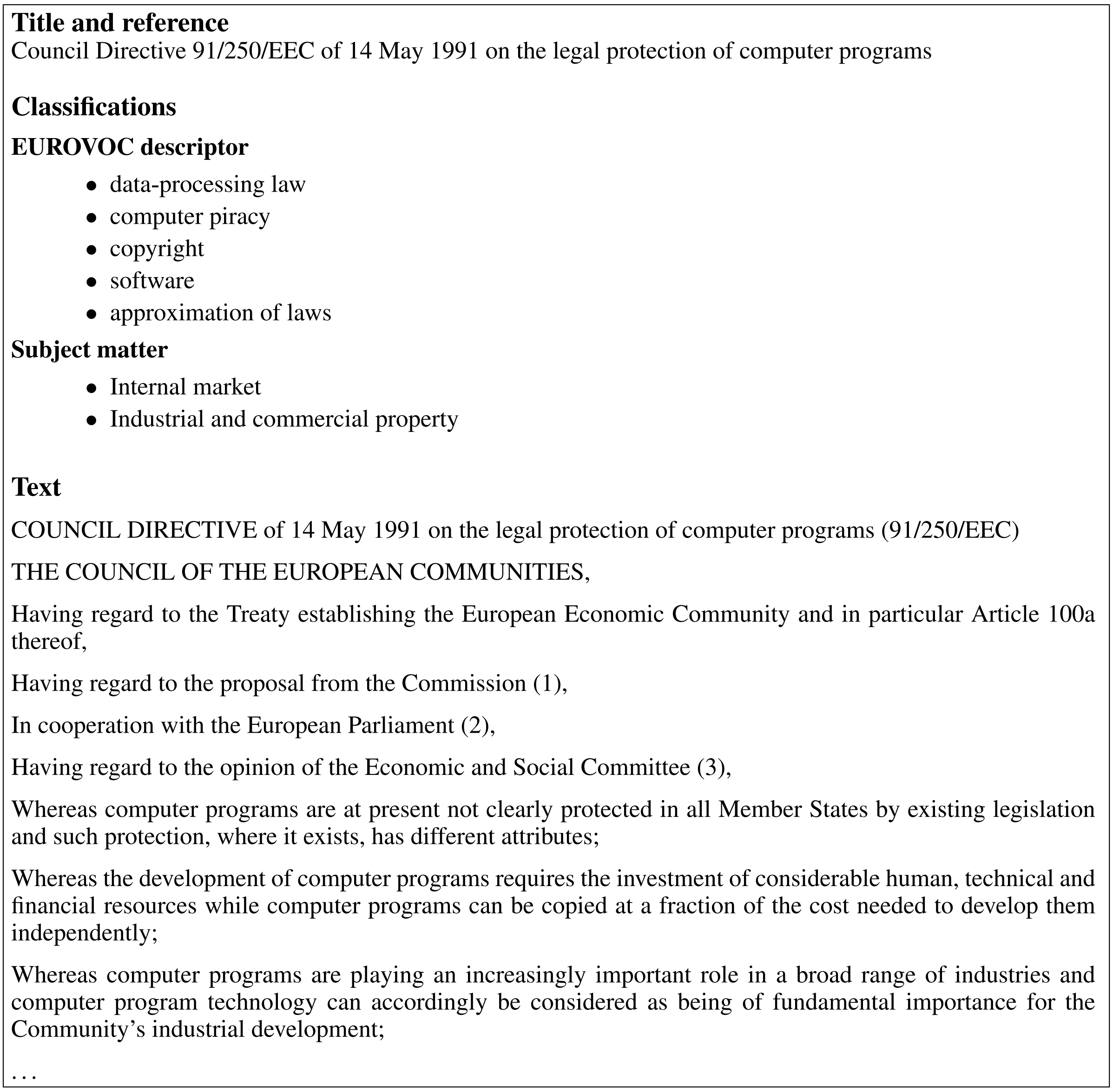
The pseudocode in Fig. 1 describes the MMP training algorithm. When the MMP algorithm receives a training instance $\bar{x}$, it calculates the inner products, the ranking and the loss on this ranking in order to determine whether the current model needs an update. For determining the ranking loss, any of the methods of Sec. 2.1 is appropriate, since they all return a low value on good rankings. This allows to optimize the ranking in accordance with the used ranking loss. If the ranking is perfect, the algorithm is done, otherwise it calculates the error set of wrongly ordered class pairs E. The wrongly ranked classes are also represented in F. In the next step, each class that is present in a pair of E receives a penalty score. This is done according to a selectable penalty function. Crammer and Singer [2003] propose several methods, including a function that returns a value proportional to the difference of the scalar products of both classes. The most successful one, however, seemed to be the uniform update method, where each pair in E receives the same score. In the next step, the update weights τ are normalized and each perceptron whose class was wrongly ordered is updated.

An example will illustrate the peculiarities of the MMP update method: Suppose that all classes are correctly ordered except for one relevant and three irrelevant classes. The three negative classes are ranked immediately over the positive. The error set contains three wrongly ordered pairs and according to the uniform update method the positive class will receive in the sum a penalty of 3 and the negatives each 1. Thus the perceptron of the positive class will be updated to a degree three times as great compared with the other three, in accordance with the degree to which it contributed to the wrong ranking. Note that regardless of the used penalty function the positive and the negative classes receive in total the same penalty scores and these are afterwards normalized, so that the degree of the overall model update only depends on δ, i.e. on the quality of the ranking. More precisely, the hyperplanes of the perceptrons of the relevant classes are translated by a total amount of $\delta\,\bar{x}$, and the remaining classes by $-\,\delta\,\bar{x}$. In summary, the degree of the update for a particular perceptron depends 1) on the used penalty method, 2) on how much it contributed to the wrong ranking, and 3) on the general ranking performance.

3 EUR-Lex Repository

The EUR-Lex/CELEX (Communitatis Europeae LEX) Site[1] provides a freely accessible repository for European Union law texts. The accessible documents include the official Journal of the European Union, treaties, international

[1] http://eur-lex.europa.eu



Title and reference

Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs

Classifications

EUROVOC descriptor

- data-processing law
- computer piracy
- copyright
- software
- approximation of laws

Subject matter

- Internal market
- Industrial and commercial property

Text

COUNCIL DIRECTIVE of 14 May 1991 on the legal protection of computer programs (91/250/EEC)

THE COUNCIL OF THE EUROPEAN COMMUNITIES,

Having regard to the Treaty establishing the European Economic Community and in particular Article 100a thereof,

Having regard to the proposal from the Commission (1),

In cooperation with the European Parliament (2),

Having regard to the opinion of the Economic and Social Committee (3),

Whereas computer programs are at present not clearly protected in all Member States by existing legislation and such protection, where it exists, has different attributes;

Whereas the development of computer programs requires the investment of considerable human, technical and financial resources while computer programs can be copied at a fraction of the cost needed to develop them independently;

Whereas computer programs are playing an increasingly important role in a broad range of industries and computer program technology can accordingly be considered as being of fundamental importance for the Community's industrial development;

. . .

Figure 2: Excerpt of a EUR-Lex sample document with the CELEX ID 31991L0250. The original document contains more meta-information. We trained our classifiers to predict the EUROVOC descriptors and the subject matters based on the text of the document.

agreements, legislation in force, legislation in preparation, case-law and parliamentary questions. The documents are available in in most of the languages of the EU, and in the HTML and PDF formats. We retrieved the HTML versions with bibliographic notes recursively from all documents in the English version of the *Directory of Community legislation in force*[2], in total 19,601 documents. Only documents related to secondary law (in contrast to primary law, the constitutional treaties of the European Union) and international agreements are included in this repository. The legal form of the included acts are mostly *decisions* (8,917 documents), *regulations* (5,706), *directives* (1,898) and *agreements* (1,597).

The bibliographic notes of the documents contain information such as dates of effect and validity, authors, relationships to other documents and classifications. The classifications include the assignment to several EUROVOC descriptors, directory codes and subject matters, hence all classifications are multilabel ones. We restricted our view to the EUROVOC and the subject matter classifications. EUROVOC is a multilingual thesaurus providing a controlled vocabulary for European Institutions[3]. Documents in the documentation systems of the EU are indexed using this thesaurus.

Figure 2 shows an excerpt of a sample document with all information that has not been used removed. The full document can be viewed at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31991L0250:EN:NOT.

3,993 different EUROVOC descriptors were identified in the retrieved documents, each document is associated to 5.37 descriptors in average. In contrast there are only 201 different subject matters appearing in the dataset, with a mean of 2.23 labels per document.

3.1 Data Preprocessing

The main text was extracted from the HTML documents, excluding HTML tags, bibliographic notes or other additio-

[2] http://eur-lex.europa.eu/en/legis/index.htm

[3] http://europa.eu/eurovoc/

	1 epoch		2 epochs		10 epochs		20 epochs	
	OAA	MMP	OAA	MMP	OAA	MMP	OAA	MMP
ISERR ×100	59.44	49.31	52.35	43.17	46.92	37.32	45.71	37.34
ONEERR ×100	27.18	22.96	21.05	18.08	17.04	14.13	16.50	14.33
ERRSETSIZE	81.64	12.57	62.64	11.12	50.07	8.825	46.79	8.356
MARGIN	59.51	10.59	47.40	9.503	38.99	7.643	36.54	7.289
AVGP	64.59	77.95	71.18	81.73	75.21	85.13	76.07	85.25

Table 1: Average losses for the *subject matter* classification.

	1 epoch		2 epochs		5 epochs	
	OAA	MMP	OAA	MMP	OAA	MMP
ISERR ×100	98.82	98.43	98.03	97.03	96.82	95.24
ONEERR ×100	47.29	70.14	40.52	50.37	34.78	36.47
ERRSETSIZE	8422.0	807.2	7320.3	926.7	6530.9	995.6
MARGIN	3214.0	582.4	2981.3	701.0	2794.1	756.9
AVGP	28.01	31.01	33.43	42.07	37.71	49.91

Table 2: Average losses for the *EUROVOC descriptor* classification.

nal information that could distort the results, and was then finally tokenized. The tokens were transformed to lower case, stop words were excluded, and the Porter stemmer algorithm was applied.[4] In order to perform cross validation, the instances were randomly distributed into ten folds. The tokens were projected into the vector space model using the common TF-IDF term weighting [Sebastiani, 2002]. In order to reduce the memory requirements, of the approx. 200,000 resulting features we selected the first 10,000 ordered by their document frequency. This feature selection method is very simple and efficient and independent from class assignments, although it performs comparably to more sophisticated methods using chi-square or information gain computation [Yang and Pedersen, 1997]. In order to ensure that no information from the test set enters the training phase, the TF-IDF transformation and the feature selection were conducted only on the training sets of the ten cross-validation splits.

4 Evaluation

4.1 Algorithm Setup

For the MMP algorithm we used the ISERR loss function and the uniform penalty function. This setting showed the best results in [Crammer and Singer, 2003] on the RCV1 data set. Both algorithms use perceptrons without thresholds, as described in Section 2.2, and all perceptrons were initialized with random values.

4.2 Ranking Performance

The results of a direct comparison of OAA and MMP for the subject matter and EUROVOC descriptor classifications are presented in Table 1 and Table 2. The values for ISERR, ONEERR and AVGP are shown ×100% for better readability, AVGP is also presented in the conventional way (with 100% as the optimal value) and not as a loss function. The number of epochs indicates the number of times that the online-learning algorithms were able to see the training instances.

For the subject matter, the results clearly show that the MMP algorithm outperforms the simple one-against-all approach (all differences are statistically significant). Especially on the losses that directly evaluate the ranking performance the improvement is quite pronounced. The smallest difference can be observed in terms of ONEERR, which evaluates the top class accuracy. Note also that the MMP algorithm is not able to improve its performance after 10 epochs. This partially confirms the results of Crammer and Singer. They observed that after reaching a certain amount of training examples, the improvement stops and after that point the performance even becomes worse. This point seems to be reached at the latest at ten epochs on the EUR-Lex data for subject matter classification.

The results on the EUROVOC descriptor data set confirm the previous results. The differences in ERRSETSIZE and MARGIN are very pronounced. In contrast, in terms of ISERR the MMP algorithm is worse than one-against-all, even after five epochs. It seems that with an increasing amount of classes, the MMP algorithm has more difficulties to push the relevant classes to the top such that the margin is big enough to leave all irrelevant classes below, although the algorithm in general clearly gives the relevant classes a higher score than the one-against-all approach. An explanation could be the dependence between the perceptrons of the MMP. This leads to a natural normalization of the scalar product, while there is no such restriction when trained independently as done in the binary relevance algorithm. As a consequence there could be some perceptrons that produce high maximum scores and thereby often arrive at top positions at the overall ranking.

The fact that in only approximately 5% of the cases a perfect classification is achieved and in only approx. 65% the top class is correctly predicted should not lead to an underestimation of the performance of the two algorithms. Considering that with almost 4000 possible classes and only 5.3 classes per example the probability to guess a correct class is less than one percent, namely 0.13%, the performance is indeed substantial.

4.3 Computational Costs

In order to allow a comparison independent from external factors such as logging activities and the run-time environment, we measured the computational cost in terms of vector additions and scalar multiplications. We also ignored minor operations that have to be performed by both algorithms, such as sorting or internal real value operations. An overview is given in Table 4.3, together with the CPU-times that were measured on a AMD Dual Core Opteron 2000

[4] The implementation from the Apache Lucene Project (http://lucene.apache.org/java/docs/index.html) was used.

subject	training	testing
OAA	48.74 s 3,545,841 mult. + 44,113 add.	5.42 s 393,960 mult.
MMP	54.08 s 3,545,841 mult. + 304,245 add.	5.39 s 393,960 mult.

EUROVOC	training	testing
OAA	621,814 s 70,440,513 mult. + 224,615 add.	98,354 s 7,826,280 mult.
MMP	818,147 s 70,440,513 mult. + 15,305,659 add.	93,467 s 7,826,280 mult.

Table 3: Computational costs in CPU-time and vector multiplications and additions on both data sets.

MHz for additional reference information.

As per design, for both algorithms the number of scalar multiplications is equal, namely 3,545,841 for each training iteration and 393,960 for testing on the subject matter data and 70,440,513 and 7,826,280 respectively for the EUROVOC data. In contrast, the number of vector addition operations differ: while the one-against-all method requires 44,113 operations for the subject matter and 224,615 for the EUROVOC classifications (for the first iteration, cross-validated), the MMP has higher costs with 304,245 and 15,305,659 operations respectively.

If we analyze the number of perceptron updates, i.e. additions, as a function of the number of classes, it is interesting to see the contrary behavior of both algorithms: while the one-against-all algorithm reduces the ratio of updated perceptrons per training example from 1.23% to 0.32% when increasing the number of classes from 202 to 3993, the MMP algorithm doubles the rate from 9.08% to 21,81%.

For the MMP this behavior is natural: with increasing numbers of classes the error set size increases and as a consequence the number of updated perceptrons. The MMP adapts itself to the increased scale and complexity. An explanation for the one-against-all case could be that due to the decreased number of positive examples for each base classifier (in average 23 = classes per example / (total number of classes * number of training examples) for the EUROVOC classes) the perceptrons quickly adopt the generally good rule to always return a negative score, which leads to only a few errors and consequently to little corrective updates. Note that a base classifier that always predicts the negative class would make approx. 95,000 errors on the training set, compared to the 224,615 of the perceptrons in the training phase.

5 Conclusions

In this paper, we evaluated two known approaches for efficiently solving multilabel classification tasks on a large-scale text classification problem taken from the legal domain: the EUR-Lex database. The experimental results confirm that the MMP algorithm, which improves the more commonly used one-against-all (OAA) approach by employing a concerted training protocol for the classifier ensemble, is very competitive and well applicable in practice for solving large-scale multilabel problems. The increase in predictive performance has to be paid by the MMP algorithm with a small increase in computational complexity that in our opinion is not important in practice.

The average precision rate for the EUROVOC classification task, a multilabel classification task with 4000 possible labels, approaches 50%. Rougly speaking, this means that the (on average) five relevant labels of a document will (again, on average) appear within the first 10 ranks in the relevancy ranking of the 4,000 labels. This is a very encouraging result for a possible automated or semi-automated real-world application for categorizing EU legal documents into EUROVOC categories.

For future research, on the one hand we see space for improvement and extension of the MMP algorithm for example by using a calibrated ranking approach [Brinker et al., 2006]. The basic idea of this algorithm is to introduce an artificial label which, for each class, separates the relevant from irrelevant labels. On the other hand, we are in the process of evaluating different binarization approaches such as pairwise learning [Fürnkranz, 2002]. An evaluation of this pairwise approach on the Reuters Corpus Volume 1 [Lewis et al., 2004], which contains over 100 classes and 800,000 documents, showed a substantial improvement over the MMP method [Loza Mencía and Fürnkranz, 2007]. The main obstacle to the applicability is the large memory requirement of the approximately 8,000,000 perceptrons that are needed to represent such an ensemble. We are currently investigating ways for reducing that number without losing the effectiveness of pairwise approaches. Furthermore, we also need to compare our performance to those of established algorithms that are potentially capable of handling large-scale multilabel data, such as the Naive Bayes algorithm.

Acknowledgements

This work was supported by the EC 6th framework project ALIS (Automated Legal Information System).

References

Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

Klaus Brinker, Johannes Fürnkranz, and Eyke Hüllermeier. A Unified Model for Multilabel Classification and Ranking. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-06)*, 2006.

Koby Crammer and Yoram Singer. A Family of Additive Online Algorithms for Category Ranking. *Journal of Machine Learning Research*, 3(6):1025–1058, 2003.

Ofer Dekel, Shai Shalev-Shwartz, and Yoram Singer. The Forgetron: A Kernel-Based Perceptron on a Fixed Budget. In *Advances in Neural Information Processing Systems 18*, 2005.

Yoav Freund and Robert E. Schapire. Large Margin Classification using the Perceptron Algorithm. *Machine Learning*, 37(3):277–296, 1999.

Johannes Fürnkranz. Round Robin Classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397, 2004.

Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz S. Kandola. The Perceptron Algorithm with Uneven Margins. In *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002)*, pages 379–386, 2002.

Eneldo Loza Mencía and Johannes Fürnkranz. Pairwise learning of multilabel classifications with perceptrons. Technical Report TUD-KE-2007-05, Technische Universität Darmstadt, Knowledge Engineering Group, 2007. http://www.ke.informatik.tu-darmstadt.de/publications/reports/tud-ke-2007-05.pdf.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

Shai Shalev-Shwartz and Yoram Singer. A New Perspective on an Old Perceptron Algorithm. In *Learning Theory, 18th Annual Conference on Learning Theory (COLT 2005)*, pages 264–278. Springer, 2005.

Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420. Morgan Kaufmann Publishers Inc., 1997. ISBN 1-55860-486-3.

Multi-objective Frequent Termset Clustering

Andreas Kaspari and Michael Wurst
University of Dortmund
Computer Science VIII
44221 Dortmund, Germany
andreas.kaspari@uni-dortmund.de
michael.wurst@uni-dortmund.de

Abstract

Large, high dimensional data spaces, are still a challenge for current data clustering methods. Frequent Termset (FTS) clustering is a technique developed to cope with these challenges. The basic idea is to first find frequent termsets and then to transform the resulting directed acyclic graph into a tree by deleting edges and termsets. While this technology was originally developed for document clustering, it can be applied in many other scenarios as well. Existing approaches to FTS clustering apply different heuristics to convert a set of frequent termsets into a final cluster set. In this work, we explore another approach. We first make the desirable properties of an FTS clustering explicit by stating different objective functions. We then show, how these functions are related to each other and that, in general, they are conflicting. This leads directly to the formulation of FTS clustering as a multi-objective optimization problem. We explore the ability of this approach to produce different, pareto-optimal solutions on a social bookmarking data set.

1 Introduction

Data clustering is a key technology to access large data and information spaces in a structured way. While significant progress was made in this area, there are still many open challenges when facing complex, high dimensional data spaces, such as text collections. Traditional clustering approaches, e.g. agglomerative clustering, are not well-suited for such scenarios. A first issue is that it is hard to find clusters in a high dimensional data space, as clusters often only exist in subspaces of this space. In order to cluster publications in the area of artificial intelligence, only terms that concern this topic or its subtopics are relevant, other terms will blur the result. Second, traditional methods often lead to clusters that are hard to interpret for the user. Comprehensible results are, however, essential for clustering, which is in large parts an explorative task. Third, many traditional clustering approaches require the user to set several parameters, a task that is far from being trivial. Finally, many clustering approaches scale poorly with an increasing number of data points and dimensions.

These observations led, among others, to several new clustering approaches, that can be summarized as frequent termset or itemset clustering (Fung *et al.* [2003]; Beil *et al.* [2002]). The basic idea is to first find frequent itemsets in the underlying data. This task can be accomplished even on very large data sets with many dimensions (Agrawal *et al.* [1993]). The actual clustering step is then performed on the resulting frequent itemsets and not on the original data. As clusters are represented by frequent itemsets, the resulting cluster structure automatically contains cluster descriptions that are comprehensible for the user. Also, the resulting clusters are by definition clusters only in subspaces of the original data space. This technique has been particularly applied to text clustering, where the number of dimensions is extremely high. In this case, frequent itemsets are sets of terms, that often co-occur in the documents to be clustered. The resulting clustering is a tree consisting of combinations of terms arranged according to the subset relation (see figure 1).

While frequent termset clustering was successfully applied to different areas, there are still some open points. Existing approaches use a number of heuristics to derive a cluster structure from a set of frequent termsets, implicitly trading-off diverse diserable criteria of such cluster structures. These criteria include maximal coverage, minimal overlap, simplicity of the overall structure etc. This implicit, heuristic merging of criteria makes it hard for the user to control the clustering process. This is particularly true, as these criteria are often conflicting. The overlap, for instance, can often be reduced by incorporating additional clusters in the final cluster structure. This, on the other hand, makes the final structure more complex and harder to overlook.

In this work we therefore choose another approach. We first analyze which properties are desirable for an FTS clustering and then derive several, partially conflicting objective functions. Instead of merging them in a heuristic manner, we first analyze their mutual relations. We then use a subset of actually conflicting criteria in an explicit, multi-objective optimization procedure. In general multi-objective optimization delivers more than one solution, as several criteria are involved, that may be in conflict with each other. This gives the user the opportunity to choose a desired solution from a set of pareto-optimal solutions, instead of having to search a large space of parameters manually in a laborous trial-and-error process.

This principle has been applied to other clustering tasks as well. In particular there are several approaches that apply multi-objective optimization to the task of feature selection for clustering. Since initial approaches, as Kim *et al.* [2000, 2002] or Morita *et al.* [2003], showed some weaknesses, a new sound framework was proposed in Mierswa and Wurst [2006a]. The proposed criteria trade off the cluster quality against the similarity of the resulting feature space to the original one. This approach is denoted as information preserving feature selection. The rationale behind this idea is, that clustering is basically an explorative task, that should describe or summarize the given data in an adequate and unbiased way. Information should only be omitted if this leads to a better and simpler cluster structure. The user can then decide in an interactive way, which features carry useful information and which of them are noise by inspecting the set of pareto-optimal solutions. In Mierswa and Wurst [2006b] this idea was extended to feature construction as well.

In this work, we propose a multi-objective framework for frequent termset clustering as well as several objective functions for this task. We analyze the mutual relation of these objective functions and the soundness of the framework on a real world social bookmarking data set. The work is structured as follows. In section 2 we give a brief introduction to frequent termset clustering and argue, that finding an optimal clustering implies finding a trade-off between different, conflicting criteria. In section 3 we then give a short introduction to multi-objective optimization. This is the point of departure for the formulation of several, partially conflicting objective functions for frequent termset clustering and for the analysis of their mutual relations in section 4. In section 5 we discuss the application of the approach to the problem of clustering tags in a social bookmarking system and present empirical results. In section 6 we summarize the results and point out the future direction of research.

2 Frequent Termset Clustering

In the following we assume a set of uniquely identified resources R and a set of terms T. We can then define the notion of a termset C as a set of terms in T, thus $C \subseteq T$.

We further assume a function $g : T \times R \rightarrow \mathbb{N}$ that assigns term occurrences to resources. In the simplest case, this function is binary, stating whether the term is assigned to the resource or not. In general, it can express the relevance of the term to the resource, as known from the vector space model.

Note that while this terminology suggests an application in text clustering applications, frequent termset clustering can be applied to other areas as well. In this case resources are general transactions and terms are items.

Based on the function g we define a cover relation $\nabla \subseteq R \times \mathcal{P}(T)$ for which the following holds:

$$r \nabla C \equiv \forall t \in C : g(t,r) > 0 \tag{2.1}$$

Thus a resource is covered by a termset, if all terms in the termset are assigned to the resource. The support of a termset is defined as the fraction of resources it covers.

Algorithms as Apriori (Agrawal *et al.* [1993]; Agrawal and Srikant [1994]) or FPGrowth (Han *et al.* [2000]) allow to efficiently find the set of frequent termsets, that have a support that exceeds a given minimal support. For an overview on frequent itemset mining refer to Goethals [2003].

This set of frequent termsets can then be arranged into a directed acyclic graph (DAG), by using the subset relation. Such a structure is however often not very well suited to access and navigate a complex information space. This is especially true, if the number of frequent termsets is very high. Several approaches have been proposed to derive flat or hierarchical cluster structures from frequent termsets. A first, simple approach to exploit the idea of frequent terms for clustering is presented in Wang *et al.* [1999]. While this algorithm does not make direct use of frequent itemset mining, it exploits the same underlying idea. Clusters consist of resources and are characterized by terms. For each term, the support within each cluster is calculated. Based on a threshold value, terms concerning a cluster are separated into large and small terms. Resources are then assigned to clusters in such a way, that each cluster contains as few small terms as possible and that the overlap of large terms among clusters is minimized. The complexity of the clustering is controlled by setting the support parameter appropriately.

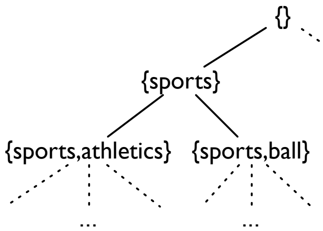


Figure 1: Excerpt from a clustering produced by the methods proposed in Fung et al. *[2003]*

A natural extension of this idea is presented in Beil *et al.* [2002]. First a frequent itemset algorithm is applied to identify frequent termsets. Then a subset of this set of frequent termsets is selected, that minimizes the overlap among the selected frequent termsets, while covering all resources. This method is extended to produce hierarchical clusters by first applying it to all frequent termsets of size one, then to all corresponding frequent termsets of size two, and so on.

In Fung *et al.* [2003] an alternative method is proposed, that converts the DAG of frequent termsets into a simpler, hierarchical cluster structure. First, a frequent itemset mining algorithm is used to identify frequent termsets. These frequent termsets form an initial clustering by arranging them according to the subset relation. Each resource (in this case text documents) is assigned to exactly one frequent termset using a centroid-based approach. Then, in a bottom-up way, the algorithm selects a parent node for each frequent termset by calculating a centroid resource of all resources in a cluster. Both steps aim to minimize the overlap among nodes on the same level of the cluster tree. The resulting tree is pruned applying several heuristics, as to reduce its complexity. Figure 1 shows an example.

While this approach can lead to good results, a general issue is the use of several heuristic steps that partially contain hidden parameters. These make it unclear which criteria are actually optimized. Also, the user has no influence on these criteria and on their combination.

In this work we propose a systematic approach to identify different desirable criteria of frequent termset clusterings. Implicitly, the above approaches aim to achieve the following:

- *Coverage*: The number of resources that cover at least one of the termsets should be as high as possible, because any resource for which no such termset exists is not represented in the cluster structure. The approaches above achieve full coverage, which is however not suitable in the presence of outliers. Therefore coverage should be considered a continuous value.
- *Overlap*: Traditional clustering algorithms, such as k-means, usually try to make clusters as separated as

possible. This principle is also applied by the FTS Clustering approaches presented above. We will discuss in section 5, whether this is always necessary and desirable.

- *Detailedness*: The resulting clustering should be as detailed as necessary, as the task of clustering is to describe the underlying data set, leaving the exploration to the user. This usually implies that the cluster tree should be as deep as possible, allowing fine grained distinctions.
- *Simplicity*: The resulting structure should be easy to navigate and overlook by a human user.

These criteria are clearly in conflict with each other. Adding additional clusters makes the clustering more complex and harder to navigate, will however usually increase the coverage and decrease the overlap. A deeper and thus more complete clustering will usually introduce additional overlap, etc.

Instead of using heuristic procedures to combine these criteria, we formalize objective functions that reflect these properties and apply them in a multi-objective optimization procedure. This allows to make the trade-off between different criteria explicit and will help to gain insight in the ways in which the criteria are related.

In the following we give a general definition for a frequent termset clustering. In contrast to Fung *et al.* [2003] we do not require the resulting cluster structure to be a tree, but a DAG.

Definition 2.1. (Frequent Termset-Clustering)
A termset clustering orders a set of (frequent) termset in a hierarchical way.

- $\mathsf{C} \subseteq \mathcal{P}(T)$ is an non-empty, finite set of termsets. Each termset represents a cluster. C is denoted as the cluster set.
- The relation $\prec$: $\mathcal{P}(T) \times \mathcal{P}(T)$ defines a hierarchical order on all clusters in C. For all pairs of clusters $C, D \in \mathsf{C}$ the following holds:

$$C \prec D \Leftrightarrow (C \subset D) \wedge (|C| = |D| - 1) \qquad (2.2)$$

We require some additional constraints on a frequent termset clustering.

Definition 2.2. (Frequent Itemset Clustering Conditions)
A cluster set $\mathsf{C} \subseteq \mathcal{P}(T)$ must fulfill the following constraints:

$$\emptyset \in \mathsf{C} \qquad (2.3a)$$
$$\forall D \in \mathsf{C} \text{ with } D \neq \emptyset : \exists C : C \prec D \qquad (2.3b)$$
$$\forall C \in \mathsf{C} : \exists r \in R : r \nabla C. \qquad (2.3c)$$

Condition (2.3a) states that the empty set must be contained in each cluster set. Condition (2.3b) ensures that there is a path from each cluster to the empty set (thus the cluster set is a connected graph). Condition (2.3c) ensures that each cluster contains at least one resource.

The set $\mathfrak{C}$ will denote all possible cluster structures that can be derived from T, considering only termsets that meet a certain minimal support.

$$\mathfrak{C} = \{\mathsf{C} \mid \mathsf{C} \subseteq \mathsf{FTS} \text{ and } \mathsf{C} \text{ valid}\} \qquad (2.4)$$

where FTS is the set of frequent termsets with a minimal support of σ.

Based on this definition, we present a multi-objective optimization algorithm that selects several cluster sets as subsets of the set of all frequent termsets. These are then presented to the user. The selection process will be governed by several optimization criteria that will be presented in section 4.

3 Multi-objective Optimization

In the last section we argued, that frequent termset clustering is an inherently multi-objective problem. In the following, we will give a more precise definition of this notion.

Definition 3.1. (Multi-objective optimization problem)

$$\max \vec{f}(x) \text{ with } x \in S \qquad (3.1)$$

The set S describes the set of valid solutions. $\vec{f}$ assigns to each element $x \in S$ a solution vector from R^k, where R is a totally ordered set. Each element of this vector represents one criterion and k is the number of criteria.

Definition 3.2. (Pareto-dominance)
A solution vector $\vec{u}$ dominates a solution vector $\vec{v}$ (short $\vec{u} \succeq \vec{v}$), iff:

$$\forall i \in \{1, ..., k\} : u_i \geq v_i \qquad (3.2a)$$

as well as

$$\exists i \in \{1, ..., k\} : u_i > v_i \qquad (3.2b)$$

A solution vector is called non-dominated, if it is not dominated by any other solution vector.

Definition 3.3. (Pareto-optimal set)
For a multi-objective optimization problem, the set of pareto-optimal solutions is defined as

$$P^* := \{x \in S | \nexists y \in S : \vec{f}(y) \succeq \vec{f}(x)\} \qquad (3.3)$$

i.e. P^* contains only non-dominated solution vectors.

The task of multi-objective optimization is to determine a set of pareto-optimal solutions.

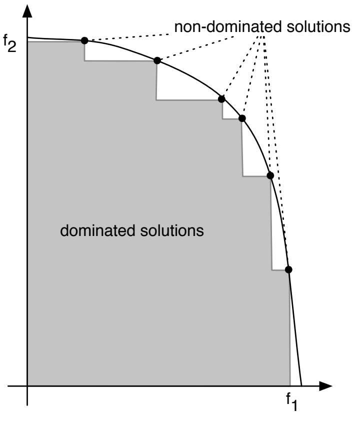


Figure 2: The pareto-optimal set is depicted as a black line. This line is often referred to as the pareto front.

Multi-objective optimization is very powerful, as it allows users to choose from a set of results, instead of a single result (Zitzler and Thiele [1999]; Coello Coello [1999]). While it would be in principle possible, to combine two or more criteria by a linear combination, this would work only, if these criteria are indeed in a linear relation to each other. In general, pareto fronts, as the one shown in figure 2, are highly non-linear and sometimes of very complex shape. The non-linear shape of the pareto front often guides the user in the process of selecting an appropriate solution, as it is very easy to visually identify interesting points, such as "elbows".

A desirable property of pareto-optimal solutions is that they should cover the space of possible solutions well, thus that the dots on a pareto front should be distributed as homogeneously as possible. This allows the user to choose from a wide variety of different solutions.

An important prerequisite for achieving such solution sets is that the criteria are in conflict. If two criteria are correlated, for instance, the solution set will contain only a single solution that dominates all other solutions.

4 Multi-objective Frequent Termset Clustering

4.1 Fitness Criteria for Frequent Termset Cluster Structures

As described above, the essential step of FTS clustering is to select the subsets of frequent termsets that are presented to the user. There are several possible criteria to assess the quality of each such subset. In general we define the fitness of a cluster structure in the following way.

Definition 4.1. Fitness of a cluster structure
A cluster structure fitness measure is a function $\mathfrak{C} \rightarrow \mathbb{R}$, that assigns to each valid cluster set $\mathsf{C} \in \mathfrak{C}$ a real-valued quality measure.

In section 2, several criteria were discussed that are implicitly used in most frequent termset clustering algorithms. These criteria are coverage, overlap, simplicity and detailedness. In the following we will derive formal definitions for these criteria and discuss whether they are well-suited for deriving comprehensible and complete cluster structures.

To ease the presentation of the individual criteria, we introduce several properties. The function $\mathrm{res} : \mathcal{P}(\mathcal{P}(T)) \rightarrow \mathcal{P}(R)$ calculates for a set M of clusters the set of all covered resources $r \in R$:

$$\mathrm{res}(\mathsf{M}) = \bigcup_{M \in \mathsf{M}} \{r \mid r \nabla M\} \tag{4.1}$$

The function $\mathrm{depth} : \mathfrak{C} \rightarrow \mathbb{N}$ calculates the depth of a cluster set:

$$\mathrm{depth}(\mathsf{C}) = \max_{C \in \mathsf{C}} |C| \tag{4.2}$$

The term $\mathsf{M}_{|i}$ denotes all sets in M that contain i elements, thus $\mathsf{M}_{|i} = \{M' \in \mathsf{M} \mid i = |M'|\}$, where $i \in \mathbb{N}$.

A property that is very popular is overlap. Algorithms as k-means aim to maximize the inter-cluster dissimilarity making individual clusters as disjoint as possible. In the context of ontologies, disjointness of concepts plays an important role for achieving sound definitions.

In the following we capture the average overlap among frequent termsets on the same level of a cluster set by the following expression.

Definition 4.2. (Overlap)
Given a clustering C, then $\mathrm{overlap} : \mathfrak{C} \rightarrow \mathbb{R}$ is defined as

$$\mathrm{overlap}(\mathsf{C}) = \frac{1}{\mathrm{depth}(\mathsf{C})} \cdot \sum_{i=1}^{\mathrm{depth}(\mathsf{C})} \frac{|\mathrm{res}(\mathsf{C}_{|i})|}{\sum_{C \in \mathsf{C}_{|i}} |\mathrm{res}(\{C\})|} \tag{4.3}$$

For $\mathrm{depth}(\mathsf{C}) = 0$, we assume $\mathrm{overlap}(\mathsf{C}) = 1$.

The function overlap measures the average overlap among clusters on each level. In the best case, each resource in R appears only in one cluster on each level. This criterion is very much akin to the overlap criterion proposed in Beil *et al.* [2002]. In section 5 we will argue, why it is not always desirable to optimize this criterion explicitely.

A second property of a frequent termset clustering is coverage. Many existing clusterings algorithms require to cluster all resources, thus to achieve a full coverage. In application areas, in which we can expect a high number of outliers, this can lead to poor results. It is therefore often desirable to ignore some resources in the clustering process, which however leads to a smaller coverage. Therefore, as second criterion, we define the coverage of a cluster set.

Definition 4.3. (Coverage)
Given a cluster set $\mathsf{C} \in \mathfrak{C}$, then the function $\mathrm{cover} : \mathfrak{C} \rightarrow [0, 1]$ is defined as follows:

$$\mathrm{cover}(\mathsf{C}) = \frac{|\mathrm{res}(\mathsf{C}_{|1})|}{|R|} \tag{4.4}$$

Overlap and coverage are usually in conflict with each other. Given a clustering that does not cover all resources, additional resources can only be covered by adding more frequent termsets. In this process the overlap can never decrease and is likely to increase, if the set of possible choices (frequent terms not yet chosen) is limited.

A third criterion is the simplicity of the clustering from a user perspective. The FTS clustering algorithms presented above achieve this by optimizing the inner cluster similarity and by applying heuristic pruning procedures.

We can however capture the simplicity of a cluster tree from a user perspective in an explicit way. For a flat clustering, the number of clusters should be chosen as small as possible, while achieving a cluster quality that is as high as possible. The underlying idea is that inspecting each cluster is connected with certain costs for the user. This idea can be transferred to hierarchical cluster sets as well. As such sets are often navigated top-down, we use the number of child nodes at the root and each inner node as indicator of the complexity of a cluster set.

Definition 4.4. (Child count)
Given a cluster set C, we define $\mathrm{succ} : \mathsf{C} \rightarrow \mathcal{P}(\mathsf{C})$ as

$$\mathrm{succ}(C) = \{D \in \mathsf{C} \mid C \prec D\} \tag{4.5}$$

and thus the set $\mathsf{C}' \subseteq \mathsf{C}$ as

$$\mathsf{C}' = \{C \in \mathsf{C} \mid |\mathrm{succ}(C)| > 0\} \tag{4.6}$$

Based on this, we can define

$$\mathrm{childcount}_{max}(\mathsf{C}) = \max_{C \in \mathsf{C}'} |\mathrm{succ}(C)| \tag{4.7}$$

Thus the complexity of a cluster structure is given as the most complex node.

The maximal child count will usually increase with increasing coverage, as to cover more resources, additional clusters are needed.

Another criterion is the depth of a cluster structure. As users often navigate a cluster structure top-down, the depth does not contribute to the complexity of the cluster tree as does the child count. In contrary, a high depth allows for more fine grained distictions at a lower level. The depth should therefore be maximized.

This is captured in the following criterion.

Definition 4.5. (**Depth of a cluster set**)

Let the set $\mathsf{C}' \subseteq \mathsf{C}$ be defined as

$$\mathsf{C}' = \{C \in \mathsf{C} \mid \nexists D : C \prec D\} \tag{4.8}$$

We can then define the following criteria:

$$\text{depth}_{avg}(\mathsf{C}') = \frac{1}{|\mathsf{C}'|}\sum_{C \in \mathsf{C}'}|C| \tag{4.9}$$

$$\text{depth}_{max}(\mathsf{C}') = \max_{C \in \mathsf{C}'}|C| \tag{4.10}$$

$$\text{depth}_{var}(\mathsf{C}') = \frac{1}{|\mathsf{C}'|}\sum_{C \in \mathsf{C}'}(|C| - \text{depth}_{avg}(\mathsf{C}')) \tag{4.11}$$

How is depth related to the other criteria? Per definition, coverage is mostly independent of the depth of the cluster tree, as only the resources covered by first level nodes are regarded. The same holds for child count. Overlap is likely to increase with increasing depth in many cases, as the distictions on top-level are stronger than on a more detailed level. This is also confirmed in our experiments, showing that if we do not explicitely optimize for depth, the algorithm produces rather shallow cluster structures.

One possibility to solve this problem is to use depth as an additional criterion in multi-objective optimization. This is however not fully satisfying, as it makes the optimization process more complex and as depth is not in strong conflict with any of the other criteria.

We therefore rather solve this problem by replacing coverage by another concept, namely completeness. The idea of completeness is, that the selected clusters should represent the given frequent termsets as good as possible.

Definition 4.6. (**Completeness**)
Given two cluster sets C and C_{ref}. We assume $\mathsf{C} \subset \mathsf{C}_{ref}$. Then the function $\text{compl} : \mathfrak{C} \times \mathfrak{C} \to \mathbb{R}$ is defined as:

$$\text{compl}(\mathsf{C}, \mathsf{C}_{ref}) = \frac{|\,\mathsf{C}\,|}{|\,\mathsf{C}_{ref}\,|} \tag{4.12}$$

Thus the more of the original frequent termsets are contained in the final clustering, the higher the completeness. This combines coverage and cluster depth in one straighforward criterion.

This criterion is in conflict with child count and with overlap. In section 5 we will analyze empirically, how the criteria are related to each other.

Beside the criteria presented here, there are many other possible criteria, such as to what grade the cluster structure resembles a tree or the number of paths by which a resource can be reached from the root. These criteria are however beyond the scope of this presentation.

4.2 Deriving Pareto-optimal solutions

Several algorithms were proposed for multi-objective optimization, almost all of them based on evolutionary computation (Coello Coello [1999]; Zitzler and Thiele [1999]). In this work we use the genetic algorithm NSGA-2 (Deb *et al.* [2000]) to approximate the set of pareto-optimal solutions. Individuals are represented as binary vectors, such that each element of the set of frequent termsets corresponds to one position in the vector. The cluster conditions are enforced by post-processing each individual.

The algorithm approximates the set $\mathfrak{C}^* \subseteq \mathfrak{C}$ of pareto-optimal cluster sets.

$$\mathfrak{C}^* = \{\mathsf{C} \in \mathfrak{C} \mid \nexists \mathsf{D} \in \mathfrak{C} : \vec{f}(\mathsf{D}) \succeq \vec{f}(\mathsf{C})\} \tag{4.13}$$

where $\vec{f}(\mathsf{D}) \succeq \vec{f}(\mathsf{C})$ states that there is no cluster set D that pareto-dominates the cluster set C with respect to the fitness functions $\vec{f}$. Thus $\mathfrak{C}^*$ contains all non-dominated cluster sets.

$\mathfrak{C}^*$ is also referred to as the pareto front. As mentioned before the algorithm should produce solutions that are equally spread across the pareto front.

5 Application and Evaluation

5.1 Social Bookmarking and Automatic Tag Clustering

Social bookmarking systems allow users to annotate resources on the internet with arbitrary textual descriptions called tags (Hammond *et al.* [2005]). These systems are extremely popular, as assigning tags is very simple (Shirky [2005]). In contrast to predefined keywords or categories, tags are very flexible and dynamic, allowing to capture the views of even rather small niche communities. Based on the “everything is a link” paradigm, users can navigate the hypergraph of user ids, tags and resources (Golder and Huberman [2006]).

While this is convenient for few tags and resources, it quickly becomes chaotic as the number of tags and resources grows. An important challenge is therefore to transform the user assigned tags into a navigation structure that is simple to overlook but on the other side reflects as many of the underlying resources as possible. Such a structure combines the best of both worlds: the relative simplicity of predefined taxonomies and the flexibility and subjectivity of user assigned tags.

Traditional clustering methods are not well suited for this task, as the data space is extremely complex and sparse. Apparently frequent termset clustering is designed to deal with exactly this kind of data.

There are several approaches that cluster tags as to make the resulting structure easier to overlook and navigate. In Hassan-Montero and Herrero-Solana [2006], tags are selected that show a high degree of diversity by applying a *tf/idf*-like measure. These tags are then clustered using Bisecting k-means and Jaccard-Similarity. Begelman *et al.* [2006] and Kaser and Lemire [2007] represent tags as graphs on which they apply graph clustering algorithms to obtain sets of similar tags. These methods suffer from the same problem as other traditional clustering algorithms, namely the extremely high number of dimensions and the high number of tags to be clustered. Also, by making tag clusters as dissimilar as possible to each other, they implicitly minimize the overlap, which is not always appropriate, as will argued below. Finally, these approaches include partially complicated parametrization (e.g. the right choice of a similarity measure, number of clusters, etc.). This leads to a laborious trial-and-error procedure in practice. This problem is even more severe, as the resulting tag clusters do not contain cluster descriptions, making them harder to interpret by the user. In contrast, the parameterization of our multi-objective clustering is controlled by the optimization procedure. Instead of trying out different parameter settings, the algorithm directly proposes to the user several promising results.

An approach that applies frequent itemset mining to tag structures is described in Schmitz *et al.* [2006]. The authors use different kinds of projections to map tag assignments to transactions. They then use frequent itemset mining to derive association rules. These association rules can then be

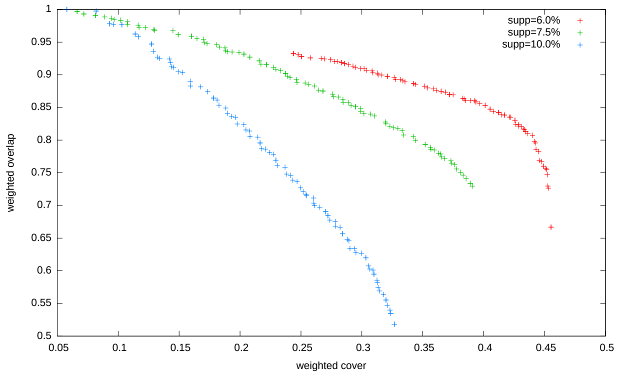


Figure 4: Pareto front for Overlap vs. Coverage for different supports

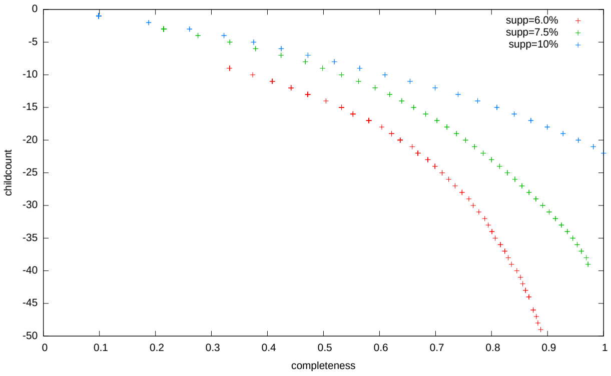


Figure 5: Pareto front for child count vs. completeness for different minimal supports

visualized as a cluster tree. The association rules are however not filtered or post-processed in any way. This, however was identified as a crucial step that frequent termset clustering achieves.

5.2 Experimental Results

We applied the approach proposed above to the freely available social bookmarking data set derived from the Bibsonomy system (Hotho *et al.* [2006]). This data set contains the tag assignments of about 780 users. The number of resources tagged by at least one user is about 59.000. The number of tags used is 25.000 and the total number of tag assignments is 330.000.

In order to find a set of frequent tagsets with respect to some minimal support σ, we must first define our notion of a tag's frequency. Although tag frequency can be defined in several ways, we consider a tag to be frequent if a certain number of users have assigned it to arbitrary resources.

We performed two experiments. In a first experiment we tried to optimize the overlap against the coverage of a cluster structure. This corresponds to the traditional idea of a cluster structure as being a level-wise disjoint structuring of entities in a domain of interest.

The resulting pareto front is shown in figure 4.

A more detailed analysis of the individual results shows the following:

- Cluster sets that fulfill the overlap criterion well are quite narrow and show a bad coverage.
- Cluster sets that fulfill the coverage criterion well are very broad and contain a lot of overlap.
- All resulting cluster structures are very shallow, as neither of the criteria forces the selection of deep clusters. Both, high coverage and low overlap can be achieved with clusters of level one.

This supports our claim, that optimizing overlap and coverage leads to not very detailed cluster structures that often resemble rather a flat partition than a hierarchical clustering.

There is also another interesting observation. Minimizing overlap removes the natural heterogeneity from the data. Some users may for instance have tagged news articles with country names (e.g. germany, france) and other users may have tagged them thematically (e.g. ecology). Now, if ecology and germany has an overlap that is very strong, then probably one of both tags will be removed, if the overlap is minimized. This is however not desirable, as both tags represent valid access paths to tagged resources.

Therefore we argue, that at least for tag clustering, minimizing the overlap is not only not an important criterion, it could even be counter-productive. In many other application scenarios, this holds as well.

In a second experiment, we used the two other criteria proposed in this work, namely maximal child count and completeness with respect to the frequent termsets. The corresponding pareto front is depicted in figure 5.

A nearer inspection of the pareto optimal results yields the following:

- Clusterings with a small maximum child count are narrow, but deep. This effect can be explained, as deep clustering yield on average a higher completeness.
- Clusterings with high completeness are broader, but still deep and contain much of the heterogeneity contained in the original data. They also show a very high coverage.

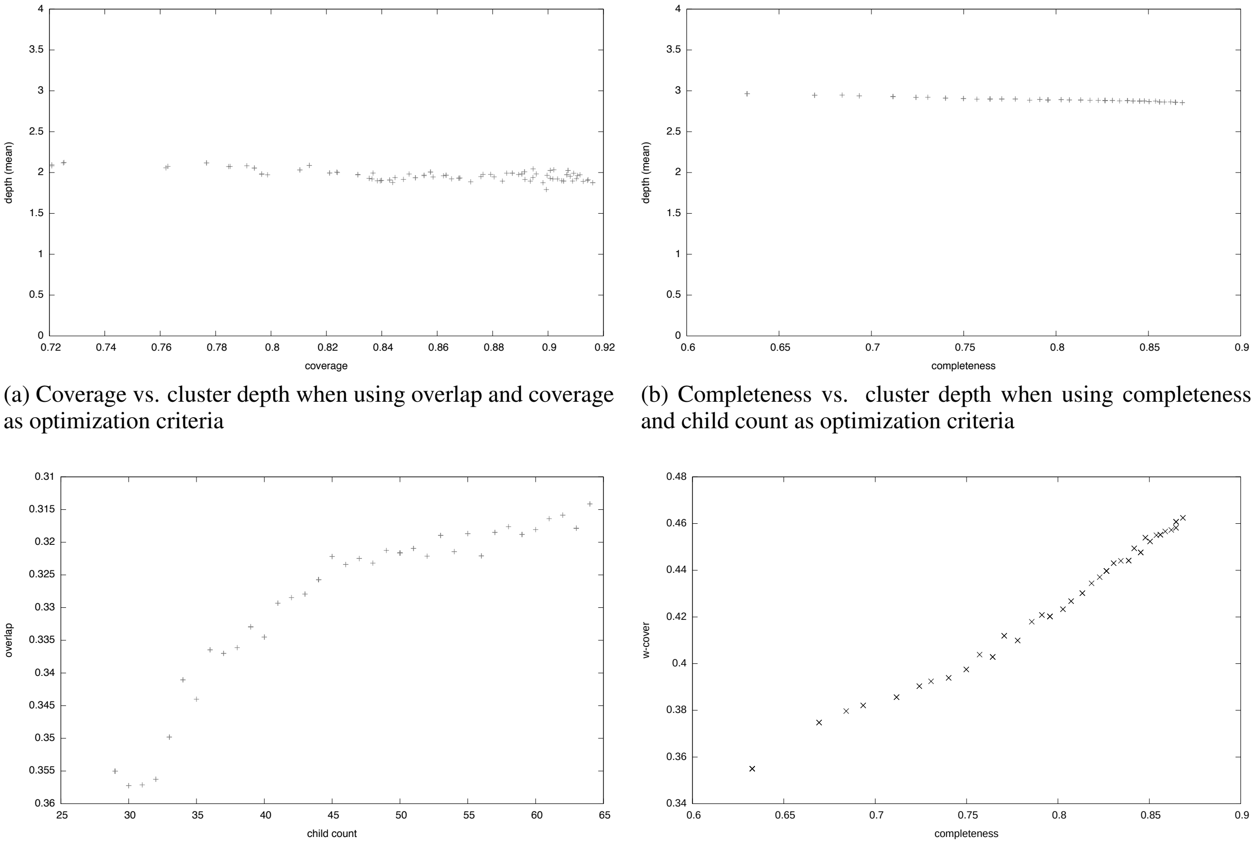
In this way we can actually optimize three criteria at once: the simplicity of the cluster structure, its detailedness in terms of cluster depth and the coverage of resources. These criteria are furthermore not biased to remove heterogeneity from the data, which is essential in many explorative applications.

Figure 3 shows how the different optimization criteria introduced in this work are related to each other. Figure 6 shows exemplarily the most simple tag structures produced by overlap vs. coverage and child count vs. completeness respectively.

6 Conclusion

In this work we presented an approach to frequent termset clustering that makes use of multi-objective optimization. This enables the user to choose from a set of promising results instead of having to search a complex parameter space in a trial-and-error way or having to rely on heuristic procedures. It also makes desirable properties of frequent termset clusterings explicit and allows to explore the relation among different optimization criteria in a systematic way.

We applied the algorithm in a social bookmarking scenario. The aim was to simplify the complex structure of user assigned tags in order to make this structure easier to navigate. We pointed out, that the overlap criterion applied by many clustering algorithms is not satisfying in this scenario, as it is likely to destroy the natural heterogeneity in the underlying data. Optimizing for small complexity and high completeness with respect to the selected frequent



(a) Coverage vs. cluster depth when using overlap and coverage as optimization criteria

(b) Completeness vs. cluster depth when using completeness and child count as optimization criteria

(c) Child count vs. overlap when using completeness and child count as optimization criteria

(d) Completeness vs. coverage when using completeness and child count as optimization criteria

Figure 3: These plots show how the different criteria are related to each other.

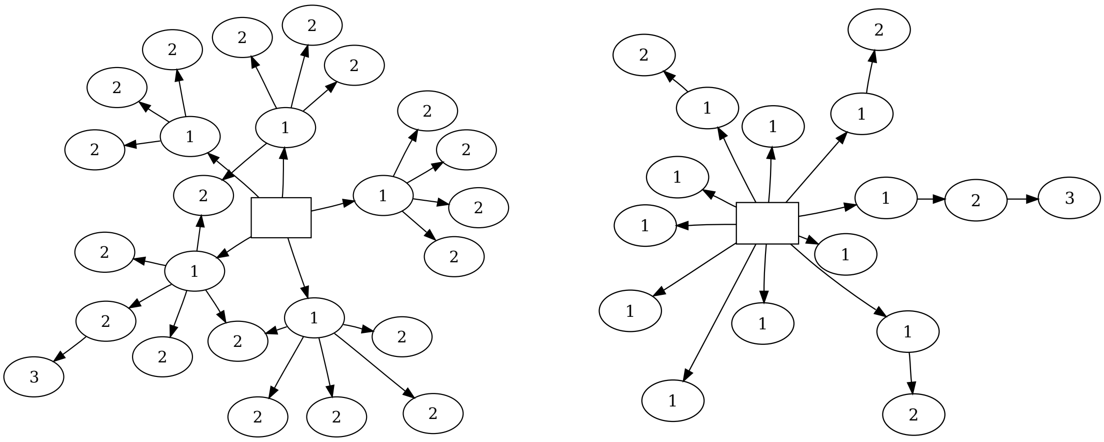


Figure 6: The simplest tag structures for child-count vs. completeness (on the left) and for coverage vs. overlap (on the right). The numbers in the nodes denote their depth in the tree. The tree on the left is deeper and better balanced than the tree on the right.

termsets on the other hand led to sound results that implicitly optimized other criteria as the cluster tree depth as well.

In our future work we plan to analyze additional criteria and their relationship. Also, we will explore the suitability of the approach in other application areas, such as customer segmentation. High-dimensional, complex data spaces are still challenging for clustering algorithms. We think that both, multi-objective optimization and frequent item based approaches will play an important role to solve these challenges in the future.

References

R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the International Conference on Very Large Data Bases*, 1994.

R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1993.

G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop*, 2006.

F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

C. A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):129–156, 1999.

K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature Conference*, 2000.

B. C. M. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *Proceedings of the SIAM International Conference on Data Mining*, 2003.

B. Goethals. Survey on frequent pattern mining. `http://www.adrem.ua.ac.be/bibrem/pubs/fpm_survey.pdf`, 2003.

S. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

T. Hammond, T. Hannay, B. Lund, and J. Scott. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine*, 11(4), 2005.

J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2000.

Y. Hassan-Montero and V. Herrero-Solana. Improving Tag-Clouds as Visual Information Retrieval Interfaces. In *Proceedings of the International Conference on Multidisciplinary Information Sciences and Technologies*, 2006.

A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. BibSonomy: A social bookmark and publication sharing system. In *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the International Conference on Conceptual Structures*, 2006.

O. Kaser and D. Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *WWW Workshop on Tagging and Metadata for Social Information Organization*, 2007.

Y. Kim, W.N. Street, and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2000.

Y. Kim, W. N. Street, and F. Menczer. Evolutionary model selection in unsupervised learning. *Intelligent Data Analysis*, 6:531–556, 2002.

I. Mierswa and M. Wurst. Information preserving multi-objective feature selection for unsupervised learning. In *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, 2006.

I. Mierswa and M. Wurst. Sound multi-objective feature space transformation for clustering. In *Workshop on Knowledge Discovery, Data Mining, and Machine Learning*, 2006.

M. Morita, R. Sabourin, F. Bortolozzi, and C.Y. Suen. Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In *Proceedings of the International Conference on Document Analysis and Recognition*, 2003.

C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme. Mining association rules in folksonomies. In *Proceedings of the IFCS Conference*, 2006.

C. Shirky. Ontology is overrated: Categories, links, and tags. `http://www.shirky.com/writings/ontology_overrated.html`, 2005.

K. Wang, C. Xu, and B. Liu. Clustering transactions using large items. In *Proceedings of the International Conference on Information and Knowledge Management*, 1999.

E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Interactive Ranking of Skylines Using Machine Learning Techniques

Weiwei Cheng, Eyke Hüllermeier, Bernhard Seeger, Ilya Vladimirskiy
Fachbereich Mathematik und Informatik
Philipps-Universität Marburg
{cheng,eyke,seeger,ilya}@mathematik.uni-marburg.de

Abstract

So-called skyline queries have received considerable attention in the field of databases in recent years. Roughly speaking, the skyline of a given set of objects, represented in terms of attributes with preferentially ordered domains, is given by the Pareto-optimal elements of this set. An important problem of Skyline queries is that answer sets can become extremely large. From an application point of view, a system response in terms of a *ranking* of elements, ordered according to the user's preferences, would hence be more desirable than an unordered set. In this paper, we propose a method for constructing such a ranking in an interactive way. The key idea of our approach is to ask for user feedback on intermediate results, and to use this feedback to improve, via the induction of a latent utility function, the current ranking so as to represent the user's preferences in a more faithful way.

1 Introduction

The *skyline* operator and corresponding skyline queries were first introduced by Baorzsaonyi et al. in 2001 [Borzsonyi *et al.*, 2001] and, since then, have attracted considerable attention in the field of databases. A skyline query is a special type of *preference query*: The skyline of a d-dimensional dataset consists of the data objects that are non-dominated in a Pareto sense and, therefore, potentially optimal for a user. Stated differently, each object that is not on the skyline is of no interest, as it is definitely worse than at least one other object in the dataset; more details about skylines will follow in Section 2.1.

Pareto-dominance is an extreme conception of dominance, as it is very demanding and, therefore, does not discriminate well between alternatives. Consequently, a skyline query may produce huge answer sets, a problem that aggravates with the dimensionality of the dataset and impairs the usefulness of skyline queries. One idea to avoid this problem is to exploit the preferences of a particular user. In fact, a specific user will usually not be indifferent between all skyline objects. Instead, the preference relation of this user will be a *refinement* of the Pareto-dominance relation, which is the weakest assumption on preferences one can make and is valid for *every* user. If the user's refined preference relation was known, it could be used to further reduce the set of candidate objects returned by the system or, even better, to order these objects according to their degree of preference.

In this paper, we present a special approach to information retrieval (IR) that combines skyline computation and ranking. The idea is to apply machine learning techniques in order to elicit a specific user's preferences, and to use this knowledge to compute a *ranking* of the skyline objects. This idea is completely in line with research trends in the IR field that are focused on user modeling [Shen *et al.*, 2005] and seek to exploit user feedback [Rochio, 1971; Robertson and Jones, 1976; Salton and Buckley, 1990; Shen and Zhai, 2005] for making retrieval systems interactive, context-aware, and adaptive [Detyniecki *et al.*, 2006].

To realize this idea, some kind of "training data" in the form of preference information is of course needed. In order to minimize the user's effort, we integrate corresponding preference questions into the learning process in a dynamic way. Roughly speaking, instead of separating the training phase from the application, the idea is to immediately exploit every piece of information: Each preference information is used to improve the current ranking of the skyline; in case the user is still not satisfied with the result, new information is requested, and this process is repeated until the user has eventually found what he was searching for. This procedure essentially corresponds to Rochio's well-known relevance feedback loop [Rochio, 1971].

The paper is organized as follows: The next section gives some background information on the skyline operator and research on ranking in the field of machine learning. Our approach to ranking of skylines is then introduced in Section 3 and empirically evaluated in Section 4. Section 5 outlines some reasonable extensions to be addressed in future work, and Section 6 concludes the paper.

2 Background

2.1 The Skyline Operator

Consider a set of objects $\mathbb{O}$ represented in terms of a fixed number d of attributes with preferentially ordered domains, that is, "the less the better" or "the more the better" attributes. In multi-criteria decision making, such attributes are also called *criteria*; typical examples are the price of a hotel and its distance from the beach. We shall denote the i^{th} attribute by A_i and its domain by $\mathcal{D}_i$. Thus, an element of $\mathbb{O}$ is a vector

$$\mathbf{a} = (a_1 \dots a_d) \in \mathcal{D}_1 \times \dots \times \mathcal{D}_d.$$

Without loss of generality, we restrict ourselves to "the more the better" attributes.

Given to objects $\mathbf{a}, \mathbf{b} \in \mathbb{O}$, the former is said to (Pareto) dominate the latter, $\mathbf{a} \succ \mathbf{b}$, if $a_i \geq b_i$ for all and $a_i > b_i$ for at least on $i \in \{1 \dots d\}$. An object $\mathbf{b}$ is non-dominated if $\mathbf{a} \not\succ \mathbf{b}$ for all $\mathbf{a} \in \mathbb{O}$. The skyline of $\mathbb{O}$ is then given by

$$\mathbb{S} \overset{\mathrm{df}}{=} \{ \mathbf{a} \in \mathbb{O} \,|\, \mathbf{a} \text{ is non-dominated} \}.$$

As mentioned previously, the skyline operator has recently attracted a lot of attention in the field of databases. A typical skyline query in SQL syntax may look as follows [Borzsonyi *et al.*, 2001]:

```
SELECT *
FROM Hotels
WHERE city = 'New Port'
SKYLINE OF price MIN, distance
   MIN
```

This query returns the set of hotels that are Pareto-optimal with respect to the dimensions price and distance (from the beach), which both ought to be minimized. Fig. 1 gives a graphical illustration of this example.

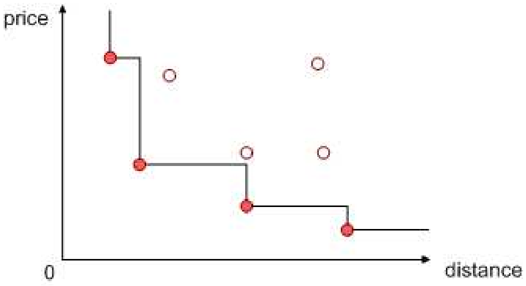


Figure 1: Skyline (filled points) for the hotel example.

The problem of computing a skyline in an efficient way has received special attention, and a large number of methods has already been devised (e.g., [Papadias *et al.*, 2005; Chomicki *et al.*, 2002; Kossmann *et al.*, 2002]). In this regard, index-based methods [Tan *et al.*, 2001] perform especially well, in particular when the size of the dataset is large. As potential disadvantages, one may note that such methods are restricted to numerical attributes for which an index can be created, and that index-based methods become problematic for high dimensions ("*curse of dimensionality*"). In our current implementation, we resort to the simple *block nested loop* approach, for which no kind of preprocessing is required [Borzsonyi *et al.*, 2001]. It maintains a set S of objects in main memory, which is initially empty, and scans the dataset. For each element $\mathbf{a} \in \mathbb{O}$, there are three possibilities:

(a) $\mathbf{a}$ is dominated by an object in S and, hence, can be discarded;

(b) one or more objects in S are dominated by $\mathbf{a}$ and, hence, can be replaced by $\mathbf{a}$;

(c) neither (a) nor (b) applies, so $\mathbf{a}$ is added to S without removing other elements.

One easily verifies that S finally corresponds to the skyline of $\mathbb{O}$, that is, $S = \mathbb{S}$.

Apart from algorithms for skyline computation, a number of conceptual modifications of skylines has been proposed in the literature, including, e.g., dynamic skylines [Papadias *et al.*, 2005], subspace skylines [Pei *et al.*, 2006], and skybands [Papadias *et al.*, 2005].

2.2 Ranking in Machine Learning

Ranking problems have received a great deal of attention in the field of machine learning in recent years. Here, the term "ranking" is used in different ways. In particular, a basic distinction between so-called *label ranking* and *object ranking* can be made [Fürnkranz and Hüllermeier, 2005]. The problem of label ranking can be seen as an extension of the basic setting of classification learning. Instead of learning a model that predicts, for each query instance, one among a finite set of class labels, the problem is to learn a model that predicts a complete ranking of all labels [Fürnkranz and Hüllermeier, 2003; Har-Peled *et al.*, 2002].

The problem considered in this paper is more related to object ranking or "learning to order things" (see, e.g., [Cohen *et al.*, 1999; Domshlak and Joachims, 2005]). Here, the task is to learn a function that, given a subset of objects $\mathbb{O}$ from an underlying reference set $\mathcal{X}$ as input, outputs a ranking of these objects. Training data typically consists of exemplary pairwise preferences of the form $\mathbf{x} \succ \mathbf{x}'$, with $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. As a key difference between object and label ranking, note that the latter associates a ranking of a fixed number of labels with every instance $\mathbf{x}$, whereas the former is interested in ranking the instances themselves.

From a learning point of view, an important issue concerns the evaluation of a predicted ranking. To this end, a loss function is needed that, given a true ranking τ and a prediction $\widehat{\tau}$ thereof, measures the "distance" between the former and the latter. The loss function is important, as it reflects the purpose that a ranking is used for. For example, in order to evaluate a ranking as a whole, one can resort to well-known rank correlation measures such as Kendall's tau [Kendall, 1955]. In many applications, however, the ranking itself is not of primary concern. Instead, a ranking is only used as a means to find certain objects, hence, only the positions of these objects are important. In this case, measures such as *precision* and *recall*, commonly used in information retrieval, are more suitable. In our experimental study, we shall use both types of measures (see Section 4.2).

3 Ranking on Skylines

As mentioned earlier, our goal is to give a *user-specific* answer to a skyline query by refining the general Pareto-dominance into a more specific preference relation valid for the user. In this regard, an important question is how to represent the user's preference. A common approach is to use a *utility function* for this purpose, that is, a mapping $U : \mathbb{O} \rightarrow \mathbb{R}$ that assigns a real utility degree to each object $\mathbf{a} \in \mathbb{O}$. Obviously, given a utility function, ranking becomes quite easy: $\mathbf{a}$ will precede $\mathbf{b}$ in the user-specific ranking if $U(\mathbf{a}) \geq U(\mathbf{b})$.

The assumption of a (latent) utility function may appear quite restrictive. However, apart from the fact that this assumption is commonly made also in many other fields, ranging from economic utility theory to information retrieval, it should be mentioned that the user is not supposed to be aware of this function, let alone to reveal it in an explicit form. In fact, we shall not require user feedback in the form of utility degrees for particular objects, i.e., we shall not directly ask for utility degrees $U(\mathbf{a})$. Instead, we shall only ask for the expression of *comparative preferences* of the form "I like object $\mathbf{a}$ more than object $\mathbf{b}$", which is much weaker and arguably more easy to elicit. Information of that kind imposes a constraint on the utility function, namely $U(\mathbf{a}) > U(\mathbf{b})$. From a learning point of view, the basic problem is hence to find a utility function $U(\cdot)$ that is compatible with a set of constraints of this type. As a potential advantage of learning a latent utility function let us also mention that, provided that reasonable assumptions about this function can be made, this is a form of background knowledge that can highly increase the efficacy of the learning process.

The *monotonicity* required for a utility function in our context constitutes an interesting challenge from a machine learning point of view. In fact, recall that all attributes are assumed to be "the more the better" criteria. Therefore,

$$(\mathbf{a} \geq \mathbf{b}) \;\Rightarrow\; (U(\mathbf{a}) \geq U(\mathbf{b})) \tag{1}$$

should hold for all $\mathbf{a}, \mathbf{b} \in \mathbb{O}$, where $\mathbf{a} \geq \mathbf{b}$ means $a_i \geq b_i$ for all $i = 1 \ldots d$. Interestingly, this relatively simple property is not guaranteed by many standard machine learning algorithms. That is, a model that implements a utility function $U(\cdot)$, such as a decision tree, may easily violate the monotonicity property, even if this condition is satisfied by all examples used as training data.

In this paper, we shall proceed from a very simple model, namely a linear utility function

$$U(\mathbf{a}) = \langle \mathbf{w}, \mathbf{a} \rangle = w_1 a_1 + \ldots + w_d a_d, \tag{2}$$

for which monotonicity can easily be guaranteed. In fact, for the model (2), the monotonicity property is equivalent to the non-negativity of the weight vector $\mathbf{w}$, i.e., $w_i \geq 0$ for $i = 1 \ldots d$.

Despite its simplicity, the linear model (2) has a number of merits. For example, it is easily interpretable, as a weight w_i is in direct correspondence with the *importance* of an attribute. Thus, it also allows one to incorporate additional background knowledge, e.g., that attribute A_i is at least twice as important as Attribute A_j, in a convenient way ($w_i > 2w_j$). Finally, the linear model is attractive from a machine learning point of view, as it is amenable to efficient learning algorithms and, moreover, to non-linear extensions via "kernalization" [Schölkopf and Smola, 2001].

Before going into more technical detail, we give a rough outline of our approach as a whole. As a point of departure, we consider a user who is searching a database for an object that satisfies his needs. Roughly speaking, we assume that the user is searching an object with high enough utility (a "top-K" object) but does not necessarily insist on finding the optimal one.

1. The first step consists of computing the skyline $\mathbb{S}$ of a relation constructed by the user (e.g., by an SQL query); in particular, this involves a projection to a subset $A_1 \ldots A_d$ of attributes the user considers relevant. The objects in $\mathbb{S}$ are the potential candidates regarding the user's final choice.
2. Starting with a utility function trained on a small initial training set,[1] the objects $\mathbb{S}$ are sorted according to their degree of utility, and the ranking is presented to the user.
3. The user is asked to inspect the ranking, typically by looking at the top elements. If a suitable object is found, the process terminates.
4. In case the user is not yet satisfied, he will be asked for additional feedback, which in turn is used to expand the training data.
5. The preference model (utility function) is re-trained, an improved ranking is derived, and the process continues with 3.

3.1 The Learning Algorithm

Suppose to be given a set of training data $\mathbb{T}$, which consists of pairwise preferences of the form $\mathbf{a} \succ \mathbf{b}$, where $\mathbf{a}, \mathbf{b} \in \mathbb{S}$. As mentioned previously, the basic learning problem is to find a utility function which is as much as possible in agreement with these preferences and, moreover, satisfies the monotonicity constraint (1). Besides, this function should of course generalize as well as possible beyond these preferences.

Due to the assumption of a linear utility model, our learning task essentially reduces to a binary classification problem: The constraint $U(\mathbf{a}) > U(\mathbf{b})$ induced by a preference $\mathbf{a} \succ \mathbf{b}$ is equivalent to $\langle \mathbf{w}, \mathbf{a} - \mathbf{b} \rangle > 0$ and $\langle \mathbf{w}, \mathbf{b} - \mathbf{a} \rangle < 0$. From a binary classification point of view, $\mathbf{a} - \mathbf{b}$ is hence a positive example and $\mathbf{b} - \mathbf{a}$ is a negative one.

Binary classification is a well-studied problem in machine learning, and a large repertoire of corresponding learning algorithms is available. In our approach, we use a Bayes point machine, which seeks to find the midpoint of the region of intersection of all hyperplanes bisecting the version space into two halves of equal volume. This midpoint, the Bayes point, is known to be approximated by the center of mass of the version space [Herbrich *et al.*, 2001]. More specifically, we use an approximate method proposed in [Herbrich *et al.*, 2001] that makes use of an ensemble of perceptrons trained on permutations of the original training data. This approach has several advantages, notably the following: Firstly, it allows us to incorporate the monotonicity constraint in a relatively simple way. Secondly, as will be detailed in Section 3.2, the ensemble of perceptrons is also useful in connection with the selection of informative queries given to the user.

The simple perceptron algorithm is an error-driven online algorithm that adapts the weight vector $\mathbf{w}$ in an incremental way. To guarantee monotonicity, we simply modify this algorithm as follows: Each time an adaptation of $\mathbf{w}$ produces a negative component $w_i < 0$, this component is simply set to 0. Roughly speaking, the original adaptation is replaced by a "thresholded" adaptation. In its basic form, the perceptron algorithm provably converges after a finite number of iterations, provided the data is linearly separable. Even though we shall not go into further detail here, we note that this property is provably preserved by our modification.

The center of mass of the version space (and hence the Bayes point) is approximated in terms of the average of the perceptrons' weight vectors. Obviously, monotonicity of the single perceptrons implies monotonicity of this approximation.

3.2 Generating Queries

In case the user is not satisfied with the current ranking, our approach envisions a training step in which the model is updated on the basis of additional user feedback. This feedback is derived from a query, in which the user is asked for his preference regarding two objects $\mathbf{a}$ and $\mathbf{b}$ and, correspondingly, consists of a pairwise preference $\mathbf{a} \succ \mathbf{b}$ or $\mathbf{a} \prec \mathbf{b}$ that complements the training set $\mathbb{T}$. The simplest way to generate a query pair $(\mathbf{a}, \mathbf{b})$ is to choose it at random from $\mathbb{S} \times \mathbb{S}$. However, realizing that the information content

[1] A minimal number of training examples is necessary to make the learning problem "well-posed". This number depends on the learning algorithm and the underlying model class. In our experiments, we always started with 5 exemplary pairwise preferences.

of different query pairs can be quite different, the goal of this step should of course be the selection of a maximally informative query, i.e., an example that helps to improve the current model as much as possible. This idea of generating maximally useful examples in a targeted way is the core of *active learning* strategies.[2]

In the literature, various strategies for active learning have been proposed, most of them being heuristic approximations to theoretically justified (though computationally or practically infeasible) methods. Here, we resort to the *Query By Committee* approach [Seung *et al.*, 1992]. Given an ensemble (committee) of models, the idea is to find a query for which the disagreement between the predictions of these models is maximal. Intuitively, a query of that kind corresponds to a "critical" and, therefore, potentially informative example.

In our case, the models are given by the perceptrons involved in the Bayes point approximation. Moreover, two models disagree on a pair $(\mathbf{a}, \mathbf{b}) \in \mathbb{S} \times \mathbb{S}$ if one of them ranks $\mathbf{a}$ ahead of $\mathbf{b}$ and the other one $\mathbf{b}$ ahead of $\mathbf{a}$.

Needless to say, various strategies to find a maximally critical query, i.e., a query for which there is a high disagreement between the committee members, are conceivable. Our current implementation uses the following, relatively simple approach: Let $\mathcal{W} = \{\mathbf{w}_1 \ldots \mathbf{w}_m\}$ be the set of weight vectors of the perceptrons that constitute the committee, respectively. In a first step, the two maximally conflicting models are identified, that is, two weight vectors $\{\mathbf{w}_i, \mathbf{w}_j\} \subset \mathcal{W}$ such that $\|\mathbf{w}_i - \mathbf{w}_j\|$ becomes maximal. Then, the two rankings τ_i and τ_j associated, respectively, with these models are considered. Starting at the top of these ranking, the first conflict pair $(\mathbf{a}, \mathbf{b})$ is found and selected as a query; obviously, this pair is identified by the first position p such that τ_i and τ_j have different objects (namely $\mathbf{a}$ and $\mathbf{b}$, respectively) on this position.[3]

4 Experimental Results

This section presents the results of some experimental studies that we conducted in order to get a first idea of the efficacy of our approach. In this regard, an important question was whether our idea of using machine learning techniques is effective in the sense that it helps to improve the ranking quality relatively quickly, that is, with an acceptable amount of user feedback. Besides, we investigated more specific questions, such as the increase in performance due to the use of a monotone learner and an active learning strategy, and the dependence of the ranking quality (training effort) on the dimensionality of the data.

4.1 Data

Our experiments are based on both artificial and real-world data. The artificial data is repeatedly extracted from a set of 50,000 points, generated at random according to a uniform distribution in the 9-dimensional unit hypercube. First, the skyline of these points is computed. Then, for each experiment, a random weight vector $\mathbf{w}$ is generated (whose entries w_i are independent and uniformly distributed in $[0, 1]$) and the skyline is sorted according the utility degrees defined by this vector.

As real-world data, we used the ranking of the top-200 universities world-wide provided by [O'Leary, 2006]. This data set is particularly suitable due to the following reasons: Firstly, it includes information about the ground-truth, namely the correct ranking. Secondly, the data fits the setting of Skyline computation, as the universities are evaluated in terms of six (normalized) numerical attributes (peer review score, recruiter review score, international faculty score, international students score, staff-to-student ratio, citation-to-staff ratio). Thirdly, the data even meets the assumptions of our linear utility model, as the universities are ranked according to a total score which is a weighted linear combination of the individual scores.

4.2 Quality Measures

To measure the quality of a prediction, a kind of distance function is needed that compares a predicted ranking $\widehat{\tau}$ with the true target ranking τ. As mentioned before, different types of measures can be used for this purpose. To measure the quality of the predicted ranking as a whole, we use the well-known Kendall tau coefficient that essentially calculates the number of pairwise rank inversions, i.e., the number of *discordant* pairs $(\mathbf{a}, \mathbf{b})$:

$$\#\left\{(\mathbf{a}, \mathbf{b}) \mid \tau(\mathbf{a}) < \tau(\mathbf{b}),\, \hat{\tau}(\mathbf{a}) > \hat{\tau}(\mathbf{b})\right\},$$

where $\tau(\mathbf{a})$ is the position of object $\mathbf{a}$ in the ranking τ. More specifically, the Kendall tau coefficient normalizes this number to the interval $[-1, +1]$ such that $+1$ is obtained for identical rankings and -1 in the case of reversed rankings.

To complement the rank correlation, we employed a second measure that is closely related to the recall measure commonly used in information retrieval. Let $\mathcal{K}$ be the set of top-k elements of the ranking τ, that is, $\mathcal{K} = \{\mathbf{a} \in \mathcal{S} \mid \tau(\mathbf{a}) \le k\}$, where k is an integer that is usually small in comparison with the size of the skyline (as a default value, we use $k = 10$); likewise, let $\widehat{\mathcal{K}}$ denote the top-K elements of $\widehat{\tau}$. We then define

$$\mathrm{recall}(\tau, \widehat{\tau}) = \frac{\#(\mathcal{K} \cap \widehat{\mathcal{K}})}{k}. \qquad (3)$$

This measure corresponds to the percentage of true among the predicted top-k elements. It is motivated by the assumption that, typically, a user will only check the top-k elements of a ranking. Thus, the more $\mathcal{K}$ and $\widehat{\mathcal{K}}$ are in agreement, the higher the chance that the user finds a satisfying object.

4.3 Experiment 1

In a first experiment, we applied our approach to the data sets described in Section 4.1. To investigated the effect of ensuring monotonicity of the learner, we used two different versions:

- Monotone: Our method that ensures monotonicity (and uses active learning to generate queries).
- Non-monotone: The non-monotone version of our learning algorithm, that is, a Bayes point machine using standard perceptrons as base learners.

The results are shown in Fig. 2–5. As can be seen, incorporating monotonicity seems to have an important effect on the predictive accuracy of the learner.

[2] The idea of active learning has already been applied in other fields of information retrieval, for example in image retrieval [Tong and Chang, 2001].

[3] In principle, an additional strategy is needed for the case where $\tau_i = \tau_j$. However, even though this problem is theoretically possible, it never occurred in our experiments. Therefore, we omit further details here.

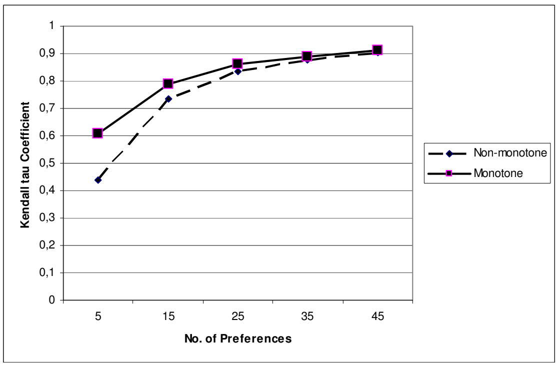


Figure 2: Rank correlation for synthetic data: Monotone vs. non-monotone learning.

4.4 Experiment 2

In a second experiment, we investigated the effect of our active learning strategy. To this end, we compared the results for two different approaches:

- Active: Our method that selects queries in an active way (and ensures monotonicity).
- Non-active: The method obtained by replacing our active learning component by a non-active strategy that selects a query at random, i.e., by randomly selecting two objects $\mathbf{a}, \mathbf{b} \in \mathbb{S}$.

The results are shown in Fig. 6–9. As can be seen, active learning indeed pays off and clearly outperforms the alternative strategy of selecting queries at random.

4.5 Experiment 3

In a third experiment, we investigated the influence of the dimensionality of the data. To this end, we used projections of the original synthetic data set to subspaces of different dimensions. The corresponding performance curves are shown in Fig. 10. Since the dimensionality of the data does have an influence on the size of the skyline and, therefore, on the length of a ranking, the recall measure (3) does not guarantee a fair comparison. Therefore, the performance is only compared in terms of rank correlation.

As expected, the results indicate that the difficulty of the problem increases with the dimensionality of the data. Fortunately, however, the dependence between dimensionality and training effort seems to be "only" linear. This is suggested by the results shown in Fig. 11, where the number of queries needed to reach a certain quality level is plotted against the dimensionality of the data.

5 Extensions

Despite the promising results reported in the previous section, our approach calls for several extensions. In the following, we shall outline some concrete points that we plan to address in future work.

- *More general utility models:* Due to the fact that a ranking is to some extent insensitive toward modifications of the utility model (different models may induce the same or at least similar rankings), sufficiently good rankings may already be obtained with utility models that are only approximately correct. Still, the assumption of a linear utility function is of course quite restrictive, and such models will probably not be flexible enough in practical applications. Therefore, we already conducted first experiments with more general,

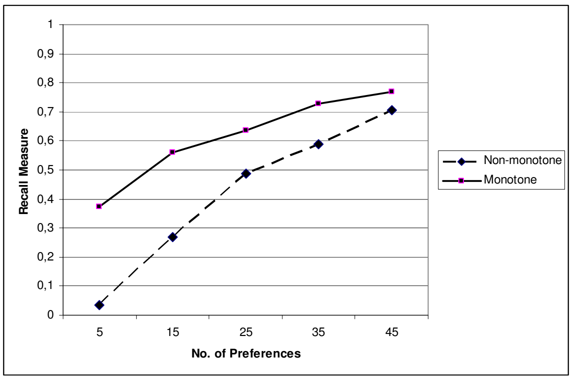


Figure 3: Recall for synthetic data: Monotone vs. non-monotone learning.

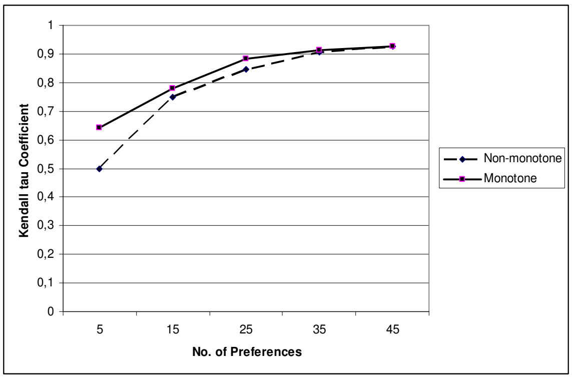


Figure 4: Rank correlation for real data: Monotone vs. non-monotone learning.

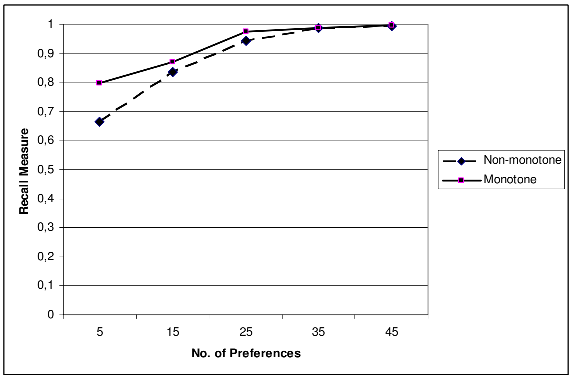


Figure 5: Recall for real data: Monotone vs. non-monotone learning.

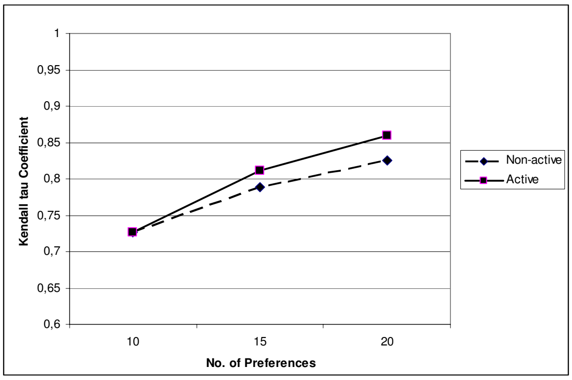


Figure 6: Rank correlation for synthetic data: Active vs. non-active learning.

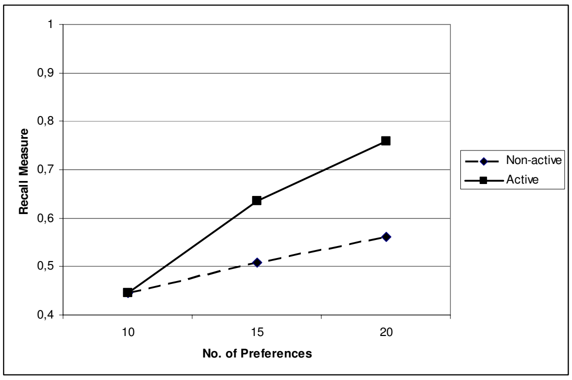


Figure 7: Recall for synthetic data: Active vs. non-active learning.

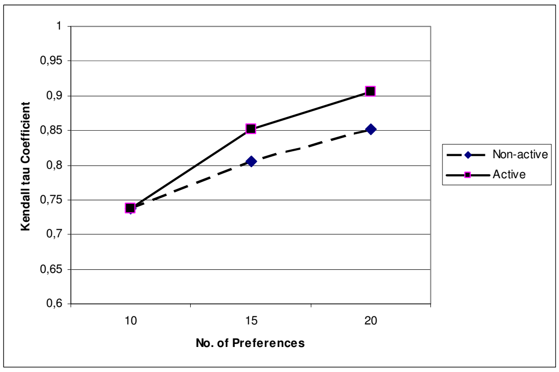


Figure 8: Rank correlation for real data: Active vs. non-active learning.

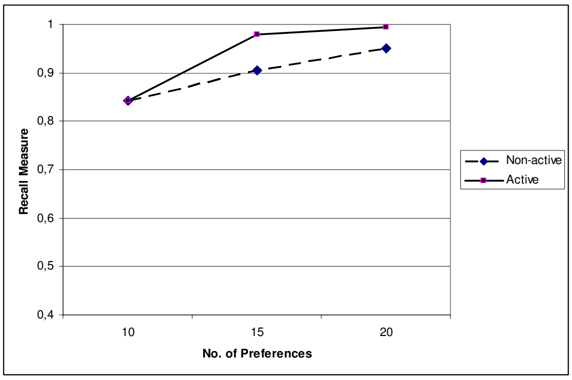


Figure 9: Recall for real data: Active vs. non-active learning.

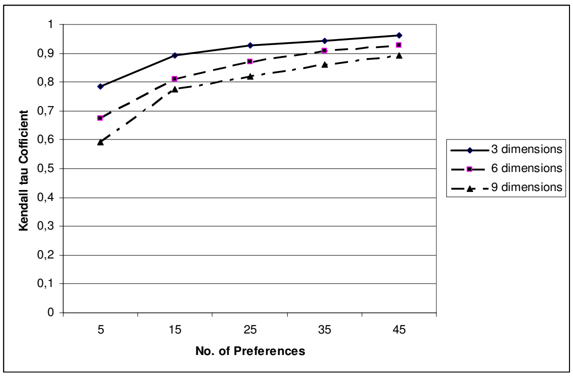


Figure 10: Rank correlation for synthetic data depending on the dimension of the data.

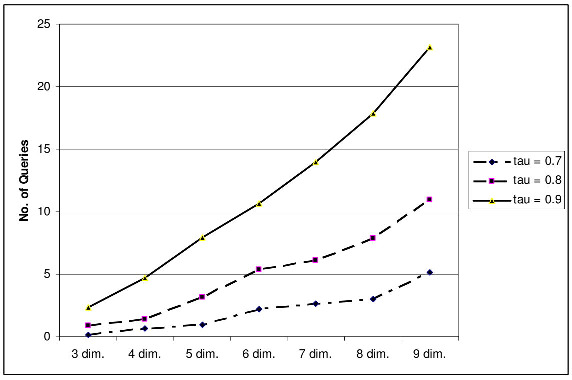


Figure 11: Number of queries needed to reach a certain quality level.

non-linear utility models and obtained results quite comparable to those presented in this paper. The basic idea is to “kernalize” the linear utility function, which is a standard approach in the field of kernel-based learning.

- *Robust learning algorithms:* In practice, user feedback will not always be correct. Instead, the preferences stated by a user may contain errors. From a machine learning point of view, it is therefore important to make learning algorithms tolerant toward “noisy” training data.
- *Learning order relations on attribute domains:* The standard setting of skyline computation assumes all attributes to be criteria, i.e., to have totally ordered domains. In practice, this assumption is quite restrictive and does obviously not hold for attributes such as, say, `color` [Balke and Güntzer, 2005]. From a learning point of view, an obvious idea is to start without prior assumptions, and instead to learn the corresponding order relation from the user’s revealed preferences, e.g., to learn that a user prefers `green` to `red`. In this regard, one may also give up the assumption of a total order and allow for partial orders. Moreover, since preferences for attribute values are clearly not independent, it may become necessary to learn order relations not only in one-dimensional but perhaps also in higher-dimensional subspaces.
- *Integration of skyline computation and ranking*: Until now, skyline computation and ranking are simply carried out in succession, so the integration between them is not very tight. This will change, however, in connection with the aforementioned learning of order relations on attribute domains, since a change of an order relation will also change the dominance relation between objects and, therefore, the skyline. Consequently, skyline computation and ranking will have to be carried out in an alternate rather than a consecutive manner.
- *Deviating from the skyline*: A monotone utility model is in agreement with Pareto-dominance in the sense that the object $\mathbf{a}^* \in \mathbb{O}$ with highest utility is non-dominated and, hence, an element of the skyline. The other way round, however, it is well possible that, according to a given utility model, a user prefers an object $\mathbf{a}$ which is not on the skyline to an object $\mathbf{b}$ which is on the skyline. In principle, this is unimportant as long as we assume that a user is only searching for a

single object. However, if the user is looking for more than one object, it will be useful to include dominated alternatives in the ranking, namely those objects with high utility. And even if the user is searching only a single object, including such objects may speed up the search process, as it increases the likelihood of finding a satisfying alternative. On the other hand, one may think of disregarding those points of the skyline which have a rather low utility, as it is unlikely that such elements will ever move to the top of the ranking. A corresponding pruning strategy would offer another means to increase efficiency. Of course, both measures, the adding of dominated alternatives as well as the pruning of non-dominated ones, presuppose a certain degree of reliability of the current utility model.

- *Valued preferences:* Instead of only asking a user whether he prefers object $\mathbf{a}$ to $\mathbf{b}$ or $\mathbf{b}$ to $\mathbf{a}$, one may allow him to express a *degree of preference*; as a special case, this includes the expression of indifference. An interesting question is whether additional information of that kind can be helpful for producing good rankings.

- *From single users to user groups:* In this paper, we focused on learning the preferences of a single user. In practice, a system will of course be used by more than one person. In this regard, an obvious idea is to exploit the preferences expressed by one user to support another one. In fact, this strategy can help to reduce the feedback requested from the latter user, provided that the preferences are sufficiently similar. A hypothetical transfer of preferences could be done, for example, on the basis of the active user's preferences expressed so far, an idea which is also on the basis of collaborative filtering techniques [Goldberg *et al.*, 1992; Breese *et al.*, 1998]. Another idea is to use clustering techniques in order to find homogeneous groups of users, and to learn utility models that are representative of these groups [Chajewska *et al.*, 2000; 2001].

6 Conclusion

This paper has presented a first study on the combination of skyline computation and machine learning techniques for ranking. The basic motivation is to make skyline queries more user-friendly by providing answers in the form of a ranking of objects, sorted in terms of the user's preferences, instead of returning an unordered and potentially huge answer set.

Our first results are promising in the sense that, apparently, relatively accurate rankings can be produced with an acceptable effort in terms of user feedback. The empirical results are interesting also at a more technical level, since they show that enforcing monotonicity of the learned utility model does indeed improve performance, just like using an active learning strategy to select informative queries.

As outlined in the previous section, we plan to extend our approach in various directions, some of which are currently investigated as part of ongoing work.

References

[Balke and Güntzer, 2005] W. Balke and U. Güntzer. Efficient skyline queries under weak Pareto dominance. In R. Brafman and U. Junker, editors, *Proc. Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 1–6, Edinburgh, Scotland, 2005.

[Borzsonyi *et al.*, 2001] S. Borzsonyi, D. Kossmann, and K. Stocker. The skyline operator. In *IEEE Conf. on Data Engineering*, pages 421–430, Heidelberg, Germany, 2001.

[Breese *et al.*, 1998] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collarborative filtering. In *Proceedings UAI–98*, Madison, WI, 1998.

[Chajewska *et al.*, 2000] U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings AAAI–2000*, pages 363–369, 2000.

[Chajewska *et al.*, 2001] U. Chajewska, D. Koller, and D. Ormoneit. Learning an agent's utility function by observing behavior. In *18th International Conference on Machine Learning, ICML–01*, pages 35–42, 2001.

[Chomicki *et al.*, 2002] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with presorting, 2002.

[Cohen *et al.*, 1999] W.W. Cohen, R.E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10, 1999.

[Detyniecki *et al.*, 2006] M. Detyniecki, JM. Jose, A. Nürnberger, and CJ. van Rijsbergen, editors. *Adaptive Multimedia Retrieval: User, Context, and Feedback*. Number 3877 in LNCS. Springer-Verlag, Heidelberg, 2006.

[Domshlak and Joachims, 2005] Carmel Domshlak and Thorsten Joachims. Unstructuring user preferences: Efficient non-parametric utility revelation. In *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, page 169, Arlington, Virginia, 2005. AUAI Press.

[Fürnkranz and Hüllermeier, 2003] J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. In *Proc.* ECML–2003*, 13th European Conference on Machine Learning*, Cavtat-Dubrovnik, Croatia, September 2003.

[Fürnkranz and Hüllermeier, 2005] J. Fürnkranz and E. Hüllermeier. Preference learning. *Künstliche Intelligenz*, 1/05:60–61, 2005.

[Goldberg *et al.*, 1992] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[Har-Peled *et al.*, 2002] Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification: A new approach to multiclass classification and ranking. Technical report, Champaign, IL, USA, 2002.

[Herbrich *et al.*, 2001] Ralf Herbrich, Thore Graepel, and Colin Campbell. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279, 2001.

[Kendall, 1955] M.G. Kendall. *Rank correlation methods*. Charles Griffin, London, 1955.

[Kossmann *et al.*, 2002] D. Kossmann, F. Ramsak, and S. Rost. Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings VLDB-2002*, Hong Kong, China, 2002.

[O'Leary, 2006] John O'Leary. World university rankings editoral - global vision ensures healthy competition. *The Times Higher Education Supplement*, 2006.

[Papadias *et al.*, 2005] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. Progressive skyline computation in database systems. *ACM Trans. Database Syst.*, 30(1):41–82, 2005.

[Pei *et al.*, 2006] Jian Pei, Yidong Yuan, Xuemin Lin, Wen Jin, Martin Ester, Qing Liu, Wei Wang, Yufei Tao, Jeffrey Xu Yu, and Qing Zhang. Towards multidimensional subspace skyline analysis. *ACM Trans. Database Systems*, 31(4):1335–1381, 2006.

[Robertson and Jones, 1976] SE. Robertson and KS. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[Rochio, 1971] JJ. Rochio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System*, pages 313–323. Prentice Hall, Englewood Cliffs, NJ, 1971.

[Salton and Buckley, 1990] G. Salton and C. Buckley. Improving retrieval performance by retrieval feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[Schölkopf and Smola, 2001] B. Schölkopf and AJ. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.

[Seung *et al.*, 1992] H. S. Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Computational Learning Theory*, pages 287–294, 1992.

[Shen and Zhai, 2005] Xuehua Shen and Chengxiang Zhai. Active feedback in ad-hoc information retrieval. In *Proc. CIKM'2005, ACM Conference on Information and Knowledge Management*, pages 59–66, 2005.

[Shen *et al.*, 2005] Xuehua Shen, Bin Tan, and Chengxiang Zhai. Implicit user modeling for personalized search. In *Proc. CIKM'2005, ACM Conference on Information and Knowledge Management*, pages 824–831, 2005.

[Tan *et al.*, 2001] Kian-Lee Tan, Pin-Kwang Eng, and Beng Chin Ooi. Efficient progressive skyline computation. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 301–310, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[Tong and Chang, 2001] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international Conference on Multimedia*, pages 107–118, Ottawa, Canada, 2001.

Parameter Learning for a Readability Checking Tool

Tim vor der Brück and Johannes Leveling
Intelligent Information and Communication Systems (IICS)
FernUniversität in Hagen (University of Hagen)
58084 Hagen, Germany
{tim.vorderbrueck, johannes.leveling}@fernuni-hagen.de

Abstract

This paper describes the application of machine learning methods to determine parameters for DeLite, a readability checking tool. DeLite pinpoints text segments that are difficult to understand and computes for a given text a global readability score, which is a weighted sum of normalized indicator values. Indicator values are numeric properties derived from linguistic units in the text, such as the distance between a verb and its complements or the number of possible antecedents for a pronoun. Indicators are normalized by means of a derivation of the Fermi function with two parameters. DeLite requires individual parameters for this normalization function and a weight for each indicator to compute the global readability score.

Several experiments to determine these parameters were conducted, using different machine learning approaches. The training data consists of more than 300 user ratings of texts from the municipality domain. The weights for the indicators are learned using two approaches: i) robust regression with linear optimization and ii) an approximative iterative linear regression algorithm.

For evaluation, the computed readability scores are compared to user ratings. The evaluation showed that iterative linear regression yields a smaller square error than robust regression although this method is only approximative. Both methods yield results outperforming a first manual setting, and for both methods, basically the same set of non-zero weights remain.

1 Introduction

Cognitive difficulties for readers are often approximated by a readability function returning a text readability score. The calculation of such a function is typically done in two steps [Flesch, 1948; Chall and Dale, 1995]:

- Determine several indicators for reading difficulty from the surface structure of the text (usually including indicators such as average sentence length and word average length).
- Compute a linear combination of weighted indicator values.

Readability scores have a long history and tradition, especially in English-speaking countries.

The parameters of a readability function may be derived automatically as follows. Given a set of user ratings for a certain text corpus, linear regression can be applied to derive the parameters, minimizing the square difference between the user ratings and the readability score.

A well-known example of a readability function following this schema is the Flesch Reading Ease Score [Flesch, 1948] for English texts, given in equation 1. It is based on computing two indicators from the surface structure, namely the average sentence length (ASL) and the average word length (AWL). For German, similar formulas exist to test the readability of texts (e.g. Amstad [1978]).

$$\begin{aligned} R_{\text{Flesch}} = 206.835 \quad & - \quad (1.015 \cdot \text{ASL}) \\ & - \quad (0.846 \cdot \text{AWL}) \end{aligned} \tag{1}$$

Readability functions of this type have several drawbacks. First, the weights have no intuitive meaning. Therefore, they are difficult to interpret and would be difficult to adjust manually. Second, a large number of indicators in such a formula can easily lead to overfitting, which means that additional work is required to reduce the number of indicators to an optimal set.

For DeLite, our readability checking tool for German texts, a different approach is employed. Before the indicator values are combined they are mapped into the interval [0, 1], which avoids the drawbacks described above and allows a comparison of weights, e.g. for different types of readers (for a detailed description see Section 4).

2 Readability Score and Indicators

DeLite is a readability checking tool for German texts. Its graphical user interface is shown in Figure 2. The readability checking in DeLite relies on a linguistic analysis of text documents with the syntactico-semantic parser WOCADI [Hartrumpf, 2003]. The experiments described here were performed largely on German texts, because the natural language processing tools rely on German resources, e.g. a large German semantic lexicon. WOCADI parses texts and returns their semantic representation, including analysis results corresponding to the morphologic, lexical, syntactic, semantic, and discourse level of linguistic units such as words, phrases, or sentences. Natural language processing results are represented as semantic networks based on the MultiNet paradigm [Helbig, 2006]. These analysis results serve as a basis to derive 47 readability indicators, which represent measurable properties of linguistic units. Indicators are associated with one of the different levels of linguistic analysis given above. Table 1 shows some typical examples of indicators. The readability indicators and

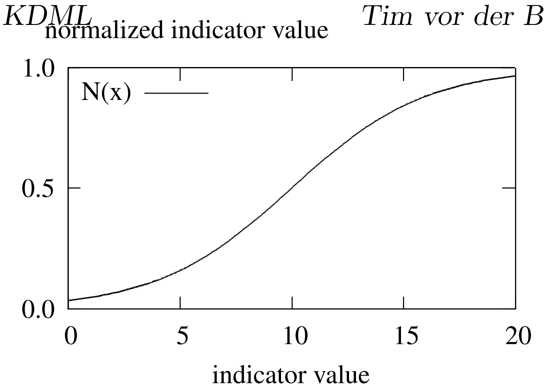


Figure 1: Normalizing function $N(x)$ derived from the Fermi function for $\mu = 10$, $\delta = 3$.

their computation from natural language processing results of the parser are described in more detail by Hartrumpf *et al.* [2006] and Jenge *et al.* [2006].

3 Data Normalization

Unnormalized indicators have vast differences in their value distribution, mean value and variance, e.g. the number of concepts in a compound usually varies between two and five while the number of nodes in a semantic network, representing a sentence, can easily exceed 20. Therefore, indicator values are normalized, mapping them into the interval [0, 1]. A simple method to normalize indicator values would be to employ a linear transformation based on the maximum and minimum values. However, this may not be a reliable solution for several reasons: In new texts, indicator values may exceed the known extreme values. Usually such values are mapped to either zero or one. But in this case, the normalization function will be no longer differentiable on the whole value range, which makes it difficult to apply non-linear optimization techniques like least-squares estimation [Greene, 1993]. Furthermore, this approach becomes very sensitive to outliers.

Many of these difficulties are avoided by the function $N(x)$ used in DeLite (see equation 2), a derivation of the Fermi function. Figure 1 shows the graph of this function for the parameter values $\mu = 10$ and $\delta = 3$.

$$N(x) = \frac{1}{1 + e^{-\frac{x-\mu}{\delta}}} \tag{2}$$

The parameter μ is the location of the 0.5-intercept ($N(\mu) = 0.5$) and δ specifies the incline of the function. For simplicity, its is presumed that indicator values are non-negative and that high unnormalized indicator values correspond to less readability.

One approach to determine the parameters consists of applying a nonlinear optimization to all constituents of the weighted sum and compute both weights and parameters simultaneously. Several (in)equality constraints have to be defined because all weights are expected to be non-negative and normalized to sum up to one. However, estimating more than 140 parameters (three for each indicator) with a constrained nonlinear optimization algorithm is quite difficult and also rather slow.[1] For DeLite, a more efficient approach is chosen, which is also guaranteed to converge. The parameter estimates derived by DeLite could be employed as an initial parameter guess for a nonlinear optimization problem as described above.

The parameter μ_j of the normalization function for a given indicator I_j determines the 0.5-intercept. It usually corresponds to some point near the center of the distribution of the indicator values. Several methods to calculate the parameters μ_j and δ_j of the individual normalization functions were tested, including techniques based on analyzing conditional probabilities, utilizing quantiles, the median, and mean value. Selecting the *mean value* of the distribution for μ_j yielded the smallest error and proved to be quite robust to outliers. The parameter δ_j was obtained by computing the arithmetic mean for solutions of $N_j(x)$ for given values of μ and maximum and minimum values of the indicator value I_j under consideration.

4 Data Combination

In DeLite, a readability score R for a text is calculated as a weighted sum, combining all normalized indicator values v_j. Equation 3 shows the general structure of this function.

$$R = \sum_{j=1}^{m} w_j v_j \tag{3}$$

In the remainder of this paper, normalized weights are assumed, i.e. $w_1 + \ldots + w_m = 1$ and $w_1 \geq 0, \ldots, w_m \geq 0$.

To compute the readability score R, the weight w_j and the normalization parameters μ_j and δ_j have to be determined for each indicator I_j individually. Several machine learning approaches to accomplish this are described and evaluated in Section 5 and Section 6.1.

Note that the indicator weights reflect the importance of each indicator with respect to global readability of a text. If necessary, it would be easy to support manual adjustments, i.e. changing user preferences via the user interface.

There is no need to determine the best set of indicators. After the training period, all indicators with a weight of zero are automatically eliminated from the readability function, i.e. they do not contribute to the readability score. However, all readability indicators – regardless of their weight – are utilized to identify and pinpoint text passages that are difficult to read.

5 Weight Learning

5.1 Problem Description

The parameters of the normalization function are determined as described in Section 3. Thus, to compute the text readability score R, only the indicator weights (w_j) in the weighted sum remain to be found. Basically, two types of machine learning algorithm have been applied to solve such types of problems. These are on the one hand algorithms that depend on a specific probability distribution and on the other hand algorithms which make no such assumption. A method of the first type is for instance the Expectation Maximization algorithm (EM, see Dempster *et al.* [1977]). This algorithm cannot be applied on data where the indicators are highly correlated among each other. A transformation technique like Principal Component Analysis (PCA, see [Jolliffe, 1986]) is necessary in this case to create a new data set with independent indicators.

Since different indicators also have varying probability distributions, an approach of the second type is preferred, which includes regression techniques. Regression can also

[1] Weight learning may have to be repeated several times, e.g. for user groups with different cognitive impairments.

Table 1: Linguistic levels of analysis and corresponding indicators.

Linguistic level	Indicator (German example/English translation, value)
Morphologic	Number of concepts in a compound (‘*Mehrwertsteuererhöhungsdiskussion*’/‘*discussion to increase value added taxes*’, 4)
Lexical	Word frequency class (‘*Stadtverwaltungen*’/‘*municipal administration*’, 36)
Syntactic	Number of syntactic readings of a sentence (‘*Polizei erschoss Mann mit Gewehr*’/‘*Police shot man with gun*’, 2)
Semantic	Number of propositions per sentence (‘*Die Familie besuchte die Tante und übernachtete dort*’/‘*The family visited the aunt and spent the night there*’, 2)
Discourse	Number of reference candidates for a pronoun (‘*Jutta und Maria trafen sich in ihrem Haus*’/‘*Jutta and Maria met in her/their house*’, ≥ 2 for the pronoun ‘*ihrem*’)

be used on highly correlated indicator values without the necessity of any data transformation. However, for most types of regression algorithms the indicator values still have to be linearly independent of each other.

In common optimization algorithms, the optimal weights are determined by minimizing the square error (see equation 4).

$$w_{\text{opt}} = arg\min_{w}(\sum_{i=1}^{n}(y_i - X_i w)^2) \tag{4}$$

The variables given above have the following meanings:

- n: The number of indicators.
- m: The number of rated texts.
- y_i: The average user rating for text i. This value is determined from the global readability ratings by the users. Values of the discrete seven-point Likert scale (Likert [1932], see Section 6.1) are converted into a numeric value between zero and one by a linear transformation. A value of one represents optimal, a value of zero the worst readability.
- X_i : Vector notation for $(x_{i1}, \ldots, x_{im})$. x_{ij} is an indicator value between zero and one for indicator I_j and text i.
- w : Vector notation for $(w_1, \ldots, w_m)$. w_j is the weight for the indicator I_j.

Because all weights are required to be non-negative, simple linear regression cannot be employed. Two alternative approaches are investigated: robust regression with linear optimization (see Section 5.2), and an approximative iterative linear regression based method (see Section 5.3).

5.2 Robust Regression with Linear Optimization

Robust regression leads to estimating parameters by minimizing the sum of the absolute error instead of the square error. The minimization can be achieved via linear optimization, usually applying the Simplex algorithm [Bertsimas and Tsitsiklis, 1997]. This kind of regression is called robust, since it is not as sensitive to outliers as linear regression.

The minimization problem for determining the weights of our readability function can be defined as follows:

$$w_{\text{opt}} = arg\min_{w}(\sum_{i=1}^{n}(|y_i - X_i w|)) \tag{5}$$

In equation 5, $|y_i - X_i w|$ can be replaced by variables z_i, if the constraints $z_i \geq |y_i - X_i w|$ are added (see Bertsimas and Tsitsiklis [1997]). Using the equivalence in equation 6, the optimization problem can be rewritten as shown in equation 7.

$$z_i \geq |y_i - X_i w| \Leftrightarrow z_i \geq (y_i - X_i w) \land z_i \geq -(y_i - X_i w) \tag{6}$$

$$\begin{array}{rcl} & & arg\min_w z_1 + \ldots + z_m \text{ , with} \\ z_i & \geq & x_{i1}w_1 + \ldots + x_{im}w_m - y_i \text{ ,} \\ z_i & \geq & y_i - x_{i1}w_1 - \ldots - x_{im}w_m \end{array} \tag{7}$$

and $i = 1, ..., n$.

This problem consists of linear equations only and can therefore be solved by traditional linear optimization algorithms.

5.3 Iterative Linear Regression

In this section, an approximative solution by using a restricted linear regression problem is discussed. A general restricted linear regression problem is given by equation 8. L contains the coefficients of one or several linear equality restrictions. In addition, the restriction that all weights sum up to one must be represented. Thus, L is set to the vector $(1, ..., 1)$. q represents the values of Lw, in our case $q = (1)$. The regression can be solved by equation 9 (see Greene [1993]).

$$W = \begin{bmatrix} X'X & L' \\ L & 0 \end{bmatrix} \quad u = \begin{bmatrix} X^T y \\ q \end{bmatrix}$$

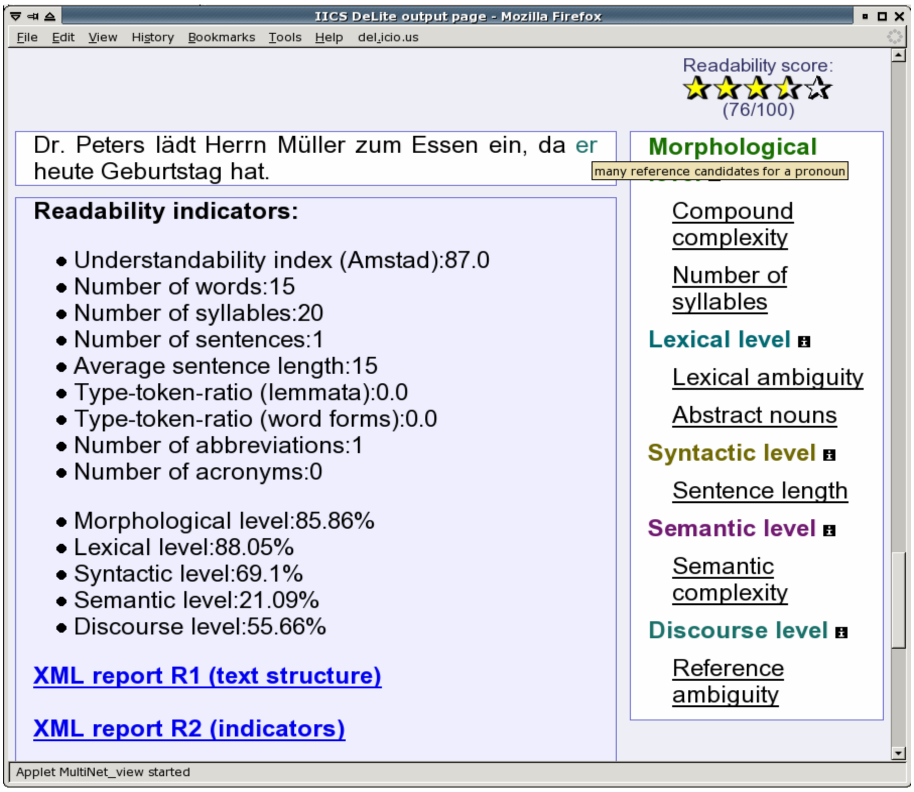
$$W \begin{bmatrix} w \\ \lambda \end{bmatrix} = u \tag{8}$$

$$\begin{bmatrix} w \\ \lambda \end{bmatrix} = W^{-1}u \tag{9}$$

Note that the resulting weights might be negative. Negative weights may have one of the following reasons: First, they might occur if some of the indicators are not correlated with our output (the readability score). Second, they may result if some indicators are strongly correlated among each other. The first problem is avoided by setting indicator weights to zero for indicators which are not correlated with the readability rating R, effectively eliminating the corresponding indicators. The regression described above only has to be applied on the remaining indicators.

The following iterative algorithm is proposed to solve the second problem:

1. Execute the restricted regression as described above.
2. Determine all negative weights and remove the corresponding indicators from the regression model.


IICS DeLite output page - Mozilla Firefox
File Edit View History Bookmarks Tools Help del.icio.us
Applet MultiNet_view started

Readability score:
★★★★☆
(76/100)

Dr. Peters lädt Herrn Müller zum Essen ein, da er heute Geburtstag hat.

Readability indicators:

- Understandability index (Amstad):87.0
- Number of words:15
- Number of syllables:20
- Number of sentences:1
- Average sentence length:15
- Type-token-ratio (lemmata):0.0
- Type-token-ratio (word forms):0.0
- Number of abbreviations:1
- Number of acronyms:0

- Morphological level:85.86%
- Lexical level:88.05%
- Syntactic level:69.1%
- Semantic level:21.09%
- Discourse level:55.66%

XML report R1 (text structure)

XML report R2 (indicators)

Morphological

many reference candidates for a pronoun

Compound complexity

Number of syllables

Lexical level

Lexical ambiguity

Abstract nouns

Syntactic level

Sentence length

Semantic level

Semantic complexity

Discourse level

Reference ambiguity

Figure 2: Graphical user interface of the DeLite readability checking tool.

3. If any negative weights are found, continue with step 1.
4. Return the set of computed weights.

A further improvement of this method may be to remove indicators which are most correlated to indicators with negative weights at every iteration, since very highly correlated (normalized) indicators are nearly exchangeable. However, in the worst case the performance becomes exponential to the number of indicators, since in every iteration several solution paths have to be followed.

6 Evaluation and Implementation

6.1 Evaluation of Parameter Learning

Training data was collected via a web experiment in which participants were asked to answer questions on the readability of given short texts. The participants in the web experiment were asked to judge the global readability of a text. Answers were given on a seven-point Likert scale labelled '*I strongly agree*', '*I agree*', '*I agree somewhat*', '*Undecided*', '*I disagree somewhat*', '*I disagree*', and '*I strongly disagree*'.

The training data for the weight learning approaches consists of user ratings of 500 texts, primarily originating from the municipality domain. The user ratings were obtained from more than 300 participants in a web experiment. The data contains more than 2800 readability ratings.

The evaluation consists of measuring the absolute and square error between the user ratings and the readability scores calculated with weights learned by either iterative linear regression or robust regression. The approximative iterative linear regression method leads to very good results in practice: It always yields a smaller square error than computing scores with the weights found by the robust regression algorithm. Table 2 shows absolute and square error for both methods together with the weights for the remaining indicators. Note that only a small number of the 47 indicators remain for computing the readability score. There are several reasons for this effect, including data sparseness and missing robustness for the semantic analysis of the texts, which causes some indicators to be available for a subset of textual units only. The table shows results for a three-fold cross-validation (CV) as well.

Additionally, the user ratings were compared to the scores obtained from a German variant of the Flesch Reading Ease Score, the Amstad understandability index (Equation 10, see Amstad [1978]).

$$R_{\text{Amstad}} = 180 - \text{ASL} - \text{ASW} \cdot 58.5 \qquad (10)$$

The relative and absolute errors for the Amstad index are 0.203 and 0.245, respectively. The correlation between user ratings and the Amstad index amounts to 0.165. This relatively low correlation may reflect that the Amstad index is not an adequate measure of text understandability, especially concerning texts of our selected municipal domain. DeLite's readability scores have a higher correlation with user ratings, and in comparison, the absolute and square errors are considerably lower (also shown in Table 2). These improvements are mainly due to a larger number of indicators and to indicators resulting from deep natural language processing methods, i.e. indicators on the semantic and discourse level.

In summary, if applied on the training data, the robust regression algorithm yields a lower absolute error than iterative linear regression, while iterative linear regression yields a lower square error. Since the differences between errors from both methods are very small, this assertion cannot necessarily be made if those methods are applied on new data which is also shown by the cross-validation.

Table 2: Weights learned by robust and iterative linear regression.

Normalized weight	Robust regression	Iterative linear regression
	Learning algorithm	
w_1	0.130	0.084
w_2	0.153	0.176
w_3	0.035	0.020
w_4	0.032	0.031
w_5	0.026	0.068
w_6	0.169	0.143
w_7	0.181	0.133
w_8	0.065	0.058
w_9	0.138	0.159
w_{10}	0.010	0.013
w_{11}	0.029	0.086
w_{12}	0.029	0.029
w_{13}	0.003	0.000
Absolute error	0.126	0.127
Square error	0.159	0.157
Absolute error (CV)	0.142	0.141
Square error (CV)	0.177	0.176

Starting with all 47 indicators, only 13 indicators remain as factors of the readability function when using robust regression (twelve if using iterative linear regression). In the DeLite implementation, the iterative linear regression algorithm is more than ten times faster than the robust regression.

6.2 The Readability Checking Tool DeLite

The readability formula as described above is used in the readability checking tool DeLite. DeLite calculates the global readability score and highlights text passages for which the indicator value exceeds a certain threshold. Figure 2 shows the graphical user interface of the readability checking tool. In the upper right corner, the readability score is displayed as a sequence of stars as well as a numerical value. On the top left, the input text is shown, which consists of a relatively simple text with a single sentence (‘*Dr. Peters lädt Herrn Müller zum Essen ein, da er heute Geburtstag hat*’/‘*Dr. Peters invites Mr. Müller to diner because he has birthday today.*’). Below the input text, several readability scores and indicator values are shown, including the Amstad readability index. On the right side, a number of indicators is aligned under the corresponding linguistic level. If selected, the text passages violating readability are highlighted. In the example, the pronoun ‘*er*’ is highlighted, because there are two reference candidates (‘*Dr. Peters*’ and ‘*Mr. Müller*’), which affects a reader’s cognitive ability to understand this text.

7 Conclusion and Outlook

In this paper, novel approaches to determine weights for a readability function were investigated. When using normalization, the importance of each indicator is denoted by its weight, which allows to adapt settings manually. Furthermore, a manual selection of a subset of readability indicators to avoid overfitting is no longer necessary.

Two methods to determine parameters and weights for the readability function were evaluated. The iterative linear regression technique outperforms the linear optimization at the minimization of the average square error. Using the linear regression method, only twelve of a total of 47 indicators remain to be computed (i.e. with a non-zero weight), with linear optimization, 13 indicators have a nonzero weight, including all twelve indicators with non-zero weights determined by linear regression.

For future work, we need to perform significance tests to see if one method performs significantly better than the other. We also intend to integrate nonlinear optimization techniques. Finally, we plan to perform experiments with different user groups sharing the same type of cognitive impairments to see which indicators are affected, i.e. which readability indicators are weighted differently compared to settings for a group of typical users and correspond to the type of cognitive impairment.

Acknowledgments

We wish to thank our colleagues at the FernUniversität in Hagen for their support, especially Sven Hartrumpf, Hermann Helbig and Rainer Osswald. The work on DeLite has been funded by the EU project ‘*Benchmarking Tools and Methods for the Web*’ (BenToWeb, FP6-004275).

References

Toni Amstad. *Wie verständlich sind unsere Zeitungen?* PhD thesis, Universität Zürich, 1978.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Massachusetts, USA, 1997.

Jeanne Chall and Edgar Dale. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books, Brookline, Massachusetts, USA, 1995.

Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1), 1977.

Rudolf Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

William Greene. *Econometric Analysis*. Prentice Hall, Englewood Cliffs, New York, USA, 1993.

Sven Hartrumpf, Hermann Helbig, Johannes Leveling, and Rainer Osswald. An architecture for controlling simple language in web pages. *eMinds: International Journal on Human-Computer Interaction*, 1(2):93–112, 2006.

Sven Hartrumpf. *Hybrid Disambiguation in Natural Language Analysis*. Der Andere Verlag, Osnabrück, Germany, 2003.

Hermann Helbig. *Knowledge Representation and the Semantics of Natural Language*. Springer, Berlin, Germany, 2006.

Constantin Jenge, Sven Hartrumpf, Hermann Helbig, and Rainer Osswald. Automatic control of simple language in web pages. In Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur Karshmer, editors, *Proceedings of the 10th International Conference on Computers Helping People with Special Needs (ICCHP 2006)*, volume 4061 of *Lecture Notes in Computer Science*, pages 207–214, Berlin, Germany, 2006. Springer.

Ian T. Jolliffe. *Principle Component Analysis*. Springer, Berlin, Germany, 1986.

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1932.

WALU — Eine Annotations- und Lern-Umgebung für semantisches Markup in Texten

Andreas Wagner und Marc Rössler
Universität Duisburg-Essen
Computerlinguistik
Lotharstraße 65
D-47048 Duisburg
{andreas.wagner,marc.roessler}@uni-due.de

Abstract

WALU (WIKINGER Annotations- und Lern-Umgebung) ist eine Software zur Annotation und (semi-)automatischen Erkennung von Eigennamen sowie Instanzen anderer semantischer Kategorien in Texten. Ziel der Entwicklung von WALU ist die Realisierung eines komfortablen Werkzeugs, das von Experten unterschiedlichster Domänen ohne computerlinguistische und informatische Vorkenntnisse eingesetzt werden kann. Dies unterscheidet WALU von existierenden Annotations- und Lern-Umgebungen im Bereich Informationsextraktion, die auf andere Tasks zugeschnitten oder multifunktionell ausgelegt sind, was einen erheblichen Konfigurationsaufwand erfordert. WALU ist Teil der kollaborativen Wissens-Infrastruktur, die im eScience-Projekt WIKINGER entwickelt wird. Darüber hinaus ist es als Stand-Alone-Tool einsetzbar. Dieser Beitrag spezifiziert die Design-Prinzipien und aktuell implementierten Funktionalitäten von WALU, gibt einen Überblick über die Pilotdomäne und skizziert laufende Experimente zum semantischen Markup mit maschinellen Lernverfahren.

1 Einleitung

WALU (WIKINGER Annotations- und Lern-Umgebung) ist eine Software zur Annotation und (semi-)automatischen Erkennung von Eigennamen sowie Instanzen anderer semantischer Kategorien in Texten. WALU wird im Rahmen des BMBF-Projekts WIKINGER entwickelt.

Dieser Beitrag gliedert sich wie folgt: Abschnitt 2 skizziert das Projekt WIKINGER. Abschnitt 3 befasst sich Besonderheiten der WIKINGER-Pilotdomäne. Abschnitt 4 diskutiert die sich aus diesem Kontext ergebenden Anforderungen an Eigennamenerkennungsverfahren im Allgemeinen und an das Design des Werkzeugs im Besonderen. Abschnitt 5 beschreibt den aktuellen Entwicklungsstand von WALU und zeigt weitere Perspektiven auf. Abschnitt 6 enthält einen generellen Vergleich von WALU mit anderen einschlägigen Annotations-Tools. Abschnitt 7 schließt mit Zusammenfassung und Ausblick.

2 WIKINGER

Das Projekt WIKINGER (WIKI Next Generation Enhanced Repository), vgl. [Hoeppner *et al.*, 2006; Bröcker *et al.*, 2007], wird im Rahmen der eScience-Initiative des BMBF gefördert (Förderungs-Zeitraum: 10/05–09/08). Ziel des Projekts ist die Entwicklung einer intelligenten Infrastruktur (Plattform) für einen effizienten Austausch wissenschaftlicher Ergebnisse. Diese Infrastruktur wird als semantisches Wiki organisiert, d.h. als Repositorium von Dokumenten und Informationen, die mit Techniken des Semantic Web kodiert und vernetzt sind. Dieses Repositorium wird von den Mitgliedern einer Forschungscommunity online erstellt und modifiziert. Dies ermöglicht den verteilten Auf- und Ausbau eines Informationsnetzes, welches Fachdomänen-Wissen effizient zugänglich macht. In WIKINGER wird zunächst die Pilotdomäne *Katholische Zeitgeschichte* behandelt. Im Sinne der Nachhaltigkeit der entwickelten Infrastruktur ist ihre Wiederverwendbarkeit, d.h. ihre Portierbarkeit auf andere wissenschaftliche Domänen und industrielle Anwendungen, jedoch ein wichtiges Desiderat.

Ein entscheidendes Projektziel ist die Entwicklung von Verfahren zum (semi-)automatischen Aufbau eines solchen semantischen Wissensnetzes aus einschlägigen Dokumenten der jeweiligen Domäne. Hierfür werden in einem ersten Schritt Named Entities (z.B. Personen, Orte, Organisationen) sowie die Vorkommen anderer für die Domäne wesentlicher semantischer Kategorien (z.B. bedeutende Ereignisse) in den Texten erkannt und klassifiziert (d.h. semantisch getaggt). In einem nächsten Schritt werden diese Entitäten mit semi-automatischen Methoden zu semantischen Netzen verknüpft, indem ihre Kookkurenzen und die zugehörigen Kontexte in den Texten analysiert werden. Sowohl die Ergebnisse des semantischen Taggings als auch die daraus resultierenden semantischen Netze unterliegen dem Feedback der Community-Mitglieder, die über das Wiki Zugriff darauf haben und ggf. Korrekturen und Ergänzungen vornehmen können.[1] Das Feedback der Domänenexperten wird zur Verfeinerung der automatischen Extraktionsverfahren eingesetzt.

An WIKINGER sind das Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS), Sankt Augustin, die Computerlinguistik der Universität Duisburg-Essen, Duisburg, sowie die Kommission für Zeitgeschichte (KfZG), Bonn, beteiligt. Die Duisburger Computerlinguistik befasst sich mit der Eigennamen-Erkennung (Named Entity Recognition, NER) bzw. dem semantischen Markup. Ein Kernstück der diesbezüglichen Aktivitäten ist die Entwicklung des Tools WALU.

3 Besonderheiten der Pilotdomäne

Die Fachexperten der KfZG stellen einen Großteil der zu erschließenden Wissensquellen bereit: Die von der

[1] Außerdem besteht die Möglichkeit, neue Daten in das Wiki aufzunehmen.

KfZG herausgegebene „Blaue Reihe" umfasst mehr als 150 Bände zur zeitgeschichtlichen Katholizismusforschung. Alle diese Bände wurden im Rahmen des Projekts digitalisiert. Das domänenspezifische Inventar semantischer Kategorien wurde ausgehend von einem Standard-Inventar für NER [Chinchor, 1998] erarbeitet. Bei der manuellen Annotation exemplarischer Werke kristallisierte sich ein adäquates Inventar heraus, das wie folgt festgelegt wurde:

- GKPE (Geographische Kirchliche/Politische Einheit)
- Ort
- Einrichtung
- Organisation
- Person
- Abgetrennter Namensbestandteil (z.B. in Registern)
- Namenszusatz
- Rolle/Funktion
- Biographisches Ereignis
- Bedeutendes Ereignis
- Datum/Zeit

Da die Zeithistoriker der KfZG im Rahmen der WIKINGER-Aktvitäten insbesondere an biographisch-bibliographischen Informationen über katholische Persönlichkeiten im 19. und 20. Jahrhundert interessiert sind, konzentriert sich die Annotation zunächst auf die sog. Biogramme, d.h. Kurzbiographien, die i.d.R. in Fußnoten präsentiert werden.

Hervorzuheben ist, dass eingebettete Annotationen ausdrücklich vorgesehen sind, wie folgendes Beispiel illustriert:

```
<Rolle>Regens des
<ORG>Regionalseminars
<GKPE>Erfurt</GKPE></ORG></Rolle>
```

Hier wird ein Teil einer Instanz der Kategorie Rolle als Organisation markiert, davon ist wiederum ein Teil als GKPE annotiert.

4 Anforderungen an Annotations-Strategien

Aus dem Ziel der Wiederverwendbarkeit der in WIKINGER entwickelten Infrastruktur, d.h. ihrer Adaptierbarkeit an neue Domänen, ergeben sich wesentliche Anforderungen an die eingesetzten Strategien zur semantischen Annotation. Dies betrifft sowohl die prinzipiellen Verfahren als auch die konkrete Ausgestaltung der verwendeten Werkzeuge.

4.1 Annotationsverfahren

Die Fülle der zu erschließenden Daten, die für eine bestimmte Domäne gewöhnlich vorliegen, macht den Einsatz automatischer und semi-automatischer Annotationsverfahren unerlässlich. Für NER existieren sowohl regelbasierte Verfahren als auch statistische (maschinelle) Lernverfahren, vgl. z.B. entsprechende Beiträge zur Message Understanding Conference (MUC) 7 [Chinchor, 1998]. Im Hinblick auf das Desiderat der Adaptierbarkeit an beliebige Domänen haben maschinelle Lernverfahren signifikante Vorteile. Bei der Anpassung von regelbasierten Verfahren an eine neue Domäne und/oder Sprache ist es i.d.R. erforderlich, den verwendeten Satz von Erkennungsregeln substanziell zu überarbeiten und schlimmstenfalls neu aufzusetzen. Da ein adäquates Regelwerk für praktische Anwendungen sehr komplex ist (sowohl hinsichtlich der Zahl der Regeln als auch hinsichtlich ihrer Interaktion), setzt diese Aufgabe profunde Kenntnisse über die neue Domäne einerseits und die Wirkungsweise und das Zusammenspiel der Erkennungsregeln andererseits voraus. Dies erfordert eine zeitintensive und aufwändige Zusammenarbeit zwischen den jeweiligen Domänenexperten und Computerlinguisten. Dagegen ist für die Anpassung von maschinellen / statistischen Lernverfahren im Wesentlichen die Erstellung neuer Trainingsdaten erforderlich. Diese können durch die Annotierung einschlägiger Texte durch die Domänenexperten gewonnen werden; der hierfür benötigte fachübergreifende Kooperationsaufwand mit Computerlinguisten ist vergleichsweise gering.

Dieser beispielbasierte Ansatz – die für das Lenrverfahren benötigten Informationen werden durch Beispiele übermittelt, nicht durch explizite Regeln und Definitionen – erhöht die Domänen-Adaptivität auch bei dem Schritt, der dem eigentlichen Lernen vorausgeht: der Definition der zu lernenden semantischen Kategorien. Durch die manuelle Annotation verschiedener Instanzen einer Kategorie grenzen die Domänenexperten die Verwendung dieser Kategorie in ihrer Domäne ein (z.B.: Gilt eine Kirchengemeinde als geographische Entität?) und liefern so eine implizite Definition. Dies erspart die aufwändige Erarbeitung einer expliziten Definition. Ebenso ermöglicht das empirische Arbeiten mit den relevanten Texten im Zuge des Annotationsprozesses eine adäquatere Festlegung des Kategorien-Inventars: Neue Kategorien können eingeführt, praktisch irrelevante Kategorien eliminiert und gleichartige Kategorien zusammengefasst werden, wenn während der Annotation ein entsprechender Bedarf festgetellt wird.

Im Sinne der Adaptivität sind also lernbasierte Ansätze zu bevorzugen. Hierbei ist es entscheidend, einerseits domänen-unabhängige Merkmale zu verwenden (z.B. Wortformen und -affixe) sowie andererseits einfache Anbindungsmöglichkeiten domänen-spezifischer Ressourcen (z.B. Listen, s.u.) zu ermöglichen. Besonders interessant sind Verfahren, die versuchen, den initialen Annotationsaufwand möglichst zu minimieren, z.B. durch Active Learning. Jedoch ist in begrenztem Umfang auch der Einsatz regelbasierter Verfahren sinnvoll, wenn sie mit geringem Aufwand gute Erkennungsleistungen erzielen und/oder domänen-übergreifend einsetzbar sind. Beispielsweise bieten sich für die Erkennung von Datums- und Zeitangaben reguläre Ausdrücke an; diese sind vergleichsweise einfach definierbar und domänen- (wenn auch nicht sprach-) unabhängig. Unter das regelbasierte Paradigma fällt insbesondere die Verwendung von allgemeinen oder domänenspezifischen Listen von Kategorie-Instanzen (z.B. Personen oder Bistümern). Diese können entweder aus den annotierten Texten oder aus externen Quellen gewonnen werden. Wir halten die Anwendung von Listen für eine zufriedenstellende Erkennungsrate sowie zur Unterstützung bei der manuellen Annotation für unverzichtbar.

4.2 Werkzeug

Damit die Anpassung des Systems, d.h. die Erstellung von Trainingsdaten und das Trainieren und Anwenden von Lernverfahren, weitestgehend selbstständig von den jeweiligen Domänenexperten durchgeführt werden kann, ist es unabdingbar, hierfür ein geeignetes Werkzeug zur Verfügung zu stellen. Entscheidend ist, dass dieses Tool von Experten unterschiedlichster Domänen ohne computerlinguistische und informatische Vorkenntnisse intuitiv bedienbar ist. Es sollte eine komfortable Oberfläche für den gesamten Annotations- und Lernzyklus bereit stellen. Dazu gehört der einfache Zugriff auf das Repositorium der zu annotierenden Texte, benutzerfreundliche Mechanismen zur manuellen Annotation, eine flexible Verwaltung des Kategorien-Inventars, vielfältige, konfigurierbare Anzeigemöglichkeiten, die nahtlose Integration listenbasierter Annotation sowie die Anbindung von automatischen Annotationsverfahren. Wie in Abschnitt 5 erläutert, ist WALU auf diese Anforderungen zugeschnitten. Im Sinne der Wiederverwendbarkeit ist WALU in Java implementiert und damit Betriebsystem-unabhängig einsetzbar.

5 WALU – aktueller Entwicklungsstand

Die Prioritäten bei der Entwicklung von WALU spiegeln die Erfordernisse in WIKINGER wider. In der bisherigen Phase stand die manuelle Annotation durch die Domänenexperten im Vordergrund. Mit den so gewonnenen Daten werden Listen erstellt und maschinelle Lernverfahren trainiert, die in den kommenden Phasen sukzessive für die automatische Annotation eingesetzt werden. Dementsprechend waren bisher für WALU vorrangig komfortable Mechanismen zur manuellen Annotation, einschließlich der Anbindung listenbasierter Annotatoren, zu realisieren. Jedoch sind bereits erste Experimente mit maschinellen Lernverfahren, die an WALU angebunden wurden, durchgeführt worden.

5.1 Manuelle Annotation

WALU bietet eine komfortable Annotationsoberfläche (vgl. Abbildung 1). Ein Default-Inventar semantischer Kategorien ist vorgegeben[2]; darüber hinaus können, auch textspezifisch, neue Kategorien definiert werden. Eine Annotation erfolgt durch Markieren einer Instanz im Text und der Auswahl der entsprechenden Kategorie (über Kontextmenü, Buttons oder Shortcuts). Die Annotationen werden im Text farblich markiert (jeder Kategorie entspricht eine bestimmte Farbe) sowie in einer separaten Liste neben dem Textfeld angezeigt. Jede Kategorie lässt sich im Text und/oder in der Liste ein- und ausblenden. Eine Annotation kann mit einem Kommentar versehen werden. Neben der Annotation ermöglicht WALU die manuelle Editierung von Dokumenten.

5.2 Einfache (semi-)automatische Annotationsmechanismen

Aus den annotierten Texten extrahiert WALU Listen von Kategorie-Instanzen, die für die automatische Annotation weiterer Vorkommen dieser Instanzen eingesetzt werden können. Eine automatische Annotation wird zunächst als “unchecked” erfasst und dargestellt; eine manuelle Bestätigung führt zum Status “checked”, der zu einer manuellen Annotation äquivalent ist. Die Listen werden bei der Durchsicht der Annotationen interaktiv angepasst, indem beim Löschen einer falschen Annotation auch der entsprechende Listen-Eintrag entfernt werden kann (jedoch nicht muss). So wird die Qualität der Listen sukzessive erhöht.

Ein weiterer Mechanismus zur automatischen Annotation sind reguläre Ausdrücke. Z.Zt. ist ein regulärer Annotator für Datums- und Jahresangaben integriert. Weitere vordefinierte und konfigurierbare reguläre Annotatoren werden hinzukommen.

5.3 Annotation durch maschinelle Lernverfahren

Momentan führen wir vielfältige Experimente zum Einsatz maschineller Lernverfahren in der Pilotdomäne durch. Zum jetzigen Zeitpunkt sind Implementationen zweier einschlägiger Verfahren in WALU integriert und können auf WIKINGER-Daten angewandt werden: MaxEnt (openNLP[3]) und SVM (SVMstruct[4]). Unser Ziel ist es, eine Reihe von Lernverfahren einzubinden, die unabhängig oder in Kombination anwendbar sein sollen, um maximale Performanz zu erreichen. Diese Methoden müssen so eingesetzt werden, dass sie die Akquisition eingebetteter Annotationen erlauben. Dies läuft letztlich darauf hinaus, den zu annotierenden Instanzen (hier: Token) multiuple Klassen zuordnen zu können (z.B. erhält “Erfurt” im Beispiel in Abschnitt 3 die Klassen Rolle, Organisation und GKPE). “Klassische” ML-Verfahren weisen jeder Instanz nur eine Klasse zu. Aus diesem Grund wenden wir mehrere Klassifizierer an, die jeweils unterschiedliche semantische Kategorien zuweisen, und unifizieren die Ergebnisse. Bei ML-Verfahren, die auf binäre Klassifizierer beschränkt sind (z.B. SVM), wird für jede Kategorie ein separater Klassifizierer benötigt. Verfahren ohne diese Einschränkung (z.B. MaxEnt) ermöglichen flexiblere Konfigurationen. Unsere bisherigen Experimente mit MaxEnt-Modellen haben ergeben, dass mit einer Kombination von Klassifizierern, die jeweils eine unterschiedliche Kategorie “ignorieren”, d.h. die, außer der jeweils ignorierten Klasse, alle Kategorien zuweisen, insgesamt bessere Ergebnisse erzielt werden als mit einer Kombination binärer Klassifizierer. Auf Token-Ebene erzielten diese vorläufigen Experimente F-Measures von bis zu 84.6% für Personen, 87,1% für Organisationen, 94,8% für GKPEs und 92,8% für Rollen.

Ein zentrales Kriterium zur Beurteilung eines NER-Systems ist die Anpassungsfähigkeit an eine neue Aufgabe, d.h. an eine neue Domäne und/oder eine neue Sprache. Um diese Eigenschaft zu überprüfen, wurde WALU in einem Experiment für die Erkennung von Named Entities in italienischen Zeitungen trainiert. Dies geschah im Rahmen des NER-Shared Task von EVALITA 2007 (Evaluation of NLP Tools for Italian, [Rössler *et al.*, 2007]). Im Vergleich mit den insgesamt sechs partizipierenden Systemen erreichte WALU die zweitbesten Erkennungsresultate, und das, obwohl keiner der Systementwickler Italienisch spricht.

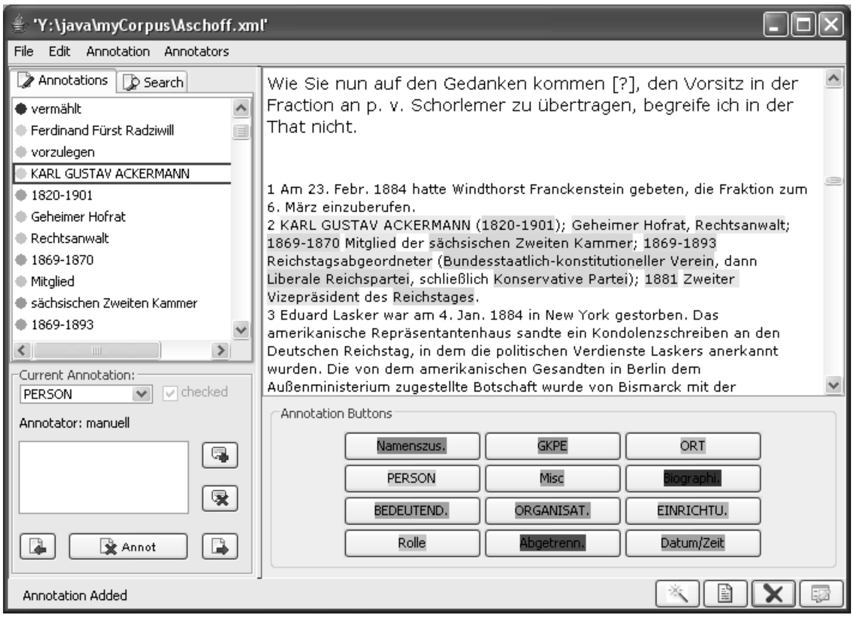
5.4 Qualitätskontrolle

WALU ist mit einer einfachen Zeichenketten-basierten Suchfunktion ausgestattet. Darüber hinaus ist ein spezieller Such- und Sortier-Modus für Annotationen implementiert. In diesem Modus werden die annotierten Entitäten (Types) mit den entsprechenden Kategorien in einer Liste angezeigt (sortiert nach Häufigkeit oder Alphabet). Klickt man auf ein Entitäts-Kategorie-Paar, werden die entsprechenden Vorkommen mit ihren Kontexten im KWIC-Format angezeigt und können direkt bearbeitet werden. So können die einzelnen Annotationen bestätigt, gelöscht oder die Kategorie geändert werden. Dies ermöglicht eine effiziente Kontrolle und Korrektur automatischer Annotationen, insbesondere im Hinblick auf Ambiguitäten. Werden beispielsweise durch listenbasierte Annotation alle Vorkommen von „Singen“ als Ort markiert, können diese Annotationen im Sortier-Modus zusammen aufgelistet werden

[2] Es besteht die Möglichkeit, unabhängige Projekte mit jeweils eigenem Default-Inventar zu definieren.

[3] http://maxent.sourceforge.net/

[4] http://svmlight.joachims.org/svm_struct.html



Wie Sie nun auf den Gedanken kommen [?], den Vorsitz in der Fraction an p. v. Schorlemer zu übertragen, begreife ich in der That nicht.

1 Am 23. Febr. 1884 hatte Windthorst Franckenstein gebeten, die Fraktion zum 6. März einzuberufen.
2 KARL GUSTAV ACKERMANN (1820-1901); Geheimer Hofrat, Rechtsanwalt; 1869-1870 Mitglied der sächsischen Zweiten Kammer; 1869-1893 Reichstagsabgeordneter (Bundesstaatlich-konstitutioneller Verein, dann Liberale Reichspartei, schließlich Konservative Partei); 1881 Zweiter Vizepräsident des Reichstages.
3 Eduard Lasker war am 4. Jan. 1884 in New York gestorben. Das amerikanische Repräsentantenhaus sandte ein Kondolenzschreiben an den Deutschen Reichstag, in dem die politischen Verdienste Laskers anerkannt wurden. Die von dem amerikanischen Gesandten in Berlin dem Außenministerium zugestellte Botschaft wurde von Bismarck mit der

Abbildung 1: WALU-Annotationsoberfläche

und eventuelle Fehler (z.B. Fälle, in denen „Singen“ einem Personennamen oder der – nicht zu annotierenden – Gesangstätigkeit entspricht) korrigiert werden. Zudem ermöglicht diese Überblicks-Darstellung ein velässlicheres Urteil darüber, inwieweit der Listen-Eintrag „Singen“ als Ort überhaupt hilfreich ist.

5.5 Import und Export

Beliebige Dokumente im Textformat können direkt in WALU importiert und annotiert werden. Die Importfunktion beinhaltet interaktive Möglichkeiten zur Auflösung von Mehrspalten-Text. Heuristisch wird zwischen Fließtext, Überschriften, Fußnoten sowie Kopf- und Fußzeilen unterschieden, die jeweils in unterschiedlichen Schriftgrößen dargestellt werden.

Annotierte Dokumente werden in einem XML-Standoff-Format gespeichert. Damit ist es möglich, diese Dokumente auch außerhalb von WALU zu bearbeiten. Details zu diesem Format sind in [Wagner and Rössler, 2007] beschrieben.

Die verschiedenen Datenformate werden in eine interne Repräsentation, den sog. *WARP (WALU Rich Paragraph) Stream*, überführt. Dies ist ein paragraphen-basierter Datenstrom, auf den auch die automatischen Annotatoren zugreifen.

5.6 Kommunikation mit der WIKINGER-Infrastruktur

WALU ist als Stand-Alone-Tool verwendbar, das Daten lokal liest und schreibt. Wie jedoch in Abschnitt 2 dargestellt, bildet WALU zudem einen Teil der in WIKINGER entwickelten verteilten Infrastruktur. Die Einbettung in diese Infrastruktur wird durch spezielle Kommunikations-Mechanismen realisiert. Die in WIKINGER verwendeten Dokumente werden in einem Dokumenten-Repository verwaltet, die zugehörigen Annotationen sowie weitere Informationen im sog. Metadata Repository. Diese Repositories werden auf einem entfernten Server betrieben und sind als gekapselte relationale Datenbanken mit Web-Service-Schnittstellen realisiert. WALU nutzt diese Web-Services zum Laden und Speichern von Daten.

6 Vergleich mit existierenden Tools

WALU ist speziell auf die Annotation semantischer Kategorien ausgerichtet. Wie in Abschnitt 4.2 angeführt, ist es ein entscheidendes Ziel, ein komfortables Werkzeug zu implementieren, das von Experten unterschiedlichster Domänen ohne computerlinguistische und informatische Kenntnisse verwendet werden kann. Dies unterscheidet WALU von existierenden Annotations- und Lern-Umgebungen, die im Bereich Informationsextraktion eingesetzt werden, z.B. GATE [Cunningham *et al.*, 2002], WordFreak [Morton and LaCivita, 2003], MMAX [Müller and Strube, 2001] oder PALinkA [Orasan, 2003]. Diese sind i.d.R. für Benutzer mit (computer-)linguistischem Hintergrund (oder zumindest mit entsprechendem nachhaltigen Support) konzipiert. Dies hat zur Folge, dass sie entweder auf andere, komplexere Tasks zugeschnitten sind (z.B. PALinkA für Diskurs-Annotation) oder in hohem Maße multifunktionell ausgelegt sind (z.B. GATE, WordFreak oder MMAX). Diese Multifunktionalität erlaubt einerseits einen flexiblen, auf komplexe Bedürfnisse zugeschnittenen Einsatz, ist jedoch andererseits mit einem erheblichen Konfigurationsaufwand und einer für „Laien“ mitunter unintuitiven Benutzerführung verbunden.[5]

Zudem ist WALU sowohl als Stand-Alone-Werkzeug als auch als integraler Bestandteil der WIKINGER-Infrastruktur konzipiert. In letzterer Eigenschaft sind spezifische Web-Kommunikationsmodule implementiert.

7 Zusammenfassung und Ausblick

WALU ist ein in der Entwicklung befindliches Werkzeug zum manuellen und (semi-)automatischen semantischen Tagging von Eigennamen und anderen Kategorien. Es wird im Kontext des Projekts WIKINGER entwickelt und bildet einen Teil der dort aufgebauten Infrastruktur. Ebenso ist WALU als Stand-Alone-Tool einsetzbar. Ein primäres Entwicklungsziel ist die nachhaltige Wiederverwendbarkeit, innerhalb und außerhalb der WIKINGER-Plattform.

[5] Z.B. muss in GATE beim Laden einer XML-Datei zusätzlich ein Dokument-Name angegeben werden, oder das System weist einen kryptischen Bezeichner zu.

Dies wirkt sich sowohl auf die Wahl der automatischen Erkennungsverfahren (Beispielbasiertheit) als auch auf die konkrete Gestaltung des Tools (Komfortabilität und intuitive Bedienbarkeit geht vor Multifunktionalität) aus. WALU wird erfolgreich in der WIKINGER-Pilotdomäne eingesetzt.

Die nächsten Entwicklungsschritte konzentrieren sich auf die Anbindung weiterer maschineller Lernverfahren an WALU sowie der Untersuchung unterschiedlicher Kombinationen dieser Verfahren. Denkbar ist sowohl die sequentielle Anwendung verschiedener Methoden, sodass aus der Ausgabe eines Klassifizierers Merkmale für den folgenden Klassifizierer generiert werden, als auch die parallele Anwendung unterschiedlicher Klassifizierer, deren Ergebnisse mit Hilfe eines Voting-Mechanismus zusammengeführt werden. Konkret planen wir die Realisierung einer Schnittstelle zur Weka-Bibliothek [Witten and Eibe, 2005], die eine Reihe interessanter und einschlägiger Verfahren zur Verfügung stellt.

Literatur

[Bröcker *et al.*, 2007] Lars Bröcker, Marc Rössler, Andreas Wagner, et al. WIKINGER - Wiki Next Generation Enhanced Repositories. In *Online Proceedings of the German E-Science Conference*, Baden-Baden, 2007.

[Chinchor, 1998] Nancy A. Chinchor, editor. *Proceedings of the Seventh Message Understanding Conference*, Fairfax, VA, 1998.

[Cunningham *et al.*, 2002] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[Hoeppner *et al.*, 2006] Wolfgang Hoeppner, Marc Rössler, Heinz Ulrich Hoppe, and Nils Malzahn. Globale Forschungsgemeinde. IT-Werkzeuge unterstützen die Vernetzung von Wissen. In *Forum Forschung*, pages 20–23. Universität Duisburg-Essen, 2006.

[Morton and LaCivita, 2003] Thomas Morton and Jeremy LaCivita. WordFreak: an open tool for linguistic annotation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003.

[Müller and Strube, 2001] Christoph Müller and Michael Strube. MMAX: A tool for the annotation of multimodal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Seattle, WA, 2001.

[Orasan, 2003] Constantin Orasan. PALinkA: A highly customisable tool for discourse annotation. In *Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan, 2003.

[Rössler *et al.*, 2007] Marc Rössler, Andreas Wagner, Felix Jungermann, and Wolfgang Hoeppner. Applying WALU to annotate named entities in Italian texts. In *Proceedings of the EVALITA 2007 (Evaluation of NLP Tools for Italian)*, Rome, September 2007.

[Wagner and Rössler, 2007] Andreas Wagner and Marc Rössler. WALU — Eine Annotations- und Lern-Umgebung für semantisches Tagging. In Georg Rehm, Andreas Witt, and Lothar Lemnitzer, editors, *Data Structures for Linguistic Resources and Applications*, pages 263–271. Gunter Narr Verlag, Tübingen, 2007.

[Witten and Eibe, 2005] Ian H. Witten and Frank Eibe. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

Learning to Cope with Critical Situations - A Simulation Model of Cognitive Processes using Multiagent Systems

Régis Newo[1], **Thomas Müller**[2], **Klaus-Dieter Althoff**[1] and **Werner Greve**[3]

[1]University of Hildesheim, Institute of Computer Sciences, Intelligent Information Systems Lab

Email: newo|althoff@iis.uni-hildesheim.de

[2]Email: thomas.mueller@tmtm.de

[3]University of Hildesheim, Institute of psychology, Email: wgreve@rz.uni-hildesheim.de

Abstract

How does someone react when he faces a critical situation in his life? We present in this paper a model for the simulation of people's behaviours in those particular situations. For this purpose, we use some coping strategies developed by researchers in the area of psychology. In our model we mainly consider the interactions between a person concerned and factors like his environment and his own abilities. We plan to implement our model by means of an holonian multi agent system approach, realized by distibuted knowledge based systems with a specific focus on cased-based reasoning technology.

1 Introduction

In our everyday life, we consistently face situations which pose more or less immense challenges. Examples can be the breakup with a partner, the loss of a job, an illness or even the death of a relative. As different as thoses challenges can be, the reactions of the persons who are facing the same kind of challenges can be very different as well. The problem consists in finding out, how someone reacts when he/she faces up a given challenge. The problem being a psychological one, there have been many reasearch groups in psychology working in that direction, beginning in the early 1980s. They developed psychological models and paradigms in order to represent and analyse people's behaviours.

In this paper, we present an agent-based approach for the representation and simulation of human behaviours in critical situations. In the next section, we will first present what has been done in the domain. It covers purely theoretical models from psychologists as well as some developed multiagent systems. We will then present our approach in Section 3 and explain how we intend to implement it. Finally in Section 4 we give a short outlook on relevant future work.

2 Related Work

As mentioned in the last chapter, many researchers in the domain of psychology have tried to find ways in order to understand human behaviour, particularly how a human being reacts when he faces any serious difficulties. They developed theories, software-based models, and simulation approaches for that purpose. As the human way to act is very complex and most of the time ambiguous, developing such software-based systems is not an easy task. It is very important to know first of all which (primary) factors influence human behaviour as well as which processes run during the respective moments and manipulate information. According to the psychologist Lewin, human behaviour can be defined as a function having the person and his environment as arguments (i.e., $B = f(P, U)$, see [Lewin, 1982]). The influence of the environment is thus a very important factor, according to Lewin.

The basis of a person's behaviour is determined by his so-called cognitive processes. Cognitive processes include skills like learning, perception, thinking, reasoning or resolving problems. There have been many approaches in the past in order to model and simulate those skills from a psychological point of view.

2.1 Cognitive Architectures

We will start by presenting some architectures which were developed in the area of social sciences. For a complete and universal definition of a congnitive theory, Newell proposed in [Newell, 1990] that the following mechanisms, which represent the abilities of a system, should be defined:

- Resolution of problems, decision finding
- Multi-purposing, learning, skills
- Perception, motor behaviour
- Language
- Motivation, emotion
- Dreaming, fantasize
- Etc.

However this list is not without controversy, since it is not easy to represent mechanisms like 'dreaming' even in psychology or social sciences. Furthermore, many other researchers are in doubt about the fact that 'dreaming' could really play a role when figuring out the behavioural processes.

In the following, we will particularly consider two types of cognitive architectures. The first type of architecture is called production systems. These systems consist of so-called productions. We can think of those productions as rules, which are used in order to transfer the actual state (of the person) into a goal state (i.e., achieve the solution of a problem). Some examples of such production systems are SOAR (State, Operator and Result) which is based on the "Physical Symbol Systems hypothesis" [Newell and Simon, 1961], ACT-R (Adaptive Control of Thought - Rational) based on the theory of Anderson [Anderson *et al.*, 2004] and EPIC [Kieras, 2004].

The second type of cognitive architecture, we would like to mention here, is the PSI theory [Dörner *et al.*, 2001]. The main difference between this architecture and the one above is the addition of emotional and social components.

It is actually conceivable that some of the behaviour in the architectures described above, up to a certain level of complexity, can be modelled (and thus implemented). Yet the models will not be quite realistic, not only because the human way of acting does not always follow a given guideline or a framework (due to the emotions and the social environment) but also because different situations most of the time imply different stress levels. Furthermore, it is not exactly defined in those architectures how a person will try to solve a problem in burdening situations. What we need here is a better definition of *coping* (resp. coping processes). In the next section we present some models and theories for that aim.

2.2 Coping Processes

We will list here some coping methods. These methods are very important for us, since that is what we need in order to predict the reaction (behaviour) of a person facing a critical situation. There already exist many coping theories. Yet most of them can only be applied for a specific area (e.g. coping strategies for people with a cancer disease, [Taubert, 2003]). Kast provides in her book [Kast, 1989] many definitions and strategies for crises, but they are not really formalized. As a consequence they are not appropriate for use in simulations. That is why we based on the theories and models below for our work.

The first theory was introduced by Richard Lazarus [Lazarus, 1984] and is called *transactional stress theory*. It is based on the assumption that the thoughts and behaviours of each person depend on the characteristics of the actual situation as well as those of the person himself. Examples of the characteristics of a person are his skills, beliefs and moral concepts, whereas some characteristics of a situation are requirements, limitations and resources. A given situation is stressful for a person if the characteristics of that situation overburden or threaten his characteristics. According to Lazarus, stress thus depends on how a person judges the correlation between his characteristics and those of the situation. For that estimation Lazarus differentiates between two types of assessments. The first one (*primary assessment*) is used to identify if a given situation is stressful or not. If a situation is stressful, the second type of assessment (*secondary assessment*) is used to identify which kinds of resources can be used in order to overcome the problem. This leads to the design of a coping strategy which can be divided into basic functions:

- problem-oriented coping and
- emotion-oriented coping.

The second modell is from Sigrun-Heide Filipp (see [Filipp, 1995]) and is based on Lazarus' theory. In Filipp's model, the person is not a passive factor which is influenced by the situation, but instead the active part of the person (e.g., concerning the perception and assessment of a situation) does play an important role. The analysis of a stressful situation is seen as a process flow along a time axis with the following units:

- precursory conditions,
- rival conditions in the person,
- rival conditions in the situation,
- characteristics of the stressful situation,
- analysis process and
- effects of the analysis.

Filipp provides with this list a good source of potential factors of influence.

The most recent theory on coping strategies is from Brandtstädter and Greve [Brandtstädter and Greve, 1994]. It is based on the fact that intentions are a key part of psychological theories of action. Except for knee-jerk or automated behaviours, human actions are motivated by intentions. When somebody faces a critical situation, his actual state strongly differs from his goal state (i.e., his intentions). In order to solve the problem, the person essentially can use one of the following three forms of coping processes:

- *Assimilative processes*: the strategy here is to solve the problem by working directly on the actual state. That is, it is an active art to work through a problem, in which the person uses the available resources in a problem oriented way. The available resources can be the person's own resources or external ones.
- *Accomodative processes*: this strategy is used when the person believes he can not change the actual state (i.e. solve the problem) by himself. He then tries to adapt his goal state such that the discrepancy to the actual state can be diminished.
- *Immunizing processes*: in this case, the person just ignores the discrepancy between the actual state and his goals. He can for example perform actions that disminisches the meaning of the discrepancy.

2.3 Agent-Based Simulation Approaches

Meanwhile there exist many multiagent systems that deal with the simulation of human behaviour. Yet most of them concentrate on the social behaviour between the agents. The most popular are EOS and Sugarscape.
EOS (Evolution of Organized Societies) was the first agent-based simulation system that dealt with the cognitive processes (see [Klügl, 2000]). Sugarscape is another popular rule-based (agent-based) system which focuses on social behaviour [Epstein and Axtell, 1996].
However, we cannot directly reuse both system approaches because they do not deal with coping.

3 SIMOCOSTS

3.1 The Model

We present here our model SIMOCOSTS (SImulation MOdel for COping STrategy Selection) for the simulation of process-based problem solving [Müller, 2006]. The model is based on the psychologycal theories developed by Filipp, Lazarus, Brandstädter and Greve.
One main difference between our simulation approach and other ones consists in the fact fact that all the other view the respective persons as normal agents. However we think that the distinct abilities of the individuals should affect each other. This implies the need of an internal communication between these abilities. Furthermore the quality of the abilities should change (e.g. decrease) in respect with

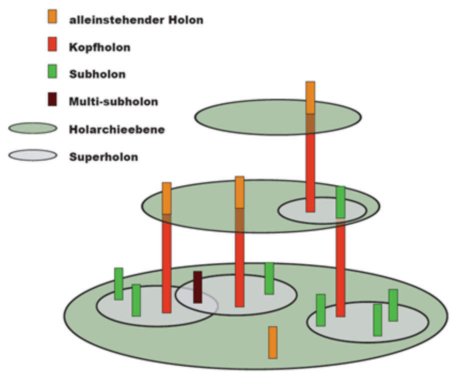


Figure 1: Representation of holons from [Glückselig, 2005].

the time elapsed since its last use in order to be able to represent obivion for example. This leads to the fact that we think the abilities inside an individual should be modelled as agents.
In addition, we also want the individual as a whole to be seen as an agent. That is why we resorted to the 'holonian agent systems'. Holons[1] are agents, which can in turn consists of further agents. With that concept we have some kind of 'recursion', where an agent that is a part of superordinate agent is called a subholon and the superordinate one is called superholon. Some of the advantages of the use of holons are:

- high flexibility,
- good scalability,
- better modelling and
- distinct level(s) of abstraction.

Figure 1 shows a view of holons by [Glückselig, 2005].

For our system, the superholon is the environment and everthing else is an agent or a subholon in it. We then consider conflicts in that environment as specific situations. The environment has some functional units (see Figure 2):

- the context generator,
- the situation generator,
- the communicator and
- the individuals.

The main purpose of the communicator is the communcation between the individuals as well as between individuals and the other components (units). It also has a timer available whose purpose will be explained later.
As our simulation is about human behaviours in critical situations, the modelling of the individual is in our case very important. Each of them has the following subholons, which represent the characteristics of the person:

- Person characteristics,
- Environment characteristics,
- Interface,

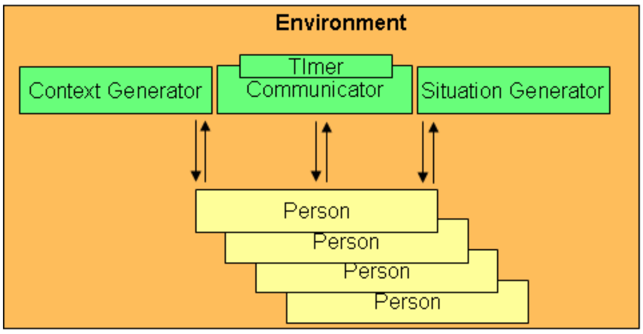


Figure 2: Environment of SIMOCOSTS

- Interpretation and
- Problem solving.

Some person characteristics are:

- The physiological situation, which is important while choosing the (coping) strategy,
- General knowledge, whose modelling can be very complex,
- Skills,
- Goals,
- etc.

The environment characteristics consists of subholons representing the way the persons represents his environment:

- The social environment,
- the material environment and
- the societal environment

The interface between the individual and his environment is composed of two components, an input and an output. The input, also called *affector*, receive messages sent from other individuals or the environment (e.g. sensors). The output (called *effector*) then plays the role of the sender of messages.

The holon 'interpretation' is used to know if a given situation is critical for the individual. It is modelled according to Lazarus' stress theory mentioned in Section 2.2 (i.e. the situation is analyzed in two phases: the primary and secondary assessment).

The problem solving part is the most important part of our simulation, since it comes into operation when a critical situation was detected. It is made up of three subholons

- *Mastering* is used when a problem was analysed that can be simply solved with the person's characteristics,
- *Coping* (strategies) are employed when a critical situation cannot be directly solved by the individual. The strategies used here are from [Brandtstädter and Greve, 1994] and were discussed in Section 2.2.
- *Decompensation*: this represents the last possibilities in the case that the problem could not be solved.

In Figure 3, we can see a detailed representation of our SIMOCOSTS model with focus on the indvidual.

[1] The current (modern) meaning of the term 'holon' was first used by the author Arthur Koestler in some of his many books (e.g., The Ghost in the Machine, 1967)

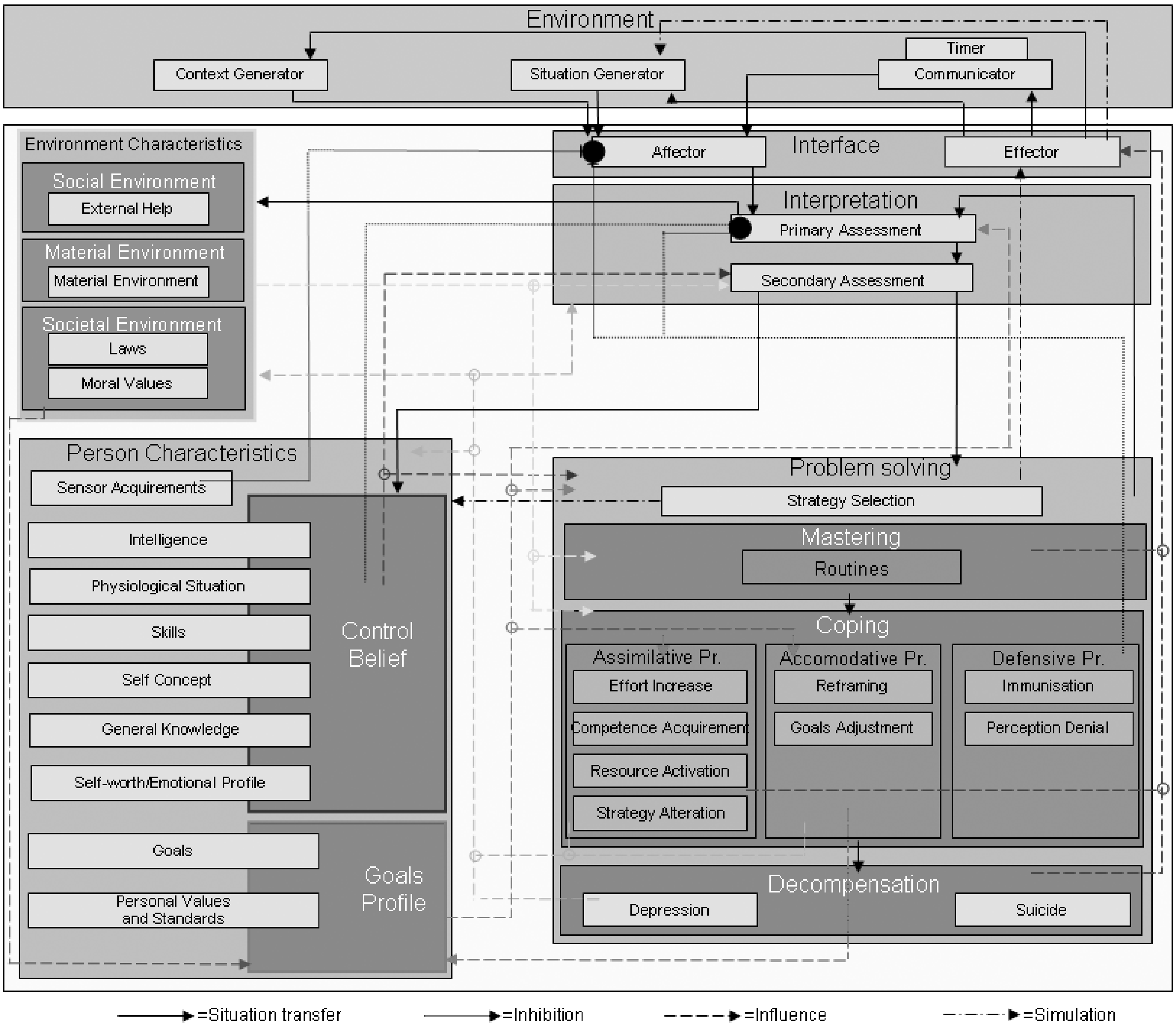


Figure 3: SIMOCOSTS model

3.2 Functionality of the Model

In order to illustrate how our model should work, we present an example here. The critical situation for the person in our example will be the breakup of a partner.
First of all, it is really important to define the situations, goals, and strategies such that they can easily communicate with each other. That means, we should also define the goals such that the properties of a goal can be compared to those of a situation. For the moment, we will define goals as well as situations as a list of weighted facts (with a weight between 1 and 10). Here, we will use natural language to represent these facts. Though we plan to use ontologies later for this purpose. In our example, the original goals of the person are shown in Table 1.

Table 1: Example for the original goals of the person.

Goals		
Goals	**Value**	**Weight**
Start a family	+	7
Self-worth	+	8
Free time	+	3

A situation is rated as burdening, if there exists some kind of discrepancy between its characteristics and those of the given goals. The critial situation that the person faces in our example (generated by the situation generator holon) can be seen in Table 2.

Table 2: Example for the critical situation of the person.

Situation: "Partner wants to break up"		
Affected characteristics/goals	**Value**	**Weight**
Start a family	-	10
Self-worth	-	6
Free time	+	6
To be single	+	10

We see from the table that the value of characteristic "start a family" for example collides with that of a goal. Now, we want to know, with the use of the primary assessment holon, whether the situation is critical or not. In Table 3, we show how the discrepancy is calculated.

The negative value of the discrepancy tells us that the situation is actually critical for the person. The secondary assessment holon should thus try to find out which strategy can be used as a remedy.
The formulation of assimilative strategies is quite complex, because it involves some of the person's characteristics like skills, self concept and general knowledge. With this kind of strategy, the person would for instance do something in

Table 3: Primary Assessment for the example.

Primary Assessment: "Partner wants to break up"	
Affected characteristics/goals	**Discrepancy**
Start a family	-10 * +7 = -70
Self-worth	-6 * +8 = -48
Free time	+6 * +3 = +18
Total discrepancy	**-100**

order to convince the partner not to break up.

With an accommodative strategy, the person would try to adapt to the current situation. Using goals adjustment in our example, the person may say that he/she currently does not want to start a family and that life is much better as a single. We show in Table 4 which goals are adjusted.

Table 4: Adjusted goals.

Strategy: **Goals adjustments**		
Application area: **"Partner wants to break up"**		
Intention: **A Family has many disadvantages,** **it is more comfortable to be single**		
Goal	**Value**	**Weight**
Start a family	+-	0
To be single	+	3
Free Time	+	5

We then recalculate in Table 5 the new discrepancy.

Table 5: Recalculation of the discrepancy after "goals adjustments" for the example.

Recalculation: "Partner wants to break up"	
Affected characteristics/goals	**Discrepancy**
Start a family	-10 * 0 = 0
Self-worth	-6 * +8 = -48
To be single	+10 * +3 = +30
Free time	+6 * +5 = +30
Total discrepancy	**+12**

The used statregy leads to a better value of the discrepancy. The situation is thus no longer burdening, but even positive.

Defensive Strategies inhibit the perception of the actual situation by affecting the affector and the primary assessment agent.

3.3 Implementation Idea

Having presented the model, we now shortly explain how we intend to implement it. The main idea consists in implementing each agent/holon of our model as an expert system (knowledge based system). Thus, the realization will be based on a distributed knowledge-based system architecture (see also [Althoff *et al.*, 2007]). Currently we intend to use the Information Access Suite of empolis for this purpose. As abilities like strategy selection need besides general knowledge also a lot experience in order to work properly, case-based reasoning will be on core technique used (among other inference techniques).

4 Outlook

One important aspect of the research shortly described in this paper is that it is interdisciplinary. Thus, we have to identify the necessary psychological knowledge to build the system and in addition, we need appropriate knowledge representation and processing techniques. In principle we plan for each agent/holon a full-sized knowledge-based system, because the tasks to handled are very challenging. As a consequence, in a first step we will concentrate on the realization of specific subparts of the model. Since we plan to implement the overall model as distributed system we hope that it becomes easier also to involve further domain experts if appropriate. Another idea we want to follow is to use cases available from life coaching situations to detail our model (e.g., Veeser 2001).

References

[Althoff *et al.*, 2007] Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Régis Newo, Meike Reichle, Martin Schaaf, and Karl-Heinz Weis. Collaborative Multi-Expert-Systems – Realizing Knowlegde-Product-Lines with Case Factories and Distributed Learning Systems. In J. Baumeister and D. Seipel, editors, *Accepted for Proc. 3rd Workshop on Knowledge Engineering and Software Engineering (KESE 2007), Osnabrück, Germany*, Berlin, Heidelberg, Paris, 2007. Springer Verlag.

[Anderson *et al.*, 2004] J. R. Anderson, M. D. Byrne, S. Douglas, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1050, 2004.

[Brandtstädter and Greve, 1994] J. Brandtstädter and W. Greve. The aging self: Stabilizing and protective processes. *Developmental Review*, 14:52–80, 1994.

[Dörner *et al.*, 2001] D. Dörner, P. Levi, F. Detje, M.Becht, and D. Lippold. Der agentorientierte, sozionische Ansatz mit PSI. *Sozionik aktuell*, 2, January 2001.

[Epstein and Axtell, 1996] J.M. Epstein and R. Axtell. *Growing Artificial Societies: Social Science from the Bottom Up*. Brookings Institution Press, Washington D.C., 1996.

[Filipp, 1995] Sigrun-Heide Filipp. Ein allgemeines Modell für die Analyse kritischer Lebensereignisse. In *Kritische Lebensereignisse*, Weinheim, 1995. PsychologieVerlagsUnion.

[Glückselig, 2005] S. Glückselig. Holonische Multiagentensimulation. Master's thesis, Universität Würzburg, 2005.

[Kast, 1989] V. Kast. *Der schöpferische Sprung: Vom therapeutischen Umgang mit Krisen*. Deutscher Taschenbuch Verlag, Munich, 1989.

[Kieras, 2004] David E. Kieras. EPIC Architecture Principles of Operation, 2004.

[Klügl, 2000] Franziska Klügl. *Aktivittsbasierte Verhaltensmodellierung und ihre Untersttzung bei Multiagentensimulationen*. PhD thesis, Universität Würzburg, 2000.

[Lazarus, 1984] R.S. Lazarus. *Stress, appraisal and coping*. Springer, New York, 1984.

[Lewin, 1982] Kurt Lewin. *Feldtheorie*, volume 4. Kurt-Lewin Werkausgabe, Bern, 1982.

[Müller, 2006] Thomas Müller. Simulation kognitiver Prozesse mit Multiagentensystemen. Master's thesis, Universität Hildesheim, 2006.

[Newell and Simon, 1961] A. Newell and H.A. Simon. GPS: A programm that simulates human thought. In H. Billing, editor, *Lernende Automaten*, pages 109–124. Oldenbourg, München, 1961.

[Newell, 1990] Allan Newell. *Unified theories of cognition*. Harvard University Press, Cambridge, MA, USA, 1990.

[Taubert, 2003] S. Taubert. *Sinnfindung, Krankheitsverarbeitung und Lebensqualität von Tumorpatienten im perioperativen Verlauf*. PhD thesis, Freie Universität Berlin, 2003.

Adaptive Agents in the Context of Connect Four

Olana Missura
Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
olana.missura@iais.fraunhofer.de

Artificial intelligence (AI) is a part of most computer games and it plays the main role in the players' satisfaction. Games where from the point of view of the player the computer is "dumb" or "too smart" quickly become boring or frustrating.

One of the aspects involved in games' AI is difficulty scaling. For a game to be interesting for a human player, it should be neither too easy, nor too difficult. The conventional method to implement this property is a setting called "level of difficulty" that a player can set up for herself. This doesn't satisfy most people. The skills of a player are changing continuously, not discretely; hence there will be times, when not one of the proposed values for the level of difficulty will match her level. Additionally, level of difficulty setting does not necessarily control the strategy or decision making abilities of a computer opponent, but rather environmental variables. Consider also the fact that generally games require several different skills to play them. The situation when computer can adjust all these skill levels automatically is more user-friendly than offering several settings for a user to set. It is also possible that developers of the game would prefer to keep the information about the specific skills hidden from players so as not to disclose too much data about the mechanics of the game.

In this work we attempt to investigate the two following questions:

1. *To what extent can methods of machine learning be used to develop an online adaptive agent?*
2. *How well does such an adaptive agent perform against humans players?*

Most research on developing online adaptive agents comes from the game developers community. One of the examples is the work of [Hunicke and Chapman, 2004]. In their paper the authors place the task of adaptation on the game engine. It traces and evaluates the player's performance and attempts to adapt the game world in such a way that the game keeps being challenging, but not too difficult for the player. The adaptation is done by adjusting the characteristics of the player's opponents, their numbers and locations, etc.

While this approach certainly has its place in solving the problem of developing games that adapt themselves to the players, modifying the game world is not the answer to the question of creating an online adaptive agent.

An example of an online adaptive agent based on a modified reinforcement learning (RL) approach is presented in the work of [Danzi *et al.*, 2003]. Here the authors use the Q-learning together with a challenge function to build intelligent agents that automatically control the game difficulty level. Q-learning produces a ranking on a set of actions available to the agent in any given state. While the ordinary Q-learning agent chooses the best possible action in every state, the agent designed by the authors uses the challenge function to evaluate the performance of its opponent. Informally, the challenge function tells the agent if the game is too difficult or too easy for the opponent. If it is too difficult (easy), the agent chooses the action that is worse (better) than the one it made before. This approach was evaluated empirically in the context of a fighting game, Knock 'em, against the non-adaptive agents developed by the authors.

The disadvantage of the proposed method is that the agent needs to be trained offline to produce the Q-matrix of the states and evaluations. Depending on the complexity of the game this matrix can be huge. There is no guarantee that while training offline the agent will encounter all possible states, which can lead to it losing the adaptive qualities. While this can be overcome with Q-regression, i.e. replacing the Q-matrix with the Q-function, it does not eliminate the fact that to learn and adapt, the agent needs to play many games.

To answer the questions stated above we needed to address the problem of evaluating a given adaptive agent. For that purpose first we decided on the game that our agents and human players would play: Connect Four. Then we constructed a test environment consisting of two parts. The first part contains four preprogrammed algorithms having four distinct skill levels in Connect Four. The second part provides an environment where human players can play against developed agents and the statistics necessary for evaluation are gathered.

We looked at the problem of creating an online adaptive agent from two different viewpoints: Can the agent estimate the skill level of its opponent and if yes, how can it adapt itself? What can the agent do if such information is not available? As the result of these considerations two distinctly different agents were created and evaluated.

To sum up the main contributions of this work are:

- Design and implementation of a test environment that allows to evaluate the adaptive qualities of a given agent in the context of Connect Four.
- Design and implementation of an adaptive agent for Connect Four based on the MiniMax algorithm. Demonstration of its good adaptive qualities based on the empirical evaluation.
- Design and implementation of an adaptive agent for Connect Four based on the SVM algorithm. Demonstration that its adaptive qualities are not yet satisfactory.

The first of the developed agents, AdaptiveMiniMax, uses a quantative approach to select an appropriate strategy. It evaluates each move made by its opponent and each move available to itself using the same heuristic, builds the ranking on the moves and chooses an appropriate action. To function, AdaptiveMiniMax requires domain knowledge in the form of a heuristic. Without the appropriate heuristic that evaluates the moves or the strategies in the game, the agent would not construct the correct ranking. An additional disadvantage of this method is that AdaptiveMiniMax can play only as good as the underlying MiniMax algorithm. Therefore, it is bound to lose its adaptive properties when playing against opponents who are stronger than MiniMax.

AdaptiveMiniMax showed good performance when playing against the preprogrammed algorithms, adapting well to their respective skill levels, with exception of the optimal algorithm [Allis, 1988], to which it was losing steadily due to the disadvantage mentioned above. In the experiments with the human players data confirming the adaptive qualities of this agent was obtained. There is no correlation between the skill level of a specific player and the percent of games this player won against AdaptiveMiniMax. From the same data it seems that for the majority of players AdaptiveMiniMax chose a strategy that was weaker than the corresponding player's skill level, i.e. the percentage of the games won by humans is mostly greater than 50%. It would be interesting to see how these statistics would change if AdaptiveMiniMax was equipped with memory, that is if it was provided with a way to incorporate the data about the win-loss proportion into the strategy choosing mechanism.

At the moment AdaptiveMiniMax's decision about which move to make is based on the average of the ranking scores of all moves made by its opponent. It is possible that human players make a lot of far from optimal moves in the beginning of the game, when situation on the board is hard to foresee. In this case the average ranking is influenced by these weak moves and it may lead to the apparent weakness of AdaptiveMiniMax when playing against humans. In the future work we would like to investigate how this behaviour can be changed, for example by introducing some kind of discounting scheme, so that the moves made recently have bigger influence on the resulting ranking score than the moves made (relatively) long time ago.

The second agent, SVM Agent, was developed in an attempt to overcome both disadvantages of AdaptiveMiniMax, the need for the good heuristic and the limit on its playing skill. The problem of choosing an appropriate strategy in the game was presented in the context of supervised learning as a binary classification problem, where training data is built from the moves that the agent's opponent made and the agent is making a prediction about which move the opponent would make. An existing implementation of the SVM algorithm [Chang and Lin, 2001] was used to solve this problem. The kernel function was designed to represent the similarities between the pairs of the board states in Connect Four. The detailed description of the problem's formulation and the kernel function can be found in [Missura, 2007].

The experiments were designed to evaluate the SVM Agent's performance using the games played by the preprogrammed algorithms against themselves. As a measure of performance the cross-validation accuracy and the comparison between the SVM Agent's predictions and the moves made in the recorded games were used. The results are dissatisfying. It was to be expected that in the beginning of a game the training set is too small to allow for any good prediction, but there was also hope that as the game progresses and the size of the training set grows, the agent's predictions are going to get better. In reality even though the cross-validation accuracy shows acceptable values (generally around 80%) the predictions that SVM Agent makes are almost always off the mark, and when it does get it right it seems more the case of a random guess succeeding.

Despite these results, we feel that more experimenting can be done with the SVM approach. Adapting the cross-validation procedure to the specifics of our training sets, replacing binary labels with continuous ones, trying out different kernel functions or different types of SVM, or equipping the agent with memory can potentially lead to the improvements. Another way to improve its performance, especially in the beginning of the game, is to provide it with additional domain knowledge, for example of the same kind that AdaptiveMiniMax uses.

References

[Allis, 1988] Victor Allis. A knowledge-based approach of connect-four. The game is solved. Master's thesis, Free University of Amsterdam, October 1988.

[Chang and Lin, 2001] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[Danzi *et al.*, 2003] G. Danzi, A. H. P. Santana, A. W. B. Furtado, A. R. Gouveia, A. Leitão, and G. L. Ramalho. Online adaptation of computer games agents: A reinforcement learning approach. *II Workshop de Jogos e Entretenimento Digital*, pages 105–112, 2003.

[Hunicke and Chapman, 2004] R. Hunicke and V. Chapman. AI for dynamic difficulty adjustment in games. *Proceedings of the Challenges in Game AI Workshop, Nineteenth National Conference on Artificial Intelligence*, 2004.

[Missura, 2007] Olana Missura. Adaptive agents in the context of connect four. Master's thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, August 2007.

Meta-Learning Rule Learning Heuristics

Frederik Janssen and Johannes Fürnkranz
TU Darmstadt, Knowledge Engineering Group
Hochschulstraße 10, D-64289 Darmstadt, Germany
{janssen, juffi}@ke.informatik.tu-darmstadt.de

Abstract

The goal of this paper is to investigate to what extent a rule learning heuristic can be learned from experience. Our basic approach is to learn a large number of rules and record their performance on the test set. Subsequently, we train regression algorithms on predicting the test set performance from training set characteristics. We investigate several variations of this basic scenario, including the question whether it is better to predict the performance of the candidate rule itself or of the resulting final rule. Our experiments on a number of independent evaluation sets show that the learned heuristics outperform standard rule learning heuristics. We also analyze their behavior in coverage space.

1 Introduction

The long-term goal of our research is to understand the properties of a heuristic that will perform well in a broad selection of rule learning algorithms. Although different classification rule learning algorithms use different heuristics, there has not been much work on trying to characterize their behavior. Notable exceptions include [Lavrač *et al.*, 1999], which proposed weighted relative accuracy as a novel heuristic, and [Fürnkranz and Flach, 2005], in which a wide variety of rule evaluation metrics were analyzed and compared by visualizing their behavior in ROC space. There are also some works on comparing properties of association rule evaluation measures (e.g., [Tan *et al.*, 2002]) but these have different requirements than classification rules (e.g., completeness is not an issue there).

Recently, [Janssen and Fürnkranz, 2006] performed an experimental comparison of commonly used rule learning heuristics, and, in particular, found parameter settings for three parametrized heuristics, which performed quite well on a large number of datasets. Interestingly, the authors observed that these values resulted in heuristics with very similar behavior. Nevertheless, the shape of these heuristics was predetermined.

In this work, we take a different road to identifying a good search heuristic for classification rule learning algorithms. The key idea is to meta-learn such a heuristic from experience, without a bias towards existing measures. Consequently, we created a large meta data set (containing information from which we assume that the "true" performance of a rule can be learned) and perform a regression with various methods on it. On this dataset, we learned an evaluation function and used it as a search heuristic inside our implementation of a simple rule learner. We experimented with various options for generating the meta datasets, tried to assess the importance of features, experimented with different meta-learning algorithms, and also tried a setting in which the learner tries to predict the performance of a *complete* rule from its incomplete predecessors.

The ruler learner, which is used for generating the meta data and for evaluating the learned heuristics, is described in Section 2, after which we continue with a brief discussion of rule learning heuristics (Section 3). The meta data generation and the experimental setup is described in Section 4. The main results are presented in Section 5.

2 Rule Learning Algorithm

For the purpose of this empirical study, we implemented a simple *Separate-and-conquer* or *Covering* rule learning algorithm [Fürnkranz, 1999] within the *Weka* machine learning environment [Witten and Frank, 2005]. Both the outer loop (the covering procedure) and the top-down refinement inside the learner are fairly standard. For details about the implementation see [Fürnkranz, 2004; 1999].

Separate-and-conquer rule learning can be divided into two main steps: First, a rule is learned from the training data by a greedy search (the *conquer* step). Second, all examples covered by the learned rule are removed from the data set (the *separate* step). Then, the next rule is learned on the remaining examples. Both steps are repeated as long as positive examples are left in the training set. The refinement procedure, which is used inside the conquer step of the algorithm, returns all possible candidate refinements that can be obtained by adding a single condition to the body of the rule. All refinements are evaluated with a heuristic, and the best rule is selected.

Our implementation continues to greedily refine the current rule until no negative example is covered any more. In this case, the search stops and the best rule encountered during the refinement process is added to the theory. Thus, the best rule is not necessarily the last one searched. A small optimization was to stop the refinement process when no refinement could possibly achieve a better heuristic evaluation than the current best rule, i.e., when a hypothetical refinement that covers all remaining positive examples and no negative example achieves a lower evaluation than the best rule found so far. We used random tie breaking for rules with equal evaluation, and filtered out candidate rules that do not cover any positive examples. Rules were added to the theory until a new rule would not increase the accuracy of the theory on the training set (this is the case when the learned rule covers more negative than positive exam-

Table 1: Search heuristics used in this study

heuristic	function
precision	$\frac{p}{p+n} \sim \frac{p-n}{p+n}$
Laplace	$\frac{p+1}{p+n+2}$
accuracy	$\frac{p+(N-n)}{P+N} \sim p-n$
WRA	$\frac{p+n}{P+N}(\frac{p}{p+n}-\frac{P}{P+N}) \sim \frac{p}{P}-\frac{n}{N}$
correlation	$\frac{p(N-n)-(P-p)n}{\sqrt{PN(p+n)(P-p+N-n)}}$

ples).

We did not use any specific pruning technique, but solely relied on the evaluation of the rules by the used rule learning heuristic. Note, however, that this does not mean that we learn an overfitting theory that is complete and consistent on the training data (i.e., a theory that covers all positive and no negative examples), because many heuristics will prefer impure rules with a high coverage over pure rules with a lower coverage.

3 Rule Learning Heuristics

Numerous heuristics have been provided for inductive rule learning, a general survey can be found in [Fürnkranz, 1999]. Most rule learning heuristics can be seen as functions of the following four arguments:

- P and N: the number of positive/negative examples in the training set
- p and n: the number of positive/negative examples covered by the rule

Examples of heuristics of this type are the commonly used heuristics that are shown in Figure 1. Precision is known to overfit the data, weighted relative accuracy (WRA) [Todorovski *et al.*, 2000] has a tendency to overgeneralize. [Fürnkranz and Flach, 2005] have shown that the Laplace heuristic and its generalization, the m-estimate which is defined by $\frac{p+m\cdot\frac{P}{P+N}}{p+n+m}$, form a trade-off between these two extremes. In [Janssen and Fürnkranz, 2006], a parameter value for the m-estimate was determined that optimizes this trade-off. The correlation heuristic also has a very good overall performance [Fürnkranz, 1994].

As P and N are constant for a given learning problem, these heuristics effectively only differ in the way they trade off completeness (maximizing p) and consistency (minimizing n), and may thus be viewed as a function $h(p,n)$. As a consequence, each rule may be viewed as a point in coverage space, a variant of ROC space that uses the absolute numbers of true positives and false positives as its axes. The preference bias of different heuristics may then be visualized by plotting the respective heuristic values of the rules on top their locations in coverage space, resulting in a 3-dimensional plot $(p, n, h(p,n))$. A good way to view this graph in two dimensions is to plot the *isometrics* of the learning heuristics, i.e., to show contour lines that connect rules with identical heuristic evaluation values. [Fürnkranz and Flach, 2005] have proposed this technique for analyzing the behavior of rule learning heuristics. Another method is to plot both contour lines and the surface of the function which is done in our visualization (cf. Section 5.3).

The goal of our work is to automatically learn such a function $h(p,n)$, which allows to predict the quality of a learned rule. However, note that most of the functions in Table 1 contain some non-linear dependencies between these values. In order to make the task for the learner easier, we will not only characterize a rule by the values p, n, P, and N, but in addition also use the following parameters as input for the meta-learning phase:

- $tpr = \frac{p}{P}$, the true positive rate of the rule
- $fpr = \frac{n}{N}$, the false positive rate of the rule
- $Prior = \frac{P}{P+N}$, the a priori distribution of positive and negative examples
- $prec = \frac{p}{p+n}$, the fraction of positive examples covered by the rule

Thus, we characterize a rule r by an 8-tuple

$$h(r) \leftarrow h(P, N, Prior, p, n, tpr, fpr, prec)$$

Some heuristics use additional components, such as

- l: the length of the rule and
- p' and n': the number of positive and negative examples that are covered by the rule's predecessor.

We will evaluate the utility of taking the rule's length into account. However, as our goal is to find a function that allows to evaluate a rule irrespective of how it has been learned, we will not consider the parameters p' and n'. Note that heuristics like FOIL's information gain [Quinlan, 1996], which include p' and n', may yield different evaluations for the same rule, depending on the order in which its conditions have been added to the rule body.

4 Meta-Learning Scenario

4.1 Definition of the Meta-Learning Task

The key issue for our work is how to define the meta-learning problem. It is helpful to view the rule learning process as a reinforcement learning problem: Each (incomplete) rule is a state, and all possible refinements (e.g., all possible conditions that can be added to the rule) are the actions. The rule-learning agent repeatedly has to pick one of the possible refinements according to their expected utility until it has completed the learning of a rule. After learning a complete theory, the learner receives a reinforcement signal (e.g., the estimated accuracy of the learned theory), which can then be used to adjust the utility function. After a (presumably large) number of learning episodes, the utility function should converge to a heuristic that evaluates a candidate rule with the quality of the *best* rule that can be obtained by refining the candidate rule.

However, for practical purposes this scenario appears to be too complex. [Burges, 2006] has tried a reinforcement learning approach on this problem, but with disappointing results. For this reason, we tried another approach: Each rule is evaluated on a separate test set, in order to get an estimate of its true performance. As a target value, we can either directly use the candidate rule's performance, or we can use the performance of its best refinement (we evaluated both approaches). The latter is described in Section 5.6. In order to assess the performance of a rule, we used its out-of-sample precision, but, again, we have also experimented with other choices.

```
procedure GENERATEMETADATA(TrainSet,TestSet)

# loop until all positive examples are covered
while POSITIVE(TrainSet) $\neq \emptyset$

   # find the best rule
   Rule $\leftarrow$ GREEDYTOPDOWN(TrainSet)

   # stop if it doesn't cover more pos than negs
   if |COVERED(Rule, POSITIVE(Examples))|
      $\leq$ |COVERED(Rule, NEGATIVE(Examples))|
      break

   # loop through all predecessors
   Pred $\leftarrow$ Rule
   repeat

      # record the training and test coverage
      p $\leftarrow$ |COVERED(Rule,POSITIVE(TrainSet))|
      n $\leftarrow$ |COVERED(Rule,NEGATIVE(TrainSet))|
      P $\leftarrow$ |COVERED(Rule,TOTALNEGATIVE(TrainSet))|
      N $\leftarrow$ |COVERED(Rule,TOTALNEGATIVE(TrainSet))|
      l $\leftarrow$LENGTH(Rule)
      $\hat{p}$ $\leftarrow$ |COVERED(Rule,POSITIVE(TestSet))|
      $\hat{n}$ $\leftarrow$ |COVERED(Rule,NEGATIVE(TestSet))|

      # print out meta training instance
      print $P, N, P/(P+N), p, n, p/P, n/N, p/(p+n), l$
      # print out meta target information
      print $\hat{p}, \hat{n}, \hat{p}/(\hat{p}+\hat{n})$

      Pred $\leftarrow$ REMOVELASTCONDITION(Pred)
   until Pred = null

   # remove covered training and test examples
   TrainSet $\leftarrow$ TrainSet $\setminus$ COVERED(Rule,TrainSet)
   TestSet $\leftarrow$ TestSet $\setminus$ COVERED(Rule,TestSet)
```

Figure 1: Algorithm for generating the Meta Data

4.2 Meta Data Generation

As explained above, we try to model the relation of the rule's statistics measured on the training set and its ”true” performance, which is estimated on an independent test set. Therefore, we used the rule learner described above for obtaining the above-mentioned characteristics for each learned rule. These form a training instance in the meta data set. The training signals are the performance parameters of the rule on the test set.

As we want to guide the entire rule learning process, we need to record this information not only for final rules — those that would be used in the final theory — but also for all their predecessors. Therefore all candidate rules which are created during the refinement process are included in the meta data as well. The GENERATEMETADATA procedure described in Figure 1 shows this process in detail.

It should be noted, that we ignored all rules that do not cover any instance on the test data. Our reasons for this were that on the one hand we did not have any training information for this rule (the test precision that we try to model is undefined for these rules), and that on the other hand such rules do not do any harm (they won't have an impact on test set accuracy as they do not classify any example).

To ensure that we obtain a set of rules with varying characteristics, the following parameters were modified:

Datasets: We used 27 datasets with varying characteristics (different number of classes, attributes, instances) from the UCI Repository [Newman *et al.*, 1998].[1]

5x2 Cross-validation: For each dataset, we performed 5 iterations of a 2-fold cross-validation. 2-fold cross-validation was chosen because in this case the training and test sets have equal size, so that we don't have to account for statistical variance in the precision or coverage estimates. We performed five iterations with different random seeds. Note that our primary interest was to obtain a lot of rules which characterize the connection between training set statistics and the test set precision. Therefore, we collected statistics for all rules of all folds.

Classes: For each dataset and each fold, we generated one dataset for each class, treating this class as the positive one and the union of all the others as the negative class. Rules were learned for each of the resulting two-class datasets.

Heuristics: We ran the rule learner several times on the binary datasets, each time using a different search heuristic. We used all of the heuristics of Table 1. The first four form a representative selection of search heuristics with linear ROC space isometrics [Fürnkranz and Flach, 2003], while the correlation heuristic [Fürnkranz, 1994] has non-linear isometrics. These heuristics represent a large variety of learning biases. For example, it is known that *WRA* and *Accuracy* tend to prefer simpler rules with high coverage, whereas *Precision* and *Laplace* show a tendency to learn possibly complex rules with high precision on the training set.

In total, our meta dataset contains $87,380$ examples.

4.3 Regression Methods

We used two different methods for learning functions on the meta data. First, we used a simple *linear regression* using the Akaike criterion [Akaike, 1974] for model selection. A key advantage of this method is that we obtain a simple, easily comprehensible form of the learned heuristic function. Note that the learned function is nevertheless non-linear in the basic dimensions p and n because of the non-linear terms that are used as basic features (e.g., $p/(p+n)$).

Nevertheless, the type of functions that can be learned with linear regression is quite restricted. In order to be able to address a wider class of functions, we used *multilayer perceptron* with back propagation algorithm and sigmoid nodes. We used various sizes of the hidden layer (1, 5, and 10), and trained for one epoch (i.e., we went through the training data once). We have also tried to train the networks with a larger number of epochs, but the results did not improve.

Both algorithms are provided by *Weka* [Witten and Frank, 2005] and were initialized with standard parameters.

[1] anneal, audiology, breast-cancer, cleveland-heart-disease, contact-lenses, credit, glass2, glass, hepatitis, horse-colic, hypothyroid, iris, krkp, labor, lymphography, monk1, monk2, monk3, mushroom, sick-euthyroid, soybean, tic.tac.toe, titanic, vote-1, vote, vowel, wine

Table 2: Accuracies for several methods

method	MAE	Accuracy	# conditions
LinearRegression	0.22	77.43%	117.6
MLP (1 node)	0.28	77.81%	121.3
MLP (5 nodes)	0.27	77.37%	1085.8
MLP (10 nodes)	0.27	77.53%	112.7

4.4 Evaluation methods

Our primary method for evaluating the introduced heuristics is to use these heuristics inside the rule learner. We evaluated the heuristics on 30 UCI data sets[2] which were not used during the training phase. Like the 27 data sets on which the rules for the meta data are induced, these 30 sets have varying characteristics to ensure that our method will perform well under a wide variety of conditions. On each dataset, the rule learner with the learned heuristics was evaluated with one iteration of a 10-fold cross validation. The performance over all sets was then averaged. We also evaluated the length of the theories in terms of number of conditions.

The fit of the learned functions to the target values can also be evaluated in terms of the mean absolute error, again estimated by one iteration of a 10-fold cross validation on the training data.

$$MAE(f') = \frac{1}{m} \sum_{i=0}^{m} |f'(i) - f(i)|$$

with m denotes the number of instances, $f(i)$ the actual value, and $f'(i)$ the predicted value of instance i. The mean absolute error measures the error made by the regression model on unseen data. Therefore it provides no clear insight into the functionalities when it is used as heuristic by the rule learner. Hence, a low mean absolute error on the meta data set does not implicate that the function works good as heuristic (cf. Table 2).

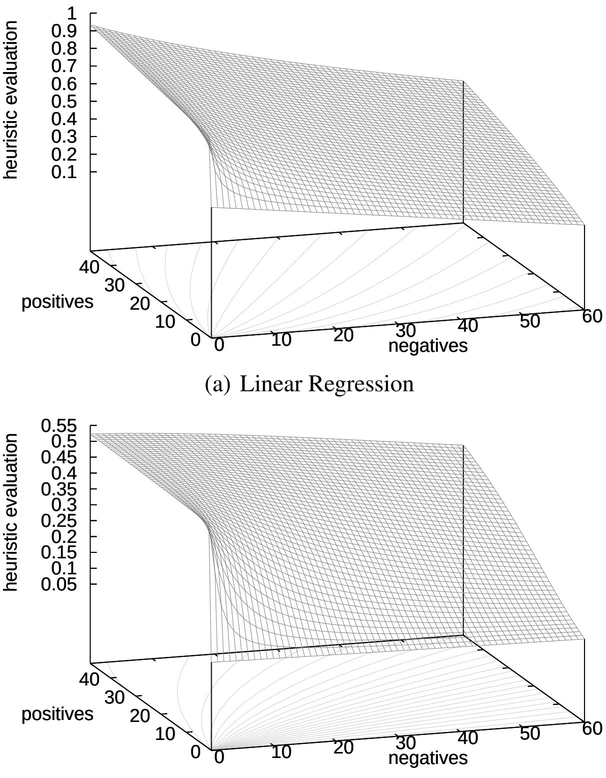
5 Results

5.1 Quantitative Results

In the first experiment, we wanted to see how accurately we can predict the out-of-sample precision of a rule. We trained a linear regression model and a neural network on the eight measurements that we use for characterizing a rule (cf. Section 3) using the precision values measured on the test sets as a target function. Table 2 displays results for the Linear Regression and 3 different neural networks, with different numbers of nodes in the hidden layer. The performances of the three algorithms are quite comparable, with the possible exception of the neural network with 5 nodes in the hidden layer. This induced very large theories (over 1000 conditions on average), and also had a somewhat worse performance in predictive accuracy. As discussed in Section 4.4, a low mean absolute error does not necessarily imply an accurate heuristic as becomes obvious when considering Table 2.

5.2 Coefficients of the Linear Regression

It is interesting to have a look at the learned concepts. Table 3 shows the coefficients of the learned regression model. The most important feature was the *a priori* distribution of the examples in the training data followed by the precision of the rule. Interestingly, while the *tpr* has a non-negligible influence on the result, the *fpr* is practically ignored.



(a) Linear Regression

(b) Neural Network

Figure 2: Isometrics of the two functions (immediate precision)

Both the current coverage of a rule (p and n) and the total example counts of the data (P and N) have comparably low weights This is not that surprising if one keeps in mind that the target value is in the range $[0, 1]$, while the absolute values for p and n are in a much higher range. We nevertheless had included them because we believe that in particular for rules with low coverage, the absolute numbers are more important than their relative fractions. A rule that covers only a single example will typically be bad, irrespective of the size of the original dataset.

In order to see whether we can completely ignore the absolute values, we learned another function which only used $\frac{P}{P+N}$, $^{p}/_{P}$, $^{n}/_{N}$ and $\frac{p}{p+n}$ as input values. The linear regression function trained on this dataset performed insignificantly worse than the one that is computed on the original set (77.43% accuracy vs. 77.20% accuracy). For the neural networks, the performance degradation was somewhat worse.

5.3 Isometrics of the Heuristics

To understand the behavior of the learned heuristics, we follow the framework of [Fürnkranz and Flach, 2005] and analyze their isometrics in ROC or coverage space. Figure 2 shows a 3d-plot of the surface of the learned heuristic in a coverage space with 60x48 examples (the sizes were chosen arbitrarily). The bottom of the graph, shows isometric lines that characterize this surface. The upper part

[2] auto-mpg, autos, balance-scale, balloons, breast-w, breast-w-d, bridges2, colic, colic.ORIG, credit-a, credit-g, diabetes, echocardiogram, flag, hayes-roth, heart-c, heart-h, heart-statlog, house-votes-84, ionosphere, labor-d, lymph, machine, primary-tumor, promoters, segment, solar-flare, sonar, vehicle, zoo

Table 3: Coefficients of the Linear Regression

P	N	$\frac{P}{P+N}$	p	n	$\frac{p}{P}$	$\frac{n}{N}$	$\frac{p}{p+n}$	constant
0.0001	0.0001	0.7485	-0.0001	-0.0009	0.165	0.0	0.3863	0.0267

of the figure displays the isometrics of the heuristic that was learned by linear regression on the data set that used only the relative features (see Section 3). The lower part shows the best-performing neural network (the one that uses only one node in the hidden layer).

Apparently, both functions learn somewhat different heuristics. Although the 3d-surfaces looks fairly similar to each other, the isometric lines reveal that the learned heuristics are, in fact, quite different. Those for the linear regression are like a variant of weighted relative accuracy, but with a different cost model (i.e. false negatives are more costly than false positives). The isometrics for the neural net seems to employ a trade-off similar to those of the F-measure. The shift towards the N-axis is reminiscent of the F-measure (for an illustration see [Janssen and Fürnkranz, 2006]), which tries to correct the undesirable property of precision that all rules that cover no negative examples are evaluated equally, irrespective of the number of positive examples that they cover.

However, both heuristics have a non-linear shape of the isometrics in common, which bends the lines towards the N-axis. Effectively, this encodes a bias towards rules that cover a low number of positive examples (compared to regular precision). This seems to be a desirable property for a heuristic that is used in a covering algorithm, where incompleteness (not covering all positive examples) is less severe than inconsistency (covering some negative examples), because incompleteness can be corrected by subsequent rules, whereas inconsistency cannot.

The isometrics of the linear regression are somewhat curious. In areas of high positive coverage they behave like those of the neural network but not with that strong bend towards the N-axis. The isometrics are symmetric which leads to a similar effect observed at the neural net. Thus, in areas with high negative coverage rules are preferred that cover a low number of negative examples.

5.4 Including the Length of the Rules

Some rule learning algorithms include the length of the learned rule into their evaluation function. For example, the ILP algorithm Progol [Muggleton, 1995] uses $p - n - l$ as a search heuristic for a best-first search. The first part, $p - n$, directly optimizes accuracy (for a fixed dataset, i.e., where the total number of positive (P) and negative (N) examples are fixed), and the length of the rule is used to add an additional bias for simpler rules. However, as longer rules typically cover fewer examples, penalizing the length of a rule may also be considered as another form of bias for high-coverage rules, which could also be expressed by maximizing p (or $p + n$).

In any case, we also experimented with the rule length as an additional parameter. For both, linear regression and neural networks this did not lead to significant changes in the performance of the heuristics. As we will see later on the data set with the 4 features derived in Section 5.2 are sufficient to learn a good heuristic with the Linear Regression.

Table 4: Comparison of various heuristics with training-set (p, n) and predicted $(\hat{p}, \hat{n})$ coverages

heuristic	args	Accuracy	# conditions
Accuracy	(p, n)	75.60%	104.77
	$(\hat{p}, \hat{n})$	75.39%	110.8
Precision	(p, n)	76.22%	129.17
	$(\hat{p}, \hat{n})$	76.53%	30.0
WRA	(p, n)	75.80%	12.13
	$(\hat{p}, \hat{n})$	69.89%	29.97
Laplace	(p, n)	76.89%	118.83
	$(\hat{p}, \hat{n})$	76.80%	246.8
Correlation	(p, n)	77.57%	47.5
	$(\hat{p}, \hat{n})$	58.09%	40.4

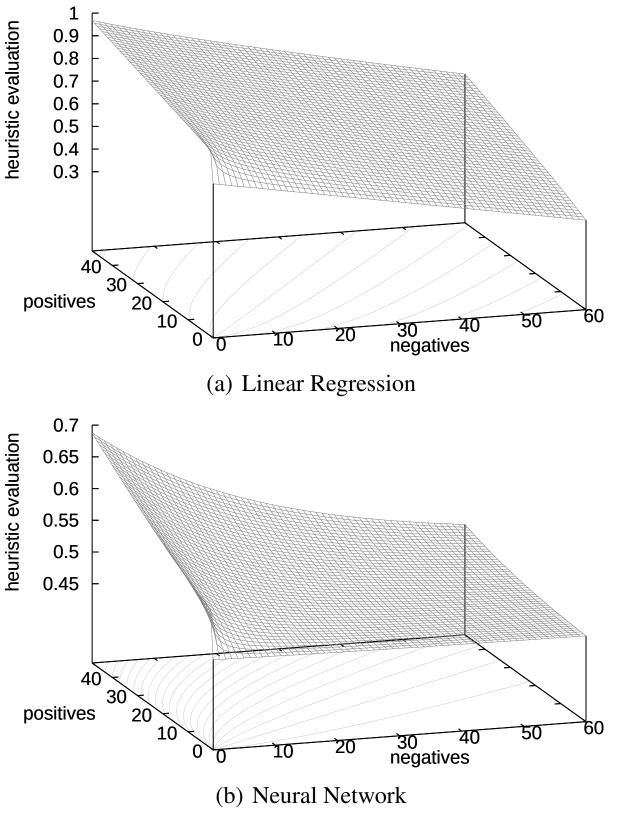
5.5 Predicting Other Heuristics

So far we focused on directly predicting the out-of-sample precision of a rule, assuming that this would be good heuristic for learning a rule set (cf. Section 3). However, this choice was somewhat arbitrary. Ideally, we would like to repeat this experiment with out-of-sample values for all common rule learning heuristics. In order to cut down the number of needed experiments, we decided to directly predict the number of covered positive ($\hat{p}$) and negative ($\hat{n}$) examples. We then can combine the predictions for these values with any standard heuristic h by computing $h(\hat{p}, \hat{n})$ instead of the conventional $h(p, n)$. Note that the heuristic h only gets the predicted coverages ($\hat{p}$ and $\hat{n}$) as new input, all other statistics (e.g., P,N) are still measured on the training set. This is feasible because we designed the experiments so that the training and test set are of equal size, i.e., the values predicted for $\hat{p}$ and $\hat{n}$ are predictions for the number of covered examples on an independent test set of the same size as the training set.

Table 4 compares the performance of various heuristics with measured and predicted coverage values on the 30 test sets. In general, the results are disappointing. For three of the five heuristics, no significant change could be observed, but for *Weighted Relative Accuracy* and the *Correlation* heuristic, the performance degrades substantially.

A rather surprising observation is the complexity of the learned theories. For instance, the heuristic *Precision* produces very simple theories when it is used with the out-of-sample predictions, and, by doing so, increases the predictive accuracy. Apparently, the use of the predicted values of $\hat{p}$ and $\hat{n}$ allows to prevent overfitting, because the predicted positive/negative coverages are never exactly 0 and therefore the overfitting problem observed with *Precision* does not occur any more. The *Laplace* heuristic shows a similar trend, but in this case the predictions result in more complex rules than the original ones.

In summary, it seems that the predictions of both the linear regression and the neural network are not good enough to yield true coverage values on the test set. A closer look at the predicted values reveals that on the one hand both regression methods predict negative coverages and that on the other hand for the region of low coverages (which is



(a) Linear Regression

(b) Neural Network

Figure 3: Isometrics of the functions (final rule precision)

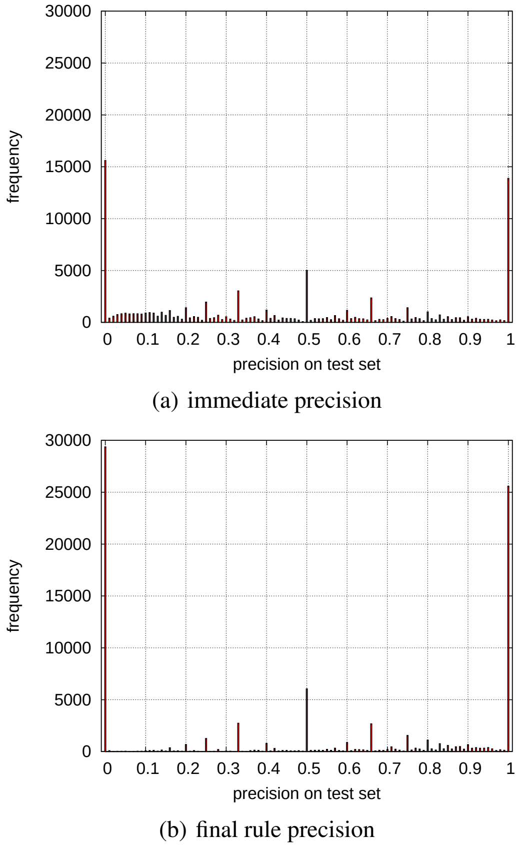
the important one) too optimistic values are predicted (for both the positive and the negative coverage). The acceptable performance is caused by a balancing of the two imprecise predictions (as observed with the two precision-like metrics) or rather by an induced bias which tries to omit the extreme values in the evaluations (which are responsible for overfitting).

Table 4 shows the results for predictions made by the neural network which performs best so far. Interestingly, the same network which was learned on the two meta data sets that include the length, performs consistently better, but the differences are not significant. All other neural networks and the linear regression perform disappointingly due to the above mentioned problems in the predictions.

5.6 Predicting the Value of the Final Rule

Rule learning heuristics typically evaluate the quality of the current, incomplete rule, and use this measure for greedily selecting the best candidate for further refinement. However, as discussed in Section 4.1, if we frame the learning problem as a search problem, a good heuristic should not evaluate a candidate rule with its discriminatory power, but with its potential to be refined into a good final rule. Such a utility function could be learned with a reinforcement learning algorithm, which will learn to predict in each step of the refinement process which refinement is most likely to lead to a good final rule. Unfortunately, in [Burges, 2006] it was pointed out that this approach does not work satisfactorily.

As an alternative, we applied a method which can be interpreted as an ”offline” version of reinforcement learning. We simply assign each candidate rule the precision value of its final rule in one refinement process. As a consequence, in our approach all candidate rules of one refinement process have the same target value, namely the value of the rule that has eventually been selected. Because of the deletion of all final rules that do not cover any example on the test set, we decided to remove all predecessors of such rules as well. Thus, the new meta data set contains only 77,240 examples in total. For us, this seems to be the best way to handle the predecessors because if we want to evaluate them we could only use their immediate precision. But in the current approach we want to use the precision of the final rule, as described above.



(a) immediate precision

(b) final rule precision

Figure 4: Constitution of the meta data

A graphical interpretation of two sample sets can be found in Fig. 4 where, for a given precision, the number of instances were counted. If a candidate rule receives the same evaluation as its final rule (shown in Fig. 4 (b)), the frequency of the worst and the best evaluation increases. Additionally, there are also more rules with a precision of 0.5 too, which are mostly rules that cover a single positive and a single negative example.

All of these large rules often receive perfect training set precision but then are evaluated on the current sample of the test set[3]. On this part of the data, they often do not cover any positive example (i.e., their precision is 0) or no negative example (i.e., their precision is 1). These rules form the majority in the meta data which is preferable because they have the greatest variance among all rules. If a rule covers many examples in the training set, its precision will not differ significantly by that obtained on the test set. The more examples are covered in the training set, the lower is the probability that the rule will cover an entirely different number of examples on the test set. If, for exam-

[3] As shown in Figure 1 the examples covered by the rule were removed from the test set too.

Table 5: Comparison of the induced heuristics with standard ones

heuristic	Accuracy	# conditions
Neural Network	78.37 %	53.97
Linear Regression	77.95 %	95.63
Correlation	77.57 %	47.50
Laplace	76.89 %	118.83
Precision	76.22 %	129.17
WRAcc	75.80 %	12.13
Accuracy	75.60 %	104.77

ple, a rule tests only one attribute and covers n examples on the training set, the probability that this rule will cover a significantly other number than n is small.

Table 5 shows the accuracies of the two heuristics that were learned in this setting, one with a linear regression, and one with a neural network with a single node in the hidden layer. In particular the neural network outperformed the original setting (cf. Table 2).

We also include the results of the 5 standard heuristics that were used to create the meta data. The induced heuristics outperform all of the standard heuristics. The Linear Regression was trained on the meta data set that only contains the 4 most important features which yield the best model. In terms of theory complexity it seems that about 50 conditions in average are necessary to obtain an accurate classifier. Weighted relative accuracy, for example, learns simpler theories (as observed in [Todorovski *et al.*, 2000]), but seems to over-generalize. The neural network classifier performs best with the third-smallest theory.

Figure 3 shows the 3d-surfaces and the isometrics of the two learned heuristics. In comparison to Figure 2, it seems that their curvature towards the N-axis is considerably less steep, which is particularly visible at the points near the P-axis. However, the general shape of the curves seems to remain approximately the same.

6 Conclusion

The most important result of this work is that we have shown that a rule learning heuristic can be learned that outperforms standard heuristics in terms of predictive accuracy on a collection of databases that were not used in the meta-learning phase. Our first results, which used a few obvious features to predict the out-of-sample precision of the current rule, were already en par with the correlation heuristic, which performed best in our experiments. Subsequently, we tried to modify several parameters of this basic setup with mixed results. In particular, predicting the positive and negative coverage of a rule on a test set, and using these predicted coverage values inside the heuristics did not prove to be successful. Also, more complex neural network architectures did not seem to be important, linear regression and neural networks with a single node in the hidden layer performed best. On the other hand, a key result of this work is that evaluating a candidate rule by its potential of being refined into a good final rule works better for learning appropriate heuristics (both the linear regression and the neural network are more precise if they are learned on this type of data).

A visualization of the learned heuristics in coverage space gave some insight into the general functionalities of the learned heuristics. In comparison to heuristics with linear isometrics (such as precision, weighted relative accuracy, and the m-estimate that trades off between those two), the learned heuristics have non-linear isometrics that implement a particularly strong bias towards rules with a low coverage on negative examples. This makes sense for heuristics that will be used in a covering loop, because incompleteness (not covering all positive examples) can be compensated by subsequent rules, whereas inconsistency (covering too many negative examples) cannot. Correlation, the standard heuristic that performed best in our experiments, implements a similar bias [Fürnkranz and Flach, 2005]. Thus, the results of this paper also contribute to our understanding of the desirable behavior of rule-learning heuristics.

Our results may also be viewed in the context of trying to correct overly optimistic training error estimates (resubstitution estimates). In particular, in some of our experiments, we try to directly predict the out-of-sample precision of a rule. This problem has been studied theoretically in [Scheffer, 2001; Mozina *et al.*, 2006]. In other works, it has been addressed empirically. For example [Vapnik *et al.*, 1994] have used empirical data to measure the VC-Dimension of learning machines. [Fürnkranz, 2004] also creates meta data in a quite similar way, and tries to fit various functions to the data. But the focus there is the analysis of the obtained predictions for out-of-sample precision, which is not the key issue in our experiments.

A promising direction for further research is to focus more strongly on the properties of the meta data. An interesting idea is to divide the learning problem into separate smaller problems by learning different models on coverage intervals. Thus, these models could be learned for rules which cover few, medium and many examples. Then, depending on the coverage of the current rule, the corresponding model can be used. Another direction is to focus more on the regression methods. Hence, a parameter optimization of the neural network or the usage of a Support Vector Machine will eventually yield to better results.

References

[Akaike, 1974] H. Akaike. A new look at the statistical model selection. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[Burges, 2006] Sven Burges. Meta-Lernen einer Evaluierungs-Funktion für einen Regel-Lerner, December 2006. Master Thesis.

[Fürnkranz and Flach, 2003] Johannes Fürnkranz and Peter A. Flach. An Analysis of Rule Evaluation Metrics. In *Proceedings 20th International Conference on Machine Learning (ICML'03)*, pages 202–209. AAAI Press, January 2003.

[Fürnkranz and Flach, 2005] Johannes Fürnkranz and Peter A. Flach. ROC 'n' Rule Learning - Towards a Better Understanding of Covering Algorithms. *Machine Learning*, 58(1):39–77, January 2005.

[Fürnkranz, 1994] Johannes Fürnkranz. FOSSIL: A Robust Relational Learner. *Lecture Notes in Computer Science*, 784:122–137, 1994.

[Fürnkranz, 1999] Johannes Fürnkranz. Separate-and-Conquer Rule Learning. *Artificial Intelligence Review*, 13(1):3–54, February 1999.

[Fürnkranz, 2004] Johannes Fürnkranz. Modeling rule precision. In *LWA*, pages 147–154, 2004.

[Janssen and Fürnkranz, 2006] Frederik Janssen and Johannes Fürnkranz. On trading off consistency and coverage in inductive rule learning. In *LWA*, pages 306–313, 2006.

[Lavrač *et al.*, 1999] Nada Lavrač, Peter Flach, and Blaz Zupan. Rule evaluation measures: A unifying view. In S. Džeroski and P. Flach, editors, *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99)*, pages 174–185. Springer-Verlag, 1999.

[Mozina *et al.*, 2006] Martin Mozina, Janez Demsar, Jure Zabkar, and Ivan Bratko. Why is rule learning optimistic and how to correct it. In *Machine Learning: ECML 2006, 17th European Conference on Machine Learning*, pages 330–340, 2006.

[Muggleton, 1995] Stephen Muggleton. Inverse entailment and progol. *New Generation Comput.*, 13(3&4):245–286, 1995.

[Newman *et al.*, 1998] D.J. Newman, C.L. Blake, S. Hettich, and C.J. Merz. UCI Repository of Machine Learning databases, 1998.

[Quinlan, 1996] J.R. Quinlan. Learning First-Order Definitions of Functions. *Journal of Artificial Intelligence Research*, 5:139–161, 1996.

[Scheffer, 2001] Tobias Scheffer. Finding association rules that trade support optimally against confidence. In *Principles of Data Mining and Knowledge Discovery, 5th European Conference, PKDD 2001*, pages 424–435, 2001.

[Tan *et al.*, 2002] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *KDD*, pages 32–41, 2002.

[Todorovski *et al.*, 2000] Ljupco Todorovski, Peter Flach, and Nada Lavrac. Predictive performance of weighted relative accuracy. In Djamel A. Zighed, Jan Komorowski, and Jan Zytkow, editors, *4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000)*, pages 255–264. Springer-Verlag, September 2000.

[Vapnik *et al.*, 1994] Vladimir Vapnik, Esther Levin, and Yann Le Cun. Measuring the VC-dimension of a learning machine. *Neural Computation*, 6(5):851–876, 1994.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data Mining — Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, 2nd edition, 2005.

Adaptive Optimierung des Prefetch-Verhaltens bei objektorientierten Multi-Tier Client-Server-Systemen

Matthias Hofmann **Arno Klein** **Gabriella Kókai**
SCHEMA Electronic Documentation Solutions GmbH
Lehrstuhls für Programmiersysteme
Friedrich-Alexander-Universität Erlangen-Nürnberg
{matthias.hofmann,arno.klein}@schema.de, kokai@informatik.uni-erlangen.de

Abstract

Prefetching – die vorsorgliche Übertragung in Zukunft wahrscheinlich benötigter Daten – stellt in verteilten Client-Server-Systemen ein wichtiges Konzept dar, um die Leistungsfähigkeit, Interaktivität und Performanz des Gesamtsystems zu verbessern. Gerade für Rich-Client-Anwendungen, bei denen die Applikationslogik in den Clients untergebracht ist, birgt ein gut an das System angepasstes Prefetch-Verhalten ein erhebliches Potential zur Steigerung der Gesamtleistung des Client-Server-Systems. Die Optimierung des Prefetch-Verhaltens eines Client-Server-Systems ist gerade bei Systemen mit wechselnder Datenmodellierung, speziellen Anpassungen für Kunden und verschiedenen Benutzerrollen sehr zeitraubend und schwierig. Ziel dieser Arbeit ist es daher, ein System zum automatisierten Erlernen einer angepassten Prefetch-Logik auf Basis der Content-Management-Plattform *SCHEMA ST4* zu entwickeln, das sich adaptiv sowohl an wechselnde Datenmodellierungen als auch an das individuelle Verhalten verschiedener Benutzer anpassen kann.

1 Einführung

Die Gesamtleistung eines Client-Server-Systems hängt wesentlich von der Effizienz der Kommunikation der beteiligten Komponenten untereinander ab. Unzureichender Datenfluss zwischen den Komponenten sorgt häufig dafür, dass in größeren Projekten die Leistung des Systems von Kunden und Benutzern schlecht bewertet wird. Verschlimmert wird die Situation dadurch, dass mit dem Aufkommen ausreichend schneller Internet-Verbindungen Teams in einem Client-Server-System oft nicht mehr lokal im Intranet, sondern weltweit zusammenarbeiten und sich damit Anforderungen an Systeme ergeben, die vor einigen Jahren noch nicht realisierbar waren. Daher ist es notwendig, effiziente und leistungsfähige Algorithmen und Architekturen zu entwickeln, die es ermöglichen, den Datenfluss zwischen den Komponenten zu verbessern.

Für die sogenannten *Thin-Client-Architekturen*, ist dies ein gut erforschtes Gebiet [Knafla, 1999][Padmanabhan and Mogul, 1996][Schäffer, 2003][Teng *et al.*, 2005]. *Thin-Clients* übernehmen dabei nur die Darstellung der Daten. Alle anderen Aufgaben wie die Applikationslogik, Datenmanagement und Datenhaltung werden von einer (*Two-Tier-Architektur*) oder mehreren Schichten *Multi-Tier-Architektur*) übernommen [Bengel, 2002][Coulouris *et al.*, 2002]. Diese Verfahren beruhen hauptsächlich auf intelligentem Caching und Prefetching derjenigen Ansichten, die durch Analyse des Nutzerverhaltens mit hoher Wahrscheinlichkeit als nächstes dargestellt werden müssen.

Rich-Clients als Konkurrenzmodell beinhalten neben der Präsentations- auch noch die Applikationslogik. In dieser Architektur-Art werden häufig Daten zwischen den Clients und der unterliegenden Serverschicht ausgetauscht und clientseitig verarbeitet. Hier besteht großes Potential in der Verbesserung des Datenflusses, da die vom Server angeforderten Daten nicht unkorrelliert angefragt werden, sondern sich anhand von Aktionen und Berechnungsschritten kategorisieren und vorhersagen lassen. Bei den meisten Frameworks – zum Beispiel *Java Data Objects* (*JDO*) oder *Enterprise Java Beans* (*EJB*) oder *SCHEMA ST4* – finden sich nur unzureichende Möglichkeiten, den Datenfluss flexibel und wenn möglich automatisiert zu optimieren.

Als Basis für die Untersuchungen dient die Content-Management-Plattform *SCHEMA ST4* der *SCHEMA Electronic Documentation Solutions GmbH*. Sie ist ein typischer Vertreter der Rich-Client-Architektur mit drei Schichten. Die Clients übernehmen Darstellung, Verarbeitungs- und Applikationslogik, während sich der Applikationsserver um das Datenmanagement und ein Datenbank-System um die Datenhaltung kümmern. *SCHEMA ST4* erlaubt es, Informationen medienneutral zu erfassen, zu organisieren und in verschiedene Ausgabeformate zu produzieren. Einsatzbereiche umfassen unter anderem Erstellung von technischen Dokumentationen, Katalogen, Verwaltung von Werkstattinformationen und Dokumentenmanagement. Die Interaktion des Benutzers mit dem Client findet mittels einer graphischen Benutzeroberfläche statt, die aus sogenannten *Viewlets* besteht. *Viewlets* sind andockbare Fenster mit eigenständiger Funktionalität, die individuell zusammengestellte und konfigurierbare Benutzeroberflächen ermöglichen.

Kennzeichen von *SCHEMA ST4* ist vor allem ein flexibles Daten- und Benutzungsmodell, das je nach Kundenwunsch in Funktionalität und Umfang angepasst werden kann. Dieses System macht eine Optimierung des Datenflusses zwischen den Komponenten wünschenswert, die einfach und flexibel anpassbar ist bzw. möglichst sogar automatisch und adaptiv den Datenfluss optimiert.

Abschnitt 2 beschreibt die grundlegenden Möglichkeiten zur Datenmodellierung sowie ein bereits vorhandenes Verfahren zur Errechung der Prefetch-Menge. Anschließend wird in Abschnitt 3 das neue Verfahren zur Lösung des Problems, das regelbasierte, adaptive Prefetching vorgestellt. In Abschnitt 4 wird die Leistung und Effizienz des Systems evaluiert und Abschnitt 5 bietet eine Diskussion der Resultate und einen Ausblick auf mögliche Erweiterungen.

2 Grundlagen

Ein großer Teil der Rechenzeit wird in der Regel mit der Kommunikation der beteiligten Schichten und Komponenten untereinander verbracht. Die konkrete Realisierung geeigneter und performanter Caching- und Prefetching-Algorithmen ist daher in großem Maße von der unterliegenden Systemarchitektur und den Wechselwirkungen zwischen den einzelnen Komponenten abhängig. Dieser Abschnitt beleuchtet zum einen die Abläufe und Datenstrukturen des Softwarepakets *SCHEMA ST4* und zum anderen *FetchExpressions*, ein serverseitiges System zur Errechnung der Prefetch-Menge, das bereits als Basislösung existierte.

2.1 Struktureller Aufbau der Daten in SCHEMA ST4

Die Optimierung der Prefetch-Daten soll durch Analyse der angefragten Daten in *SCHEMA ST4* eine Verbesserung des Datenflusses errechnen. Deshalb muss zunächst geklärt werden, welchen strukturellen Aufbau die Daten haben und welche Anfragemöglichkeiten existieren.

Datenstruktur

Die Daten besitzen in *SCHEMA ST4* eine objektorientierte Zugriffsstruktur. Eine Persistenzschicht sorgt für das Mapping der Objekte auf die relationale Datenbank. Jedes Objekt besitzt eine eindeutige ID, über die es systemweit identifizierbar und ansprechbar ist.

Abbildung 1 zeigt den primären Aufbau der Daten in *SCHEMA ST4*. *Knoten* (Typ: *Node*) sind durch gerichtete Kanten, sogenannte *Links*, miteinander verbunden. In der Regel werden bedeutungstragende Inhalte an Knoten gespeichert und mit Hilfe von Links zusammengestellt oder in Kontext gesetzt. Es gibt hierarchische Links (durchgezogene Linie) und nicht-hierarchische Links (gepunktete Linie). Alle Knoten sind über mehrere Ebenen hierarchisch und azyklisch verbunden. Zwischen Knoten können zudem über nicht-hierarchische Links semantische Beziehungen (auch zyklisch) hergestellt werden. Vom Knoten aus ergeben sich somit vier verschiedene Sichtweisen: der hierarchische Link zum Elternknoten (*ParentLink*), hierarchische Links zu den Kinderknoten (*ChildrenLinks*) und beliebige, nicht-hierarchische eingehende und ausgehende Links (*IncomingLinks* bzw. *OutgoingLinks*).

Sowohl an Knoten als auch an Links ist es möglich Datenwerte zu setzen. Zusätzliche Mächtigkeit verleiht den Datenwerten das Konzept der *Dimensionen*. Eine Dimension enthält *Aspekte*, die als zusätzliche Indirektionsebene zum eigentlichen Inhalt eines Datenwerts eingefügt worden ist. Ein Beispiel wäre eine Dimension 'Sprache', welche die Aspekte Deutsch, Englisch und Französisch enthält. Wird ein Datenwert in Abhängigkeit der Dimension 'Sprache' definiert, können drei verschiedene Datenwertprimitive existieren, in denen der Wert für die jeweilige Sprache enthalten ist. Wenn der Datenwert allerdings *dimensionsunabhängig* ist, gibt es nur ein Datenwertprimitiv, welches den Inhalt des Datenwerts enthält.

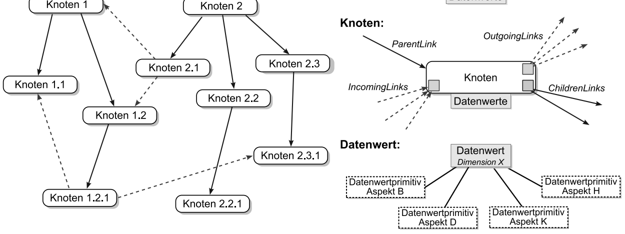


Abbildung 1: Schematische Darstellung der primären Datenstruktur in *SCHEMA ST4*

Informationsmodell

Neben der syntaktischen Objektstruktur ist in *SCHEMA ST4* ein semantisches Meta-Modell – das sogenannte *Informationsmodell* – vorhanden, welches eine Klassifikation und eine Festlegung von Beziehungen zwischen einzelnen Knoten, Links und Datenwerten ermöglicht.

Knoten sind je nach Bedeutung unterteilt in mehrere *Knotenklassen*, die per Vererbungshierarchie gebunden sind. Pro Knotenklasse können verschiedene Datenwerte registriert werden, die wiederum in Form von Knoten dieser Knotenklasse instanziierbar sind.

Auch Links sind je einer *Linkklasse* zugeteilt, die die semantische Bedeutung des Links zuweist. So sind innerhalb der beiden Grundformen (hierarchisch und nicht hierarchisch) feinere Unterschiede möglich. Genau wie an Knotenklassen können auch an Linkklassen Datenwerte registriert werden. Linkklassen besitzen sogenannte *Linkklassenprimitive*. Diese legen fest, welcher Klasse der Quell- und der Zielknoten angehören muss, damit der Link gezogen werden darf und ob er ein Hierarchie-Link ist.

Die Modellierungsmöglichkeiten von Datenwerten machen sich in zweierlei Hinsicht bemerkbar. Zum einen besitzen Datenwerte einen Typ, der bestimmt, welcher Datentyp in den Datenwerten gespeichert werden kann. In *SCHEMA ST4* sind folgende Typen möglich: *DateTime*, *Decimal*, *Dictionary*, *Double*, *Int32* und *String* verhalten sich analog zum .NET-Gegenstück [Liberty, 2003]. Dazu kommen noch *History* (versionierter Wert), *Label* (Versionsmarkierung), *Selection* (Enumeration), *Stream* (binärer Datenstrom), *Text* (streamingfähige Zeichenkette) und *XML* (XML-Fragment). Zum anderen sind Datenwerte gekennzeichnet durch *Datenklassen*, in denen festgelegt sind, an welcher Knoten- bzw. Linkklasse mit welchem Namen der Datenwert ansprechbar ist. Ein Datenwert kann zudem mit einem anderen Datenwert derselben Datenklasse verknüpft werden. Das bedeutet, dass die zugehörigen Datenwertprimitive die des verknüpften Datenwerts sind.

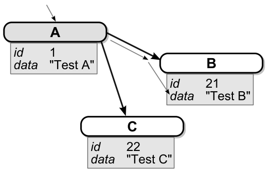
Anfragesprachen: StPath und STT

Zur Vereinfachung der Anfrage an die Objektstruktur und zur Navigation auf den Objekten selbst existiert in *SCHEMA ST4* eine spezielle Anfragesprache: *StPath*. Diese orientiert sich sehr stark an *XPath 1.0* [Consortium, 1999a] für XML. An die Stelle des „XML Document Object Model (DOM)“ führt *StPath* Anfragen auf dem Objektmodell von *SCHEMA ST4* aus. Neben der *XPath*-ähnlichen Funktionalität gibt es allerdings in *StPath* zusätzlich noch spezifische Erweiterungen für das ST4 Objektmodell, zum Beispiel die Bestimmung der Klasse eines Knotens oder die Navigation über ausgehende oder eingehende Links.

Ein einfache Anfrage kann anhand von Abbildung 2 erläutert werden. Die Modellierung in *SCHEMA ST4* links entspricht dem XML-Auschnitt auf der rechten Seite. Knoten werden in XML zu Elementen, Datenwerte werden Attributen gleichgesetzt. Die Anfrage

```
child::B/@data
```

selektiert in *SCHEMA ST4* erst alle Kindknoten vom Typ *B* (des aktuellen Kontextknotens) und gibt den



```
<A id="1" data="Test A">
    <B id="21" data="Test B" />
    <C id="22" data="Test C" />
</A>
```

Abbildung 2: Wirkungsweise von StPath

Datenwert *content* der selektierten Knotenmenge aus. Analog werden in XML alle Kinder mit dem Elementnamen *B* gesucht und deren Attribute mit Namen *content* ausgegeben – in diesem Beispiel die Zeichenkette „Test B“.

Weiterhin erlaubt das analog zu *XSLT 1.0* [Consortium, 1999b] aufgebaute *STT* komplexe zusammengesetzte Transformationen auf der Objektstruktur. Die Mächtigkeit der Ausgabe-Operationen von *STT* und *XSLT* sind in etwa vergleichbar. Zudem kann sowohl StPath als auch STT über in C# geschriebene Funktionen erweitert werden. Somit werden zum Beispiel auch komplexe, performante oder systemnahe Funktionalitäten möglich.

2.2 Vorhandenes Verfahren: FetchExpressions

In *SCHEMA ST4* existiert bereits ein Verfahren, welches es erlaubt, das Prefetch-Verhalten der Clients zu manipulieren. Dieses wird im folgenden Abschnitt vorgestellt.

Prinzip

Das *FetchExpression-System* ist ein serverbasiertes Verfahren zur Berechnung der durch Prefetching geholten Objektmenge. Die Clients teilen dem Server nur mit, welches Objekt sie aktuell in welcher Version vom Server geliefert bekommen wollen. Kommt eine Objekt-Anforderung beim Server an, so wird diese kategorisiert. Für jede registrierte Kategorie existieren *StPath*-Ausdrücke, die bestimmen, welche Objekte zusätzlich als Prefetch-Menge dem Client mitgeliefert werden sollen.

Das FetchExpression-System kennt zwei Strukturierungsebenen, um Anfragen in Kategorien aufzuteilen. Zum einen können *Gültigkeitsbereiche* festgelegt werden, sogenannte *(FetchExpression-)Scopes*, denn je nach durchgeführter Aktion werden bestimmte, eventuell unterschiedliche Daten, die vom angefragten Objekt abhängen, benötigt. Der Client teilt dem Server bei Anfrage immer mit, in welchem Gültigkeitsbereich er die Anfrage stellt.

Zum anderen existieren innerhalb der Gültigkeitsbereiche Regeln, die bestimmen, welche Objekte genau geholt werden sollen, die eigentlichen *FetchExpression*s. Diese funktionieren nach folgendem Prinzip: Erfolgt eine Anfrage an ein bestimmtes Objekt, wird versucht, dieses Objekt einer FetchExpression zuzuweisen. FetchExpressions können entweder auf den Typ oder auf die Klasse des Objekts reagieren. Die erste Regel, die zur Anfrage passt, wird mit dem angeforderten Objekt als Kontext ausgewertet. Alle Objekte, die der Server zur Auswertung des Ausdrucks benötigt, werden dem Client als Prefetch-Menge mit übermittelt.

Das FetchExpression-System ist eine proprietäre Technologie von *SCHEMA ST4*. Allerdings gibt es auch andere, teils standardisierte Systeme, die auf demselben Prinzip beruhen. Im kommenden Standard *JSR 243: Java Data Objects 2.0 (JDO)* [Process, 2005] zum Beispiel, das bei Erstellung dieser Arbeit als *Proposed Final Draft* vorlag, gibt es sogenannte *FetchPlans*. Diese erlauben es analog zu Gültigkeitsbereichen, je nach Anwendungsfall einen sogenannten *FetchPlan* festzulegen, der definiert, welche Daten bei Anforderung eines bestimmten Objekts zurückgeliefert werden müssen.

Probleme und Anforderungen

FetchExpressions werden in *SCHEMA ST4* bereits seit mehreren Jahren eingesetzt, um die Gesamtleistung des Client-Server-Systems zu erhöhen. Trotz der prinzipiellen Möglichkeiten, das Prefetch-Verhalten der Clients zu steuern, gibt es Einschränkungen, die eine grundlegende Verbesserung des Systems wünschenswert machen:

1. FetchExpressions werden zentral auf dem Server hinterlegt und gelten gleichermaßen für alle Clients. Es können weder Benutzerrollen noch Benutzerverhalten noch Ausbaustufen der Clients hinsichtlich der Funktionalität berücksichtigt werden.
2. Zeitlich variable Anfragestrukturen können von FetchExpressions nicht abgebildet werden. Vor allem durch Änderungen an der Benutzeroberfläche – zum Beispiel das Hinzu- oder Wegnehmen von Viewlets – kann sich das Anfrageverhalten aber sehr stark ändern. Mit FetchExpressions kann darauf nicht oder nur sehr eingeschränkt reagiert werden.
3. Die manuelle Erstellung und Optimierung der FetchExpressions gestaltet sich aus zwei Gründen schwierig: Zum einen sind die Implikationen bei Änderungen an den Regeln vielschichtig und erfordern viel Spezialwissen um die Struktur und den Aufbau der Modellierung. Zum anderen ist es nicht einfach, mögliche Stellen für die Optimierung ausfindig zu machen, sofern schon eine Basisoptimierung erfolgt ist. Hierfür ist es bis jetzt notwendig, lange Log-Dateien durchzuschauen und durch Ausprobieren und Erfahrung die richtige Stelle zu finden.

Ein besseres System zur Berechnung der Prefetch-Menge sollte also nach Möglichkeit alle diese Probleme beseitigen, indem eine Optimierung per Client stattfindet, die sich adaptiv an die jeweiligen Gegebenheiten und Anfragestrukturen anpassen kann. Daraus ergibt sich insbesondere, dass ein solches System fähig sein muss, die Daten in Echtzeit ohne relevanten Einfluss auf den Betrieb des Clients zu verarbeiten und verbesserte Optimierungsregeln daraus abzuleiten. Hinsichtlich der Qualität des Prefetchings muss sich das neue, automatische System auch in etwa mit den handoptimierten FetchExpressions messen können.

3 Regelbasiertes, adaptives Prefetching (RBAP)

RBAP ist ein Prefetch-Verfahren, welches eine optimale Menge an Prefetch-Objekten durch Kombination von client- und serverseitigen Modulen errechnet. Da die Applikationslogik bei Rich-Client-Anwendungen im Client untergebracht ist, sind nur dort alle notwendigen Informationen vorhanden, die das Verfahren benötigt, um eine optimale Prefetch-Menge zu berechnen. Im Client kann bestimmt werden, welche Objekte in welcher Version mit welchem Gültigkeitsbereich angefragt wurden, welche Objekte vom Server geliefert wurden und welchen Cache-Zustand ein Objekt bei Anfrage hat. Mögliche Zustände sind *Cache-Hit* (Objekt ist im Cache vorhanden), *Cache-Miss* (Objekt ist nicht im Cache und muss vom Server geholt werden) und *Cache-Prefetch* (Objekt wurde aufgrund einer Anfrage als Prefetch-Objekt mitgeliefert).

Abbildung 3 stellt den schematischen Aufbau von *RBAP* dar. Die Optimierung des Prefetch-Verhaltens erfolgt durch ein *Lernmodul* im Client, welches die relevanten Informationen sammelt und auswertet. Anhand dieser Datenbasis kann das Lernmodul eine Verbesserung des aktuell gültigen *Prefetch-Modells* vornehmen. Dieses umfasst alle Daten, die das *Auswertungsmodul* im Server benötigt, um auf eine Anfrage vom Client die passende Objektmenge zu berechnen und zurückzuliefern. Sollte das Lernmodul feststellen, dass sein Prefetch-Modell verbessert werden kann, wird sowohl sein eigenes als auch das korrespondierende Modell im Server angepasst.

3.1 Lernprozess

Der Lernprozess ist ständig als Hintergrund-Prozess im Client aktiv und muss schnell anhand der aktuellen Datenlage Entscheidungen treffen. Daraus ergibt sich eine wichtige Einschränkung: Das Verfahren und die Algorithmen müssen so gestaltet sein, dass der Lernprozess nicht zu viel Rechenzeit und Hauptspeicher verbraucht. Somit ist es nur schwer möglich etablierte Standard-Verfahren einzusetzen. Statt dessen kommt für *RBAP* eine Kombination aus regel- und schwellwertbasiertem Verfahren zum Einsatz, das in zwei Phasen eine Optimierung des Prefetch-Modells unter Beachtung der Beschränkungen vornehmen kann.

Knoten als grundlegender Optimierungsblock

Die wichtigen bedeutungstragenden Einheiten in *SCHEMA ST4* sind Knoten und Links zusammen mit den an beiden Objekttypen vorhandenen Datenwerten und Datenwertprimitiven. Alle anderen Typen stellen *Verwaltungsinformationen* dar. Die für den Lernprozess relevanten Daten sind daher hauptsächlich Knoten, Linkklassen und Datenwerte (mit ihren Datenwertprimitiven).

Für den Lernprozess ist es aus Gründen der Datenreduktion sinnvoll, eine Einheit zusammengehöriger Objekte zu definieren, den sogenannten *Optimierungsblock*. Um alle wichtigen Elemente aufzunehmen, gibt es zwei sinnvolle Möglichkeiten, einen Block darzustellen:

- *Knotenbasierter Optimierungsblock*
  Zum Optimierungsblock gehören alle Objekte, die von einem Knoten ausgehend erreichbar sind, ohne dass ein anderer Knoten traversiert werden muss.
- *Linkbasierter Optimierungsblock*
  Zum Optimierungsblock gehören alle Objekte, die von einem Link ausgehend erreichbar sind, ohne dass ein anderer Link traversiert werden muss.

Für FetchExpressions wird hauptsächlich in knotenbasierter Ansatz verwendet. Nur in wenige Fällen, für die eine knotenbasierte Herangehensweise keine zufriedenstellenden Ergebnisse geliefert hat, ist durch einen linkbasierten Ansatz teilweise eine Verbesserung erzielt worden. RBAP verwendet einen knotenbasierten Optimierungsblock aus zweierlei Gründen: Zum einen gestaltet sich Phase 1 (Registrierung) eindeutiger und damit einfacher. Ein Knoten (unterschieden nach Knotenklasse) kann Unterstruktur von mehreren Links (unterschieden nach Linkklasse) sein, je nach Anzahl der Linkklassenprimitiven, die diesen Knoten als Quelle oder Ziel referenzieren. Ist der Link jedoch dem Knoten untergeordnet, dann gibt es für den Link nur zwei Zuordnungsmöglichkeiten – die Quelle oder das Ziel. Zum anderen wird in der Verwendung der linkbasierten Herangehensweise kein offensichtlicher Vorteil gesehen. Zudem gestaltet sich die Vorstellung und das Nachvollziehen der Struktur anhand der eher auf Knotenvisualisierung ausgelegten Oberfläche von *SCHEMA ST4* einfacher.

Abbildung der Anfragestruktur

Die Optimierung nur mit Knoten verbundener Daten beschränkt die mehr als 100 vorhandenen Typen in *SCHEMA ST4* auf etwa 40, die wirklich für die Optimierung relevant sind – hauptsächlich Datenwerttypen, Knoten und Links. Die übrigen Objekttypen legen entweder Verwaltungsinformationen oder Strukturen des Informationsmodells fest.

In Abbildung 4 ist links die Datenstruktur für einen Optimierungsblock dargestellt, wie sie für die Realisierung von *RBAP* im Rahmen dieser Arbeit entwickelt worden ist. Der *NodeOptimizer* ist der Einstiegspunkt in den Optimierungsblock. Er beinhaltet allerdings keine weitere Optimierungslogik, sondern wird für die Optimierung nur traversiert, da der Knoten selbst nicht aus dem Prefetch-Modell entfernt werden kann. Allerdings kümmert sich der *NodeOptimizer* um alle Registrierungen am Optimierungsblock und verteilt die Registrierung weiter auf die richtigen Unterstrukturen. Die Behandlung von Datenwerten übernimmt der *DataValuesOptimizer*. Anfragen an Datenwerte werden kategorisiert über die ID ihrer Datenklasse. Für Links ist ein mehrstufiger Prozess notwendig. Eine Kategorisierung erfolgt zunächst nach ihrem Typ (*LinksOptimizer*) und dann nach ihrer Linkklasse (*LinkClassDictionary* in *LinkOptimizer*). Dies ist notwendig um eine ausreichend mächtige und zugleich flexible Handhabung von Links zu ermöglichen.

Auf der rechten Seite von Abbildung 4 ist das dem Optimierungblock entsprechende Datenlayout im *NodePrototype* aufgezeigt. Datenwerte werden sowohl an Knoten als auch an Links als eine Liste von IDs (*Int64*) repräsentiert. Das Bitfeld *_linkPolicy* fasst die Navigationsmöglichkeiten für Links zusammen. Entweder es werden nur Daten vom Link-Container abgefragt (z.B. die Anzahl der ausgehenden Links) oder es werden alle Links abgefragt und gegebenenfalls gefiltert (z.B. alle ausgehenden Links der Linkklasse *Informationshierarchie*). Im ersten Fall muss nur der Link-Container, im zweiten Fall der Link-Container mit al-

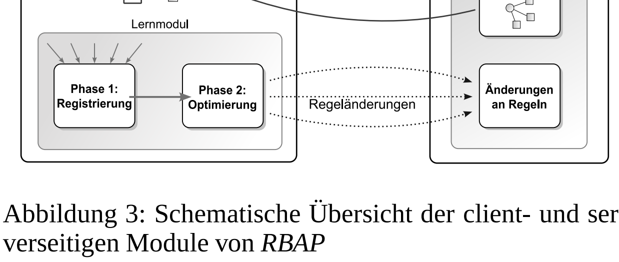
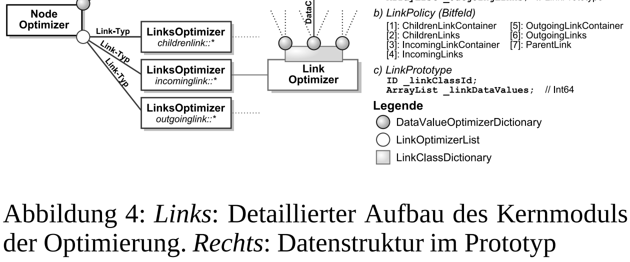


Abbildung 3: Schematische Übersicht der client- und serverseitigen Module von *RBAP*



Datenlayout

a) *NodePrototype*

```
ID _nodeClassID
byte _linkPolicy;
ArrayList _nodeDataValues;        // Int64
ArrayList _parentLinkDataValues;  // Int64
ArrayList _childrenLinks;  // LinkPrototype
ArrayList _incomingLinks;  // LinkPrototype
ArrayList _outgoingLinks;  // LinkPrototype
```

b) *LinkPolicy (Bitfeld)*

```
[1]: ChildrenLinkContainer     [5]: OutgoingLinkContainer
[2]: ChildrenLinks             [6]: OutgoingLinks
[3]: IncomingLinkContainer     [7]: ParentLink
[4]: IncomingLinks
```

c) *LinkPrototype*

```
ID _linkClassId;
ArrayList _linkDataValues;   // Int64
```

Abbildung 4: *Links*: Detaillierter Aufbau des Kernmoduls der Optimierung. *Rechts*: Datenstruktur im Prototyp

len Links geholt werden. Weitere Möglichkeiten sind navigationstechnisch für Links nicht vorgesehen.

PHASE 1: Registrierung der Objekte

Jede Objekt-Anfrage wird zunächst vom Lernmodul im Client registriert. Das heißt das jedes Objekt seinem zugehörigen Optimierungsblock zugewiesen und die ID sowie der Cache-Zustand an geeigneter Stelle im Optimierungsmodul festgehalten wird. Zum Zweck der Datenreduktion werden Knoten nach Knotenklasse, Links nach Linkklassen und Datenwerte nach Datenklassen kategorisiert.

Für untergeordnete Strukturen (Datenwerte und Links) wird die Anzahl der Verwendungen am Knoten eingetragen (unterhalb der Struktur für die zugehörige Klasse). Für einen Cache-Hit wird die Anzahl der Verwendungen um *1* erhöht, ein Cache-Miss setzt die Zahl der Verwendungen für das Objekt auf *1* ein Cache-Prefetch auf *0*. Somit ist es möglich, herauszufinden, wie viele Objekte einer bestimmten Klasse pro Optimierungseinheit angefragt wurden und ob angefragte Objekte einer bestimmten Klasse verwendet oder nur per Prefetch geholt worden sind. Anhand dieser Informationen entscheidet Phase 2, ob das Prefetch-Modell verbessert werden kann. Wenn genügend Objekte in mindestens einem Optimierungsblock erfasst und verarbeitet worden sind, wird ein Lernprozess gestartet, der den Prefetch-Vorgang optimieren soll.

PHASE 2: Verbessern des Prefetch-Modells

Phase 2 beschreibt den eigentlichen Lernprozess. Sämtliche vorhandenen, untergeordneten Strukturen des Optimierungsblocks werden traversiert. Pro Struktur wird entschieden, ob sie ins Prefetch-Modell aufgenommen bzw. entfernt wird.

1. *Struktur ist nicht im Prefetch-Modell vorhanden*
   Die Struktur kann dem Prefetch-Modell hinzugefügt werden, wenn die Anzahl der angefragten Objekte, die dieser Struktur zugeordnet sind, einen bestimmten Schwellwert überschreitet. Der Schwellwert richtet sich nach der Anzahl der Anfragen an den übergeordneten Knoten und nach der Anzahl der verfügbaren Instanzen der Knotenklasse.
2. *Struktur ist im Prefetch-Modell vorhanden*
   Die Struktur kann aus dem Prefetch-Modell entfernt werden, wenn gewisse Zeit nach der ersten Registrierung des Objekts kein oder nur wenige Objekte der Struktur verwendet worden sind. Als Entscheidungskriterium dient ein Schwellwert, der sich an der Anzahl der Anfragen an den übergeordneten Knoten orientiert.

Änderungen am Optimierungsblock werden dem Server mitgeteilt, damit dieser künftig anhand des verbesserten Prefetch-Modells Objekte an den Client zurückliefern kann. Das Prefetch-Modell wird also angepasst, sobald genügend Daten vorhanden sind, die eine Optimierung ermöglichen. Sofern sich das Benutzerverhalten oder das Anfrageverhalten (z.B. durch Hinzufügen von Viewlets) nicht wesentlich ändert, konvergiert das Prefetch-Modell mit steigender Zahl der Optimierungsläufe gegen das aufgrund der Datenbasis bestmögliche Prefetch-Modell.

Generationen: Modellierung temporaler Eigenschaften

Für die effiziente Abarbeitung der Optimierung des Prefetch-Modells ist es wichtig, dass nur eine begrenzte Anzahl von Daten vorliegen. Die Verarbeitung zu langer Anfrageprotokolle nimmt zu viel Zeit in Anspruch. Außerdem ist die Modellierung eines zeitlichen Verhaltens für die Optimierungsphase wichtig, denn der Lernprozess muss entscheiden, ob bestimmte Strukturen gebraucht wurden oder nicht.

Zur Lösung dieser Probleme werden sogenannte *Generationen* eingeführt. Generationen gelten unabhängig pro Optimierungsblock. In die nächste Generation wird vorgerückt, sobald sich eine Regel im Optimierungsblock ändert oder eine bestimmte Zahl von Optimierungen an diesem Knoten versucht wurden. Es wird also mit Hilfe der für die Optimierung verfügbaren Daten eine zeitliche Struktur modelliert, da ein Optimierungslauf in einer Optimierungseinheit abhängig ist von der Zahl der Informationen für eine Optimierung. Aus Gründen der Datenreduktion gibt es eine maximale Anzahl von Generationen die vorgehalten werden. Im Rahmen der Tests von *RBAP* hat sich eine Zahl von maximal sechs Generationen als günstig in Hinsicht auf Geschwindigkeit und Speicherverbrauch erwiesen.

3.2 Auswertung der Daten im Server

Die Aufgabe des Applikationsservers ist es, auf eine Anfrage des Clients passende Objekte zurückzuliefern. Es muss auf jeden Fall immer das angefragte Objekt in der richtigen Version vom Server mitgeliefert werden, weil ansonsten der Client einen Fehler an der Remoting-Verbindung meldet und die Verbindung zum Server abbricht. Welche weiteren Objekte als Prefetch-Menge übertragen werden, bestimmen sowohl das für die Sitzung gültige Prefetch-Modell als auch eine Menge von Standardregeln.

Zur Auswertung einer Anfrage wird im Server in der Regel – sofern Knoten angefragt werden – das aktuell gültige Prefetch-Modell herangezogen. Im Modell wird nach einer passenden Prefetch-Regel für den Knoten gesucht und ausgewertet. Dadurch wird festgelegt, welche Objektmenge zusätzlich an den Client übertragen werden soll. Bei der Auswertung der Regeln orientiert sich das Server-Modul an den Navigationsstrukturen, die *StPath* vorgibt. Die Prefetch-Regel soll idealerweise für ein Objekt *O* all diejenigen Objekte zurückliefern, die ein *StPath*-Ausdruck benötigt, um eine Anfrage an das Objekt *O* erfolgreich auszuführen.

Zusätzlich wird diese Menge noch gefiltert, um Informationen die sich in der Regel im Client-Cache befinden (zum Beispiel Informationen über Knotenklassen) nur beim ersten Aufruf bzw. nicht mehrfach zu übertragen.

Als Beispiel dient die Abfrage eines Datenwerts an einem Knoten. In diesem Fall muss neben dem Datenwert selbst und dem Datenwertprimitiv im eingestellten Aspekt auch noch die Datenklasse und der dazugehörige Rechte-Container zurückgeliefert werden. Zudem müssen noch alle Objekte berücksichtigt werden, die zur Evaluierung notwendig sind. Zu jedem Objekt muss ein vollständiger Pfad vom Ausgangsobjekt aus verfügbar sein. Im Fall eines Datenwerts muss deswegen zusätzlich der Datenwert-Container am Knoten zur Prefetch-Menge hinzugefügt werden.

Standardregeln

Das Standardregelwerk sorgt dafür, dass für alle angefragten Objekte, die keine Knoten sind, eine sinnvolle Objektmenge an den Client übertragen wird. Die einfachste Realisierungsmöglichkeit besteht darin, nur das angefragte Objekt selbst zurückzuliefern. Allerdings ist dies von Seiten

	Anfragen	Objekte
Unoptimiert	884	884
FetchExpressions	239 (≈ 0.27)	2324 (≈ 2.63)
RBAP (initial)	322 (≈ 0.36)	1612 (≈ 1.82)
RBAP (optimal)	292 (≈ 0.33)	1578 (≈ 1.79)

Tabelle 1: Vergleichswerte der Verfahren für *Test 1: Start des Clients*

	Anfragen	Objekte
Unoptimiert	916 (+ 32)	916 (+ 32)
FetchExpressions	431 (+ 192)	3992 (+ 1668)

Tabelle 2: Anzahl der Anfragen bei eingeschaltetem Lernmodul (*Test 1: Start des Clients*). Der Wert in Klammern kennzeichnet die Änderung der Werte gegenüber nicht aktivem Lernmodul.

der Performanz sehr bedenklich, da in diesem Fall unnötig viele einzelne Anfragen an den Server gestellt werden.

Die Standardregeln sind vom Prinzip her den ursprünglichen FetchExpressions sehr ähnlich. Sie agieren als zusätzliche Prefetching-Instanz, die eine serverseitig optimierte Objektmenge zurückzuliefert, wo das Prefetch-Modell aufgrund der Granularität nicht weiterhilft. Der Unterschied zu den FetchExpressions besteht darin, dass alle vom Prefetch-Modell behandelten Regeln nicht mehr beachtet werden müssen und die Regeln wesentlich einfacher und allgemeiner gehalten sind.

Die Standardregeln sind fest in den Server einprogrammiert. Ihre Gestaltung ist durch Analyse von Protokolldaten der Interaktion zwischen Client und Server manuell festgelegt worden und stellt implizites Wissen über Anfragestrukturen des Systems dar. Prinzipiell scheint es auch möglich, die Standardregeln über einen Lernprozess zu definieren. Im aktuellen System wurde dies aber noch nicht realisiert, da das Verfahren zur Ableitung dieser Regeln wesentlich komplexere, zeitaufwändigere Untersuchungen (und somit eine Offline-Auswertung) benötigen würde und explizites Wissen über alle Navigationsmöglichkeiten des Datenmodells modelliert werden müssten.

Intern werden für Standardregeln zwei verschiedene Fälle berücksichtigt: Objekte außerhalb und Objekte innerhalb eines Optimierungsblocks. Eine Unterscheidung kann in der Regel durch den Objekttyp getroffen werden.

Objekte außerhalb der Optimierungsblöcke sind hauptsächlich Verwaltungsinformationen. Ihre Anzahl ist im Vergleich zu Knoten, Links und Datenwerten in der Regel sehr gering. Die Standardregeln sorgen hier hauptsächlich dafür, dass diese Objekte nicht einzeln vom Server abgeholt werden, sondern blockweise zusammen. Dies stimmt auch mit der Art überein, wie sie vom Client benötigt werden.

Innerhalb der Optimierungsblöcke werden Standardregeln für Objekte eingeführt, die wegen Performanz und Granularität nicht mit optimiert werden. Geregelt wird vor allem der Zugriff auf einzelne Datenwertprimitive. So ist zum Beispiel festgelegt, dass beim Zugriff auf einen Datenwert gleich das zugehörige Datenwertprimitiv im aktuell eingestellten Aspekt mitgeliefert wird. So können unnötige Latenzen verhindert werden. Allerdings muss auch der Client angemessen auf diese Objekte reagieren und in der Registrierung berücksichtigen, dass diese Objekte serverseitig – ohne bestehende Optimierungsregel im Client – mitgeliefert worden sind.

4 Ergebnisse

In diesem Abschnitt wird die Gesamtleistung von *RBAP* als adaptives System zum Erlernen des Prefetch-Verhaltens analysiert, bewertet und mit FetchExpressions verglichen. Eine Analyse der Systeme findet hauptsächlich qualitativ statt. Gemessen werden die Anzahl der Anfragen an den Server und die Gesamtzahl der zurückgelieferten Objekte. Zeitmessungen gestalten sich aufgrund möglicherweise unterschiedlicher Netzwerktopologien als nicht sehr aussagekräftig. Liegt eine sehr hohe Latenz vor dann nimmt die Anzahl der einzelnen Anfragen maßgeblich Einfluss auf die Geschwindigkeit. Ist das Problem der Datendurchsatz der Netzwerkverbindung, dann verlangsamt eine große Gesamtzahl an vom Server zurückgelieferten Daten das Gesamtsystem.

Außerdem ist zu bedenken, dass die Verfahren, die miteinander verglichen werden, einen von Grund auf verschiedenen Aufbau haben. FetchExpressions sind ein rein serverseitiges Verfahren, während RBAP client- und serverseitig agiert. Es mussten viele Kernkomponenten (vor allem im Server) ausgetauscht werden, die nicht gleichwertig ersetzt wurden. Die Registrierung und das Lernen im Client benötigen zudem mehr Daten als eigentlich von der Anwendung in der Regel gebraucht werden.

Beide Verfahren werden anhand von ausgewählten Testfällen objektiv beurteilt. Als Datenbasis dient der Standard-Export der Datenbank für die Version *1.3.0* des *ST4 DocuManagers 1.3*.

- *Test 1: Start eines Clients*
  Der Start eines Clients testet die Güte der Standardregeln. Es werden viele Verwaltungsinformationen angefragt, aber nur wenige Knoten. Die Werte sind als Basis für die weiteren Testfälle zu sehen, da die anderen Testdaten immer unter der Voraussetzung gewonnen werden, dass der Client gestartet wurde und sich die dafür notwendigen Daten bereits im Cache befinden.
- *Test 2: Produktion eines Dokuments*
  Nach Start des Clients wird direkt ein HTML-Dokument produziert. Als Daten dienen das Beispielprojekt *Projekt HTML*. Dieser Test behandelt die Fähigkeit des Optimierungsmoduls, die für die Produktion notwendigen Datenwerte zu erlernen. Eine Produktion läuft in einem gesonderten Gültigkeitsbereich. Die Regeln überschneiden sich daher nicht mit den vorher gewonnenen Optimierungsdaten.
- *Test 3: Aufklappen aller Ebenen des Dokumentbaums*
  Dieser Testfall behandelt vor allem die Anpassungsfähigkeit des Verfahrens, wenn viele Links und zugehörige Linkdatenwerte ausgewertet werden müssen. Der Startknoten ist der Wurzelknoten des Dokumentbaums. Von dort aus werden alle Unterebenen rekursiv in einer Aktion aufgeklappt.

4.1 Vergleich zwischen FetchExpressions und RBAP

Im folgenden Abschnitt wird überprüft, wie gut sich *RBAP* als Prefetch-Verfahren im Vergleich zum FetchExpression-System eignet. Dies wird anhand der drei Testfälle bewertet, indem untersucht wird, wie viele einzelne Anfragen an den Server gestellt und wie viele Objekte vom Server zurückgeliefert wurden. Als Basis dienen die Werte des unoptimierten Clients. Generell gilt: Je weniger einzelne An-

	Anfragen	**Objekte**
Unoptimiert	3465	3465
FetchExpressions	600 (≈ 0.17)	5708 (≈ 1.65)
RBAP (initial)	1045 (≈ 0.30)	4310 (≈ 1.24)
RBAP (optimal)	522 (≈ 0.15)	4819 (≈ 1.39)

Tabelle 3: Vergleichswerte der Verfahren für *Test 2: HTML-Produktion*

	Anfragen	**Objekte**
Unoptimiert	2090	2090
FetchExpressions	350 (≈ 0.17)	3279 (≈ 1.57)
RBAP (initial)	577 (≈ 0.28)	2558 (≈ 1.22)
RBAP (optimal)	233 (≈ 0.11)	2632 (≈ 1.26)

Tabelle 4: Vergleichswerte der Verfahren für *Test 3: Aufklappen aller Ebenen des Dokumentbaums*

fragen an den Server gestellt werden und je näher die Anzahl der zurückgelieferten Objekte am Ausgangswert des unoptimierten Clients liegt, desto besser ist das Prefetch-Verhalten.

Test 1: Start des Clients

Die Messergebnisse aus Tabelle 1 zeigen, dass *RBAP* mit dem FetchExpression-System durchaus mithalten kann. Das neue Verfahren stellt zwar etwas mehr Anfragen an den Server, allerdings werden auch wesentlich weniger Objekte vom Server zurückgeliefert. Ist die Latenz im Netzwerk der dominierende Faktor, dann haben FetchExpressions einen leichten Vorteil. Wenn allerdings der Datendurchsatz im Netzwerk eher problematisch ist, hat *RBAP* einen großen Vorsprung.

Ein wichtiger Faktor, der nicht außer Acht gelassen werden sollte, besteht in dem Mehraufwand, den das neue Verfahren leisten muss. In Tabelle 2 ist aufgelistet, wie sich ein unoptimierter Client und ein Client mit FetchExpressions verhält, wenn das Optimierungsmodul im Client eingeschaltet ist. Insgesamt werden nur 32 Werte mehr angefragt – gegenüber den ca. 900 Objekten sonst eine Vergleichsweise geringe Zahl. Interessant ist der doch sehr hohe Zuwachs an Anfragen bei FetchExpressions. Dieser lässt sich hauptsächlich dadurch erklären, dass sich durch das Einschalten des Optimierungsmoduls die Reihenfolge der Objektanfragen ändert und somit Objekte, die sonst durch einen Prefetch mitgeliefert worden wären, einzeln angefragt werden. Dieser Wert unterstreicht die gute Integration der Standardregeln in das neue Verfahren.

Test 2: HTML-Produktion

Bei der HTML-Produktion (Tabelle 3) zeigen sich die Stärken von *RBAP*. Es kann sich genau darauf einstellen, welche Daten für die Produktion gerade benötigt werden. Das neue Verfahren benötigt etwa 10% weniger Anfragen und sorgt noch dazu für wesentlich weniger zurückgelieferte Objekte als FetchExpressions. Wichtig für diesen Testfall ist, dass die Produktion immer in einem gesonderten Gültigkeitsbereich abläuft. So kann das Optimierungsmodul unabhängig von anderen Einflüssen sehr schnell eine angepasste Prefetch-Lösung bereitstellen, die nach wenigen Optimierungszyklen gegen das Optimum konvergiert. Neben der HTML-Produktion wurden noch andere Produktionen untersucht. Diese werden aber nicht separat aufgeführt, da die Ergebnisse bei allen Produktionen abgesehen von minimalen Abweichungen gleichwertig sind.

	Lauf 1	**Lauf 2**	**Lauf 3**	**Lauf 4**	**Lauf 5**
Test 1	322 (3)	295 (1)	292 (0)	292 (0)	292 (0)
Test 2	1045 (33)	689 (10)	579 (4)	522 (0)	522 (0)
Test 3	577 (16)	346 (7)	233 (0)	233 (0)	233 (0)

Tabelle 5: Anpassungsfähigkeit von *RBAP* nach *n* Optimierungsläufen. Der Wert in Klammern ist die Zahl der Änderungen am Prefetch-Modell im Zug des Optimierungslaufs.

Anhand dieses Testfalls lässt sich gut verfolgen, dass *RBAP* nicht nur aufgrund von Standardregeln gute Ergebnisse erzielt. Im initialen Zustand dominiert noch der Einfluss der Standardregeln. Allerdings bewirkt erst die Anpassung des Prefetch-Verhaltens mit der Zeit, dass ein deutlicher Vorsprung zum FetchExpression-System erzielt wird.

Test 3: Aufklappen aller Ebenen des Dokumentbaums

Bei der Interpretation der Ergebnisse von *Test 3* (Tabelle 4) muss bedacht werden, dass dieser Test im selben Gültigkeitsbereich abläuft, wie viele andere Aktionen. Das Ergebnis kann stark von früheren Optimierungszyklen beeinflusst sein. In der Regel wird deshalb der optimale Zustand bei *RBAP* nicht erreicht werden. Im optimalen Zustand ist *RBAP* dem FetchExpression-System deutlich überlegen, sowohl was die Anzahl der Anfragen als auch die Menge der zurückgelieferten Objekte anbelangt. Der initiale Zustand beinhaltet die erwartete maximale Anzahl an Anfragen für *Test 3*, da noch keine Unterstrukturen am Anfang mitgeliefert werden. Die Zahl der zurückgelieferten Objekte kann sehr stark schwanken, je nachdem, wie viele Datenwerte an Knoten und Links bei Start des Tests im Prefetch-Modell als mitzuliefern markiert waren.

4.2 Anpassungsfähigkeit

Eines der wichtigsten Merkmale des neuen Verfahrens ist die Fähigkeit, sich adaptiv an Anfragedaten und Modellierung anzupassen. In Tabelle 5 finden sich Messwerte, wie sich das Prefetch-Verhalten für die drei Testfälle über die Zeit ändert. Als Ausgangspunkt dient immer ein initiales Prefetch-Modell ohne Daten. Deutlich zu sehen ist, dass bereits nach dem ersten Optimierungslauf eine deutliche Verbesserung des Prefetch-Verhaltens eintritt. Nach drei Durchläufen ist das Prefetch-Modell in der Regel optimal angepasst. Dies hat damit zu tun, dass im ersten Durchlauf bereits die eindeutigen Änderungen durchgeführt werden und meist nur noch Anpassungen an bereits modifizierten Knoten übrig bleiben, die aufgrund der Datenlage und fortgeschrittenen Generationen nicht berücksichtigt wurden. Obwohl in den Testfällen hauptsächlich Strukturen hinzugefügt werden, kann es durchaus vorkommen, dass sie im späteren Lauf wieder entfernt werden. Die Testfälle haben also nicht rein additiven Charakter.

Je nach Leistungsfähigkeit des Clients ist es möglich, dass sich die konkreten Zahlen für das Prefetch-Verhalten geringfügig ändern. Da der Lernvorgang asynchron verläuft, ist es möglich, dass er aufgrund der Auslastung des Clients erst später durchgeführt wird als bei Rechnern, die noch Kapazitäten frei haben. In seltenen Fällen kann es vorkommen, dass sich aufgrund dieser Effekte das finale Prefetch-Modell geringfügig unterscheidet. Allerdings halten sich die Schwankungen in engen Grenzen. Auf fünf getesteten Rechnern unterschiedlicher Leistungsklasse liefert *RBAP* jeweils ähnliche Ergebnisse und einen ähnlichen Verlauf der Prefetch-Optimierung.

5 Bewertung und Ausblick

Das regelbasierte, adaptive Prefetching erfüllt die Erwartungen und scheint zumindest nach den ersten Testläufen in der Lage, als automatisiertes Verfahren die manuell erstellten FetchExpressions vollwertig und teilweise sogar besser zu ersetzen. Obendrein besitzt es auch ein größeres Repertoire an Möglichkeiten, wie zum Beispiel die flexible Anpassung an neue Datenmodellierungen ohne weiteren Aufwand und die Fähigkeit sich an das individuelle Verhalten des jeweiligen Benutzers anzupassen.

In der Regel ist die zurückgelieferte Datenmenge besser an das eigentliche Ziel der Anfrage angepasst und die Anzahl der gestellten Anfragen für ein bestimmtes Problem ist oft sogar geringer als bei FetchExpressions. Bei statischen Aktionen in einem eigenen Gültigkeitsbereich (zum Beispiel Produktionen) wird dieses Optimum auch erreicht, bei dynamischen Aktionen liegen die Ergebnisse in der Regel unter dem Optimum – je nach Art und Beschaffenheit der Anfragen und der konkreten Ausprägung des Prefetch-Modells. Trotz eines erhöhten Rechenaufwands im Client ist gegenüber FetchExpressions keine erhöhte Laufzeit festzustellen, wenn Effekte durch Latenzen und Datenübertragung weitgehend ausgeschlossen werden.

Das Softwarepaket, welches im Laufe der Untersuchungen entwickelt wurde, erfüllt somit weitgehend die notwendigen Anforderungen für den Einsatz in *SCHEMA ST4*. Allerdings gibt es durchaus noch Ansätze, wie das Verfahren bzw. die Software verbessert werden kann:

- *Echte Multi-Level-Optimierung*
  RBAP ist durch den Optimierungsblock festgelegt auf die Reichweite einer Knotenebene. Aus der Praxis zeigt sich, dass viele Aktionen Einfluss auf mehrere Knotenebenen nehmen können. Durch eine Erweiterung des Optimierungsblocks über weitere Knoten hinweg könnte eine Verbesserung der Qualität der Lernverfahrens erreicht werden. Allerdings steigt gleichzeitig auch die Komplexität und der Aufwand für die Berechnungen.

- *Reduktion der Rechenzeit im Client*
  Gerade um komplexere Verfahren zu realisieren, müssen Möglichkeiten gefunden werden, die Rechenzeiten in Grenzen zu halten.
  Ein Ansatzpunkt ist die feste Definition dynamischer und statischer Gültigkeitsbereiche. Im ersten Fall ist eine Änderung der Anfragestruktur jederzeit möglich und der Lernvorgang muss immer ausgeführt werden. Das Prefetch-Verhalten für statische Aktionen (wie zum Beispiel Produktionen) müsste hingegen nur einmal erlernt werden. Sofern sich in der Datenmodellierung und am Ablauf der Aktion nichts ändert, muss kein weiterer Lernvorgang ausgeführt werden.
  Außerdem sollte es möglich sein, den Lernvorgang so umzugestalten, dass eine statistische Auswertung ermöglicht wird. Dabei werden nicht mehr alle Anfragen registriert, sondern im Durchschnitt nur jede *n*te Anfrage.

- *Erweiterte Lernmöglichkeiten*
  In Kombination mit einer intensiven Offline-Analyse von Protokolldaten sollte es möglich sein, bisher manuell festgelegte Erfahrungswerte, zum Beispiel Schwellwerte für die Optimierung oder das Standardregelwerk, über einen Lernprozess zu definieren, um möglichst optimal ans jeweilige System angepasste Regeln zu erhalten.

- *Temporäre Gültigkeitsbereiche*
  In etlichen Fällen ist es wünschenswert, Daten für Aktionen, die in mehreren Gültigkeitsbereichen ablaufen, zu exportieren. Anstatt für diesen Export eine neue Aktion und einen neuen Gültigkeitsbereich zu definieren, könnte ein kombinierter Gültigkeitsbereich (*Combined Scope*) geschaffen werden, der als Prefetch-Modell die additive Hülle aller beteiligten Gültigkeitsbereiche erhält. Somit würde ein erneuter Lernprozess entfallen und die Aktion würde von sämtlichen Verbesserungen in jedem einzelnen Gültigkeitsbereich profitieren.
  Das Gegenteil stellt das Konzept der *temporären Differenz von Gültigkeitsbereichen* (*Delta Scopes*) dar. Bei Wechsel von einem Gültigkeitsbereich *G* zu einem anderen Gültigkeitsbereich *H* kann es vorkommen, dass sich Daten von angefragten Objekten bereits im Cache befinden. Wird ein Cache-Miss auf einem Datenwert unter *H* registriert, obwohl der zugehörige Knoten sich im Cache befindet, kann durch eine Übertragung der Differenzmenge von *G* und *H* eine effektivere und sparsamere Prefetch-Menge für die Anfrage geliefert werden als durch reinen Prefetch mit Gültigkeitsbereich *H*.

Literatur

[Bengel, 2002] Günther Bengel. *Verteilte Systeme – Client-Server-Computing für Studenten und Praktiker*. Vieweg, 2nd edition, Jan. 2002. ISBN 3-528-15738-0.

[Consortium, 1999a] World Wide Web Consortium. XML Path Language (XPath) Version 1.0, 11 1999. `http://www.w3.org/TR/xpath`.

[Consortium, 1999b] World Wide Web Consortium. XSL Transformations (XSLT) Version 1.0, 11 1999. `http://www.w3.org/TR/xslt`.

[Coulouris *et al.*, 2002] George Coulouris, Jean Dollimore, and Tim Kindberg. *Verteilte Systeme – Konzepte und Design*. Addison-Wesley, 3rd edition, Jan. 2002. ISBN 3-8273-7022-1.

[Knafla, 1999] Nils Knafla. *Prefetching Techniques for Client/Server, Object-Oriented Database-Systems*. PhD thesis, University of Edinburgh, 1999.

[Liberty, 2003] Jesse Liberty. *Programming C#*. O'Reilly, 3rd edition, 2003. ISBN: 0-596-00117-7.

[Padmanabhan and Mogul, 1996] Venkata N. Padmanabhan and Jeffrey C. Mogul. Using predictive prefetching to improve World-Wide Web latency. In *Proceedings of the ACM SIGCOMM '96 Conference*, Stanford University, CA, 1996.

[Process, 2005] Java Community Process. JSR 243: Java$^{\mathrm{TM}}$Data Objects 2.0 – An Extension to the JDO specification (Proposed Final Draft), Aug. 2005.

[Schäffer, 2003] Bruno Schäffer. Durch dick und dünn – Probleme bei Thin-Clients. *JavaSPEKTRUM*, Jan. 2003.

[Teng *et al.*, 2005] Wei-Guang Teng, Cheng-Yue Chang, and Ming-Syan Chen. Integrating web caching and web prefetching in client-side proxies. Technical Report 15, IEEE Transactions on Parallel and Distributed Systems, May 2005.

Incremental Mining for Facility Management

Katja Hose Marcel Karnstedt
Daniel Klan Kai-Uwe Sattler
Department of Computer Science and Automation,
TU Ilmenau, Germany

Jana Quasebarth
Research and Development,
NT Neue Technologie AG, Germany

Abstract

Modern buildings are equipped with high-tech systems that take care of several fundamental aspects, e.g., air-conditioning, heating and water supply. The requirements posed on facility management by such buildings are challenging. Modern techniques implement adaptive control systems to achieve this, in which decisions are preferably based on the results of (multiple correlated) mining tasks on recently gathered sensor data. In this work, we discuss the general relationship between such control systems and the underlying mining tasks. We exemplary choose change detection in the context of pattern analysis as a representative, because this mining task involves general requirements known from stream processing like the need for incremental algorithms, but also poses specific challenges like in-time detection. We present three concrete approaches for this and an according evaluation.

1 Introduction

Because of the growing number of installed systems within modern office buildings configuring all these systems manually is difficult or even impossible. The interaction between those systems is rarely considered. State-of-the-art controllers provide primitive functions for controlling the cooperation of different devices. However, the components are often adjusted and optimized manually. In many cases systems run suboptimally with factory or initial settings made at installation time. Because of the complexity of system functionalities and correlations, not only between the system components themselves but also between the system and external factors such as the weather, expert knowledge is required to change system settings properly. With respect to rising energy costs and pollution control, there is a growing interest in (automatically) optimizing the operation of buildings.

Finding a configuration that reconciles the goals of all concerned parties is rather complicated and depends on many influencing factors. Consequently, a solution that does this (semi-)automatically and reacts upon events such as changes in the price of electricity is highly desirable. In summary, the main objectives of an intelligent facility management are

- minimization of maintenance and operation costs
- minimization of ecological costs
- improving or at least preserving user convenience
- forecasting of operation costs and the environmental impact of the system under different premises (what-if-analyses)

As modern buildings are equipped with many installations, e.g., heating, air-conditioning, etc, there are many sensors and actuators that continuously produce streams of readings. A good idea in this context is to apply methods known from data mining and signal processing. This may involve burst detection, classification as well as detecting correlations between different sensors. As sensor data is usually generated continuously at a rapid rate, approaches from stream mining are becoming the focus of interest. Especially for discovering correlations which are unknown or even never supposed to exist, methods for pattern recognition promise to be very effective. An item can be seen as a combination of a sensor ID and the (optionally discretized) sensor value, an itemset as a set of items collected at (again, optionally discretized) times. For instance, $\{s1, x20, y18\}$ may encode the sensor information “sun is shining, temperatur x is 20, temperatur y is 18”. Applying this scheme, different itemsets may be formed by combining different sensors (multiple times), depending on application specific conditions. If multiple sensor data are combined and encoded to form itemsets, frequent itemset mining is applied as a basis for frequent pattern detection and association rule mining. A pattern (resp. itemset) is frequent in a stream of transactions, if it occurs in more than σ percent of the actual stream. Here, σ is called the required *support* of the itemset. With, usually approximating, stream mining methods, the actual support of itemsets declared as frequent may vary by a provided *error bound* ϵ. We use the term transaction to refer to a set of itemsets, which merely stems from the popular application field of customer basket analysis. An outstanding requirement of automated facility management solutions is the capability of in-time detection of changes and according reaction. This also holds for the detection of changes in correlations symbolized by frequent patterns. Thus, efficient and in-time change detection is a field of high interest in this sector, but almost unexplored. The chosen field of application implies the incremental character of appropriate methods as a major requirement. As a second main requirement for automated facility management, we expect any algorithm to support querying arbitrary time intervals. Beside the fast reaction to changing situations, this allows for analyzing and reasoning about sensor correlations belatedly, e.g., in order to explain problems that occurred or to optimize settings for the future. In this work, we present three different approaches for solving this exemplary task.

This paper is organized as follows. In Section 2 we describe the idea of automated facility management and discuss the importance of data mining in this context, as well as a general architecture for such systems. In Section 3 we briefly present the frequent itemset mining algorithm applied for pattern detection, and afterwards propose three concrete methods for change detection on these frequent patterns. Related work is covered in Section 4, whereas Section 5 includes exemplary results from an extended evaluation. Finally, we conclude in Section 6.

2 Automated Facility Management

This section gives an overview of the main challenges and application scenarios in automated facility management. Furthermore, we sketch the architecture of such an environment and point out in what aspects data mining is beneficial.

2.1 Application Scenarios

Nowadays, facility management is no longer a simple matter. First, various autonomous systems have to be calibrated individually. Second, these systems should be configured in a way that enables them to work together efficiently. A worst case scenario for example would be a situation where the air-conditioning cools the air while radiators are warming it. However, the usual case is not that extreme but there is great potential for optimization in nearly all buildings. As initial situation we assume each building to have a number of autonomous installations (air-conditioning, heating, automatic rolling shutters, lights, water, etc.). In general, these systems are programmed with primitive functions without consideration of much additional information.

Optimizing a single installation already may be challenging. Let us take a conventional boiler heating system as an example. Heating installations can be programmed with primitive functions such as “reduce operation at night” or “do not operate the radiator heating circuit in summer, just heat drinking water”. Unfortunately, transition from winter to summer operation mode and vice versa is either initiated by the user or determined by a fixed date disregarding the actual weather conditions. A more intriguing solution would be to consider weather forecast information. However, given the information that it is going to be a rather warm day, maybe heating the rooms will not be necessary at all. Such considerations are particularly interesting for transition times from one season to another and are closely related to the consumption of resources and to costs.

Second, users mostly have to specify the times for starting reduced or standard operation instead of specifying the times of utilization. Central heating installations are rather slow systems. Thus, they have to start to increase their flow temperature and turn on the pumps supplying the radiator heating circuit sometimes over two hours in advance in order to provide the set temperature for the time of utilization. This often results in a long-term manual trial-and-error approach until the desired comfort is reached and the right start and end times are found.

Third, conventional heating installations consider outdoor temperature using sensors and adapt the flow temperature according to the outdoor temperature. This is done by a simple characteristic curve relating the outdoor temperature to a specific flow temperature and a controller regulating the flow temperature. This curve can be shifted by a parallel translation and its inclination can be changed. Settings in this context depend on the building characteristics, but they have impact on comfort and costs. However, such functions are usually static in conventional installations – once manually defined by the installer and never adapted.

Altogether, an automated facility management system receives input from a considerable quantity of sensors (flow temperature of the heating installation, inside and outside temperatures, etc.) and actuators (burner, pumps, valves, ventilators, etc.) providing a vast amount of *process data* describing current states and actions. However, current installations mostly work completely independent from each other. Thus, air-conditioning and heating do not know from each other. Besides, since the air-conditioner is in most cases also capable of warming air instead of only cooling it, it might be interesting to combine this heating capacity with the actual heating installation. Especially with respect to energy costs (electricity, heating oil) the economic optimum might change frequently and is not easy to find. Furthermore, aspects such as physical well-being, room properties, and condition (room isolation, room size, etc.) are not considered. As long as the systems do not interact with each other or consider additional information, coherences and dependencies cannot be identified and consequently an optimal solution cannot be found.

2.2 System Architecture

The architecture for automated facility management that we propose consists of a large number of sensors/actuators, a controller, and the facility management system. It is shown in Figure 1.

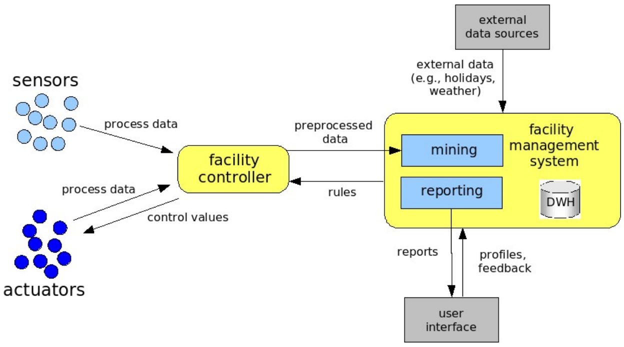


Figure 1: System architecture

As stated above, current buildings accommodate different autonomous installations such as heating or air-conditioning systems. In general, installations may have actuators as well as sensors. These sensors record process data such as the current flow temperature of the heating installation. In addition to the sensors integrated into these installations, further sensors are used to obtain additional information about parameters specific to a particular building, e.g., air humidity, current room temperature, movement, and light. Both types of sensors as well as actuators produce a continuous data stream.

The facility controller collects these readings, preprocesses them and forwards the results to the facility management system. Within the preprocessing step the controller already runs a plausibility test on the input sensor and actuator readings and notifies the facility management system about possibly defective sensors. Each building has its own facility controller. On the contrary, facility management systems are not specific to individual buildings. Multiple controllers can communicate with one global facility management system. In order to reduce transmission costs only changes in the actuator and sensor readings that are detected by the facility controller are sent to the facility management system. Such a data item is a triple (s, v, t), where s denotes the sensor resp. actuator id, v denotes the measured value, and t denotes a time stamp that indicates when the values were measured.

The facility management system generates rules for the controller based on the current process data, user profiles, and external metadata such as holidays, the daily weather report, or the current tariffs for electricity, water, or gas. Finally, the controller can use these rules to generate control values for the actuators. In addition to the output rules the knowledge management system generates reports [Motegi and Piette, 2002] for the users. Such reports represent information about which installation causes what costs and may serve as a basis for decision-making with respect to comfort settings and costs. The high number of sensors and actuators results in a continuous data. Storing all these tuples into a database is impossible.

2.3 Mining Tasks

The applications of data mining in automated facility management are manifold. We need classification and clustering as well as algorithms for creating association rules.

An example for the application of classificators is to decide whether the current room temperature conforms to the user well-being. Sometimes this is rather simple, 15°C in an office where people used to work is definitely too cold. However, that temperature is still acceptable for an office that is momentarily unused (e.g., because of holidays). Furthermore, in cases of sudden increase of temperature and heat the system should detect that change and for example raise an alarm since this could mean fire. Furthermore, classification strategies allow us for instance to define classes of actuator readings whose combination should be avoided (e.g., air-condition systems and heating systems running at the same time). We can also use the classifier to identify substitution classes. Then, the rule generator can use the substitution classes to define different configurations with the same effect but different costs. Clustered sensor values within a room possibly indicate that the sensor concentration within the room is too high.

Association rules can be used to detect new coherences between sensors, actuators, and user preferences resp. maintenance costs. Based on the sensor readings that have been sent to the facility management system, buffer resp. window techniques are applied and a stream of itemsets is formed. On this stream frequent itemsets are determined. Based on these itemsets association rules are determined, e.g., if the sun shines the temperature sensor that is located at the window of a room measures 2 degrees more than the temperature sensor at the door (located several meters away from the window). Assume there is an event that changes this coherency such that the difference between the two sensors is only 1 degree. The reason might be that blind has been lowered.

Operating facility management systems when knowing about all correlations of dimensions (sensor/actuator reading, energy consumption) is straightforward. However, it is a challenging task to detect correlations between arbitrary dimensions that are not obvious and thus not known at the initial start-up of the facility management system. In order to detect such correlations we propose the use of frequent itemset mining. However, it is not enough to detect such correlations since they can change over time. In some cases it is necessary to react immediately upon such changes. This is the problem that we focus on in this paper: (incremental) change detection of frequent itemsets.

3 Detecting Frequent Patterns in Data Streams

Frequent itemset mining deals with the problem of identifying sets of items occurring together in so-called transactions frequently. Basically, two classes of algorithms can be distinguished: approaches with candidate generation (e.g., the famous apriori algorithm [Agrawal *et al.*, 1993; Agrawal and Srikant, 1994; Manku and Motwani, 2002]) as well as without candidate generation. Here, only the latter ones are suitable for stream mining. Usually, these approaches are based on a prefix-tree-like structure. In this tree – the frequent pattern (FP) tree – each path represents an itemset in which multiple transactions are merged into one path or at least share a common prefix if they share an identical frequent itemset [Han *et al.*, 2000]. For this purpose, items are sorted as part of their transaction in their frequency descending order and are inserted into the tree accordingly. Again, the FP tree is used as a compact data summary structure for the actual (offline) frequent pattern mining (the FP growth algorithm).

In order to mine streaming data in a time-sensitive way an extension of this approach was proposed [Giannella *et al.*, 2003]. Here, so-called tilted time window tables are added to the nodes representing window-based counts of the itemsets. The tilted windows allow to maintain summaries of frequency information of recent transactions in the data at a finer granularity than older transactions. The extended FP tree, called pattern tree, is updated in a batch-like manner: incoming transactions are accumulated until enough transactions of the stream have arrived. Then, the transactions of the batch are inserted into the tree. For mining the tree a modification of the FP growth algorithm is used taking the tilted window table into account. The original approach assumes that there is enough memory available to deliver results in the given quality and no way is described how to proceed if the algorithm runs out of memory.

As frequent itemset mining algorithms on data streams usually produce approximate results, there may be some false positives in the resulting output. Therefore, we need an algorithm that guarantees an error threshold. Additionally, the approach has to be time-sensitive. The FP-Stream approach in [Giannella *et al.*, 2003] is capable to satisfy these requirements. Asked for the frequent itemsets of a time period $[t_s, t_e]$, FP-Stream guarantees that it delivers all frequent itemsets in $[t_{s'}, t_{e'}]$ with frequency $\geq \sigma \cdot W$, where W is the number of transactions the time period $[t_{s'}, t_{e'}]$ contains. $t_{s'}$ and $t_{e'}$ are the time stamps of the used tilted time window table (TTWT) that correspond to t_s and t_e, depending on Q_{Tg}. The result may also contain some itemsets whose frequency is between $(\sigma - \varepsilon) \cdot W$ and $\sigma \cdot W$.

[Franke *et al.*, 2005; 2006] presents a modified version of this algorithm, which was designed to be resource- and quality-aware in parallel. The main contribution of this work is the identification of parameters the resource consumption of the algorithm is sensitive to. Based on this, it proposes when and how to change these parameters in order to meet given resource limits, while adhering to specific output quality requirements.

The focus of this paper is the application of the frequent itemset mining algorithm from [Franke *et al.*, 2005; 2006] with priority on a in-time (subsequent or parallel) change detection. The incremental approaches for such a change detection are presented in detail in the next subsection.

3.1 Incremental Change Detection

In [Franke *et al.*, 2006] we already discussed general approaches for incremental change detection on data streams, particularly in the context of frequent itemset mining. We reasoned about the challenges and inferential requirement for this task, and identified two general approaches: *(i)* approaches on separate data structures, and *(ii)* approaches operating directly on the pattern tree. Beside particular pros and cons for each, which will be discussed along with the corresponding algorithms later on in this section, they are characterized by a major difference: methods following the first approach will be represented by separate operators in a complex mining task, they are processed on the output of a preceding frequent itemset operator. Methods of the second class are integrated in these operators, i.e., change detection is processed in parallel to the mining for actual frequent itemsets.

The first method, called *CT* (*C*hange *T*able), stores all frequent itemsets produced so far together with additional information (e.g., temporal) in a table. Each row of that table represents one frequent itemset. This is a naive, but effective approach. The content of that table can be queried for changes in arbitrary time intervals. This allows for detecting a wide variety of changes, even temporal ones. Unfortunately, the task of detecting changes in one specific itemset (e.g., itemset $\{a1, a3, c2\}$ changes to $\{a1, a3\}$ or $\{a1, a3, b2\}$) is complicated, because there is no information about the location of itemsets in the pattern tree if they are registered after being output from the frequent itemset operator. Thus, all itemsets in that table must be compared, which may consume a lot of computing time. Of course, this data structure allocates memory in parallel to the frequent itemset operator, thus, resources must be shared between both. However, the resource-adaptive techniques introduced in [Franke *et al.*, 2006] could be applied to this data structure in order to meet given resource limits. Other approaches on separate data structures have not been implemented yet, as they promise similar characteristics and challenges in their handling. The CT method is simple, but effective, and thus, representative for this class of change detection methods. The incremental processing of input data is implied in this case, by means of the output frequency of the preceding mining operator. Each time a new set of frequent itemsets is output and inserted in the table, we can detect simple as well as sophisticated (see [Franke *et al.*, 2006] for a more detailed explanation) changes for each single itemset – if new or already inserted before. This is incremental, but with rising table size very expensive.

A maybe more intuitive idea is to try to detect changes directly from the pattern tree used in the frequent itemset mining operator. Beside the expected improvement in the reaction time, this would allocate no extra memory. We implemented two methods based on this idea: *CDM* (*C*hange *D*etection during *M*anipulation) detects changes as soon as the pattern tree is modified, and *CDFISM* (*C*hange *D*etection during *F*requent *I*temset *M*ining) detect changes after executing the FP-growth algorithm on the pattern tree (when looking for the actual frequent itemsets after each batch). We have to consider three different modifications:

- a new node is inserted
- a node is deleted
- the TTWT in a node is changed

Moreover, we use the information stored in the TTWTs in order to detect changes over arbitrary time intervals. Note that the insertion of a node does not urgently reflect in a change of the frequent itemset. This happens not until the required support is achieved, which is signalized by modifying the according TTWT entries.

CDM is the method which promises the best reaction time, which is intuitively a major requirement for change detection in data streams, especially for the aspired automated facility management. But, we can only detect changes in itemsets that are modified in the current batch run. If a change in an itemset is only recognized during the run of FP-growth, the CDM method will not detect this change, but the CDFISM method will. If the TTWT entries of an existing node are modified, we always have to compare the current frequency with the former ones. In analogy, the deletion of a node does not urgently reflect in the drop of a frequent itemset. Again, more detailed comparisons have to be made. Due to its nature, querying time intervals which do not start at the current point in time is senseless, as manipulations in former time steps cannot be reconstructed. With the CDM method we may signalize false positives (so called *change candidates*), because changes are detected before a batch is finally processed. This incremental character allows, in parallel, for a very fast change detection.

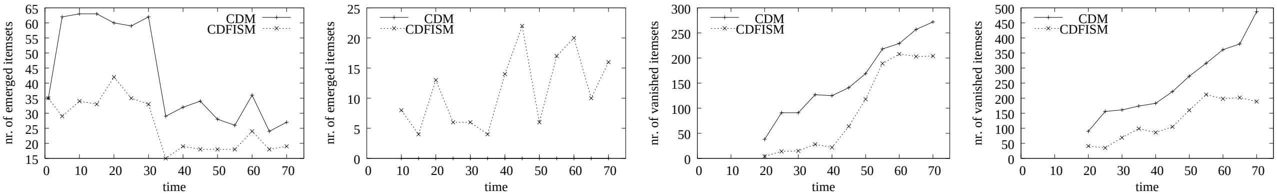
The disadvantage of CDFISM is that it can only be run after processing a whole batch and has to completely traverse the tree – which leads to worse reaction time and less information about new or deleted nodes. Thus, incremental processing is bounded to the size of gathered batches. But inserting single transactions into the pattern tree is not suitable, due to the principles of the FP-Stream algorithm. For details we refer to [Giannella *et al.*, 2003; Franke *et al.*, 2005; Franke, 2006]. Note that nevertheless the size of a batch may vary, as we do not have to collect a sudden amount of transactions to complete a batch, but may also refer to a sudden interval of time. The CDFISM method is also suited for querying time intervals that do not start at the current point in time, because it does not depend on the currently modified parts of the pattern tree. Additionally, there are no change candidates. In contrast, the method cannot detect nodes deleted from the pattern tree Advantages of both approaches, in contrast to those on separate data structures, are low runtime, easy detection of temporal changes (the TTWTs containing temporal information can be analyzed directly) and no extra memory consumption. A common disadvantage is the limited temporal quality, because it depends on the pruning steps based on the TTWTs.

After a theoretical analysis of the algorithms, we expected the actual choice of algorithm depending on the goals actually desired by the user and the characteristics of the data stream. These expectations have been substantiated by the experiences of an evaluation. These results are presented in Section 5.

4 Related Work

Optimization of building automation systems and decrease of resource consumption are of interest for politics, research and development since the last decade.

In [Garces *et al.*, 2006; Österlind *et al.*, 2007] wireless sensor networks are proposed as a feasible solution



(a) $\sigma = 0.03$, time 0,1 (b) $\sigma = 0.05$, time 6,12 (c) $\sigma = 0.03$, time 7,20 (d) $\sigma = 0.05$, time 7,20

Figure 2: Comparison of CDM and CDFISM

q_s	q_e
0	1
0	10
6	12
7	20
10	30

(a) Queried times

Support σ	**Error** ϵ
0.02	0.002
	0.008
0.03	0.003
	0.012
0.04	0.004
	0.016
0.05	0.005
	0.02

(b) Values for support and error bound

Figure 3: Parameter values used in tests

to gain additional information for building automation systems. By means of such sensor networks room status and changes can be monitored continuously. In fact, this is a precondition for a continuous optimization of user convenience and reduction of costs and resource consumption.

Tagging fault sensor data is a problem to deal with in a preprocessing step before actually analyzing the data. In [Kolokotsa *et al.*, 2005] the detection of fault sensor data is especially investigated for building energy management systems, while [Klein *et al.*, 2007] follows a more general approach for streaming or static data.

A very good survey over models and problems in the field of data stream mining provides [Babcock *et al.*, 2002]. This work lists several other contributions dealing with specific as well as general mining tasks on data streams. Change detection is not an explicit focus of this survey, but is covered by other proposals. [Chakrabarti *et al.*, 1998; Ganti *et al.*, 2001; 2002] refer to so-called “evolving” data and discuss the detection of changes in these. [Chawathe *et al.*, 1998] deals with change detection on semi-structured data. The efficient detection of bursts is discussed in [Zhu and Shasha, 2003; Kleinberg, 2003]. [Zhu and Shasha, 2003] apply a wavelet-based approach for change detection on data streams, [Kleinberg, 2003] deals with so-called “word bursts”, i.e., the occurrence of frequent words. In [Aggarwal *et al.*, 2003; Aggarwal, 2003] a framework for change detection is proposed which uses spatial distance estimations and places boosted attention on heuristics to detect trends, rather than applying formal statistical models. [Kifer *et al.*, 2004] is one of the few works explicitly discussing the detection of sophisticated changes and how to present them to the user in a preferably intuitive way.

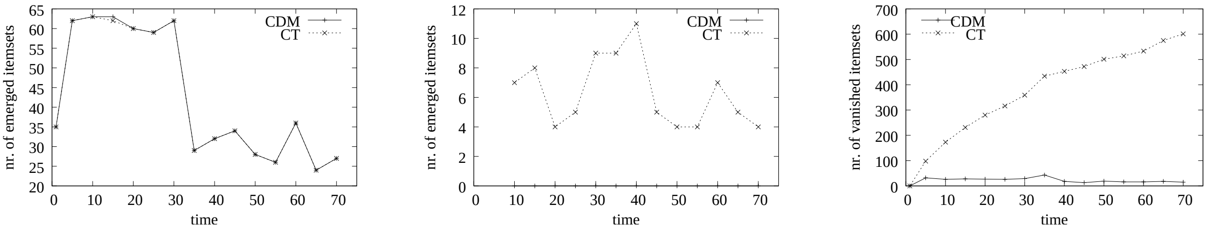
Beside the approaches developed by database researchers, there are several proposals from the machine learning community related to our work. [Widmer, 1997] deals with incremental (on-line) meta-learning and applies different classifiers in order to detect changing contexts of concepts. The guesses for concept changes based on these contexts changes are related to association rule mining. This work is based on the notion of concept drift, closely related to what we identify as changes, which occurred first in [Schlimmer and R. H. Granger, 1986]. The authors of the rather recent work [Scholz and Klinkenberg, 2007] propose boosting classifiers taking drifting concepts in account. Change detection in the context of regression is researched as well, e.g., by [Herbster and Warmuth, 1998] which deals with the problem of finding best regressors by lifting static bounds to shifting bounds. Two works covering change detection in the context of pattern analysis are [Rozsypal and Kubat, 2005] and [M.-Ch. Chena and Chang, 2005]. While [M.-Ch. Chena and Chang, 2005] focuses on the application of change detection for customer relationship management, [Rozsypal and Kubat, 2005] presents an apriori-based algorithm working on windows and blocks which we see as unqualified for our needs. Particularly, the authors aim for detecting context changes on the basis of single basic changes in the itemsets, rather than signalizing each single change, which may be basic or sophisticated.

Most of the existing works focus on detecting one specific kind of change, preferably on detecting changes in the distribution of the stream elements. Rather often they are based on statistical models and functions. The detection of changes over different time intervals or sophisticated changes in sets of elements is rather unexplored. Especially in the specific field of pattern mining we do not know about any proposals providing approaches for detecting basic and sophisticated changes, neither in parallel to the actual mining nor as a subsequent processing step. We extend existing ideas by these aspects and include time-sensitiveness, change presentation and the crucial question of *what* changed in detail (in contrast to only signalizing the pure *occurrence* of changes) into our considerations. Moreover, to the best of our knowledge, we are the first applying change detection in the context of facility management. Last but not least, our whole framework is especially designed for achieving quality- and resource-aware stream mining.

5 Evaluation

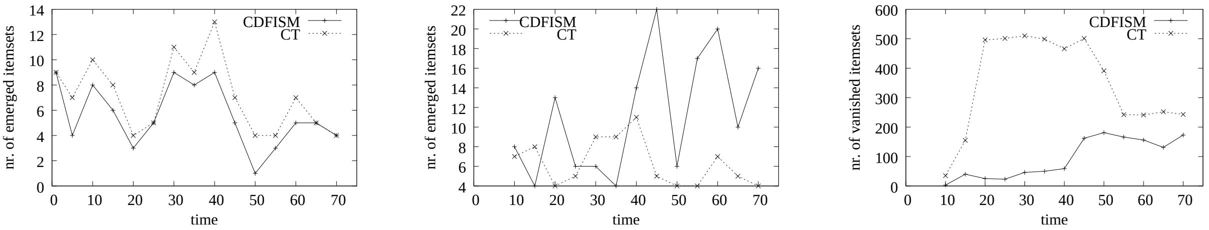
In this section we compare the three introduced approaches for incremental change detection. The following results are taken from [Fauth, 2005], which was finished under the chair of the database group at TU Ilmenau. The main purpose of this evaluation is to substantiate the advantages and disadvantages of each method, and the dependence of these on the specific requirements of the user as well as the characteristics of the data stream. Moreover, we five first directions for choosing the appropriate approach in the right situation.

Due to the interim lack of representative facility data, we ran our tests on data generated from IBM’s popular pattern



(a) $\sigma = 0.03$, time 0,1 (b) $\sigma = 0.05$, time 6,12 (c) $\sigma = 0.03$, time 0,1

Figure 4: Comparison of CDM and CT



(a) $\sigma = 0.05$, time 0,1 (b) $\sigma = 0.05$, time 6,12 (c) $\sigma = 0.03$, time 6,12

Figure 5: Comparison of CDFISM and CT

generator Quest [IBM, 2005]. Currently, we are preparing the collection and processing of such data in a multifaceted project cooperation with industrial as well as scientific institutions. For details about the used data generator, we refer to [Ramesh *et al.*, 2005]. For the purpose of this evaluation, it is sufficient to know the main characteristics of the produced data streams: As proposed by [Ramesh *et al.*, 2005], the generated data follows a Poisson distribution. For representing the used data sets, we apply the same notion as generated by the IBM tool: $Tx.Iy.Dz$ symbolizes a data stream with z transactions of average size x and average length of maximal frequent itemsets y. For instance, $T3.I4.D1000K$ refers to a data stream containing one million transactions with 3 elements per transaction in average, the average size of the maximal frequent itemsets is 4. In the presented experiments, every five seconds one transaction is generated.

The main goals of this evaluation are to analyze:

1. the completeness of detected changes
   - with reference to different queried time intervals
   - with reference to different values for the support σ and the error bound ϵ
   - with reference to different data streams
2. the exactness with reference to false positives resp. change candidates
3. the memory consumption

We chose to query changes from time intervals $[t_s, t_e]$ shown in Figure 3(a). Interval $[0, 10]$ means we query all changes between the current point in time (0) and 10 seconds before. Figure 3(b) shows the values for σ and ϵ used in each interval. As illustrated, we switch between an ϵ value of 10% and 40% of the support value. We used generated data streams $T3.I2.D1000K$, $T3.I4.D1000K$ and $T4.I2.D1000K$. From all tests run, we present and discuss selected results which are particularly representative. We spotlight on directly comparing the introduced algorithms with respect to their ability of detecting basic changes. In the following figures, the number of emerged itemsets refers to those itemsets detected as being additional compared to prior sets. In analogy, the number of vanished itemsets refers to those itemsets the particular algorithm detected as being missing compared to prior sets.

In the first series of experiments, we compare the three approaches mutually pairwise. The following results are all gathered from runs on the $T3.I4.D1000K$ data stream using an error bound ϵ of 10%. Figure 2 illustrates differences between CDM and CDFISM. In Figure 2(a) both approaches on the pattern tree conform closely in the number of detected new itemsets. CDFISM detects slightly less itemsets, because new inserted nodes cannot be determined as new. Figure 2(b) reveals significant differences. This shows that the CDM method is only suited for detecting current changes. Analyzing past time intervals is rather poorly supported by this method. Figure 2(c) and Figure 2(d) however exemplary show close conformance for different support values when detecting dropped itemsets. In all cases, the CDM method is capable of finding a larger amount of itemsets. This is due to the possibility to recognize the deletion of nodes in the pattern tree, which is not possible using the CDFISM method.

In further tests both methods revealed to perform rather bad when querying large time intervals. This is primarily due to the summarizing character of TTWTs in the pattern tree, which also results in false positives with both methods.

In Figure 4 we compare the CDM and CT methods. Figure 4(a) shows a case where both methods behave almost identical, whereas in Figure 4(b) we illustrate a case where only the CT method is suited to continuously detect changes. Figure 4(c) shows that the number of dropped itemsets detected by the CT method increases linearly, while that of the CDM method stays constant. A case where both methods conform close in the number of detected dropped itemsets could not be found during all our tests.

In the series of comparisons, we finally illustrate the relationship between CDFISM and CT in Figure 5. Again, Figure 5(a) and Figure 5(b) each show a case of good and bad conformance. Remarkable are the peaks of the CDFISM method in Figure 5(b). Figure 5(c) illustrates the capabilities of detecting dropped itemsets. Obviously, the CT method is capable of detecting more dropped itemsets, particularly in the time between 20 and 45. From time 50 upward both plots slowly approach each other again. Similar to the CDM method, we could not find any case where the CDFISM method was suited for detecting a similar amount of dropped itemsets as the CT method.

The results gained using the CT method show the improvements gained by affording more memory, which are as expected. By storing and analyzing all itemsets found so far, it is capable of detecting more changes than the methods on the pattern tree. Of course, without pruning this method could easily break resource limits. Due to the detection of changes by directly comparing (sub)sets, the CT method implicitly signalizes more changes, even subsets that were never contained in the pattern tree. Ignoring these itemsets, the CT method still detects slightly more changes than the CDM and CDFISM methods..

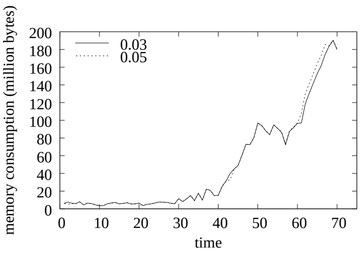


Figure 6: Memory FPStream for different σ

Beside the exposed differences in the particular capabilities of detecting different kinds of changes in different situations, the methods also differ in performance aspects and resource requirements. Figure 6 shows the memory consumption of the actual pattern tree for different support values, Figure 7 shows the same for the CT method. Note that the memory for the second approach is needed in addition to that allocated for the pattern tree. This memory requirement is almost equal for both support values, and significantly increases in the later time steps. Figure 7 reveals constantly increasing memory requirements of the CT method, depending on the support value

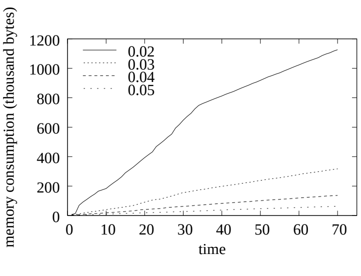


Figure 7: Memory change table for different σ

Last but no least, we compare the detection capability of all three methods on different data streams in Figure 8. These tests compare the results on the $T3.I2.D1000K$ and $T3.I4.D1000K$ data streams using a support value of $\epsilon = 0.03$ and querying time 0,1. The CDM and CT method behave very similar, whereas the CDFISM method always detects a slightly smaller amount of new itemsets.

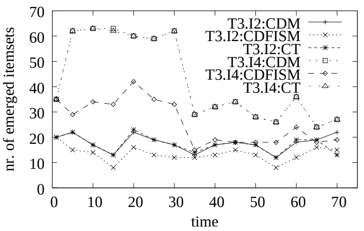


Figure 8: Varying input data

Summarizing, we can state that the CT method is suited for detecting the most changes. This goes along with high costs, especially with increasing time if no pruning technique is applied. Negative is the need for another processing step in order to eliminate wrong candidates. The methods on the pattern tree work especially good in recent time intervals, which is fine for implementing a signal and control mechanism that depends on fast reactions. All methods show no significant dependence on the actual stream characteristics or specific parameter values. Our ongoing work is to analyze the algorithms with reference to their real-time capabilities and CPU time consumption.

6 Conclusion

Facility management imposes a wide variety of challenges if it is aimed to be automated and efficient. In a setting proposed in this work, data mining approaches can be applied in an incremental manner in order to support the detection of sophisticated correlations and fast reactions. Especially for supporting fast reactions, change detection is an important field of interest. We introduced three basic methods for fulfilling this task in the context of pattern recognition. We showed that each method reveals advantages and disadvantages, and that each of the method should be preferred in specific situations. Further, we gave first directions on the essential factors influencing the optimal choice. The achieved results suggest to combine the different approaches in order to implement a reliable and flexible change detection. In future work we will extend the introduced framework for facility management and highlight further aspects. This covers other mining tasks as well as management and application challenges. Change detection will be an integral part of any such implementation. Before, we will extend the evaluation presented in this work.

Acknowledgments

We would like to thank Matthias Fauth and Conny Franke, who did great work in implementing and evaluating algorithms introduced in this work in the context of their diploma theses at the TU Ilmenau.

References

[Aggarwal *et al.*, 2003] Ch. C. Aggarwal, J. Han, J. Wang, and Ph. S. Yu. A Framework for Clustering Evolving Data Streams. In *VLDB 2003, Berlin, Germany*, pages 81–92, 2003.

[Aggarwal, 2003] Ch. C. Aggarwal. A Framework for Diagnosing Changes in Evolving Data Streams. In *SIGMOD 2003, San Diego, USA*, pages 575–586, 2003.

[Agrawal and Srikant, 1994] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB*, pages 487–499, 1994.

[Agrawal *et al.*, 1993] R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *SIGMOD 1993, Washington, D.C., USA*, pages 207–216, 1993.

[Babcock *et al.*, 2002] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proceedings of PODS 2002, Madison, USA*, pages 1–16, 2002.

[Chakrabarti *et al.*, 1998] Soumen Chakrabarti, Sunita Sarawagi, and Byron Dom. Mining Surprising Patterns Using Temporal Description Length. In *VLDB*, pages 606–617, 1998.

[Chawathe *et al.*, 1998] S. S. Chawathe, S. Abiteboul, and J. Widom. Representing and Querying Changes in Semistructured Data. In *ICDE*, pages 4–13, 1998.

[Fauth, 2005] M. Fauth. Änderungserkennung in Datenströmen, 2005. Diploma Thesis at TU Ilmenau (available in German only).

[Franke *et al.*, 2005] C. Franke, M. Hartung, M. Karnstedt, and K. Sattler. Quality-Aware Mining of Data Streams. In *IQ*, pages 300–315, 2005.

[Franke *et al.*, 2006] C. Franke, M. Karnstedt, and K. Sattler. Mining Data Streams under Dynamicly Changing Resource Constraints. In *KDML: Knowledge Discovery, Data Mining, and Machine Learning*, pages 262–269, 2006.

[Franke, 2006] C. Franke. Ressourcen-Adaptives Frequent Itemset Mining in Datenströmen, 2006. Diploma Thesis at TU Ilmenau (available in German only).

[Ganti *et al.*, 2001] V. Ganti, J. Gehrke, and R. Ramakrishnan. DEMON: Mining and Monitoring Evolving Data. *IEEE Trans. Knowl. Data Eng.*, 13(1):50–63, 2001.

[Ganti *et al.*, 2002] V. Ganti, J. Gehrke, R. Ramakrishnan, and W.-Y. Loh. A Framework for Measuring Differences in Data Characteristics. *J. Comput. Syst. Sci.*, 64(3):542–578, 2002.

[Garces *et al.*, 2006] D. Garces, A. Krohn, and O. Schoch. Energy Management in Buildings with Sensor Networks. In *EWSN 2006*, 2006.

[Giannella *et al.*, 2003] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu. Mining Frequent Patterns in Data Streams at Multiple Time Granularities. In *Workshop on Next Generation Data Mining*, 2003.

[Han *et al.*, 2000] J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In *SIGMOD 2000, Dallas, USA*, pages 1–12, 2000.

[Herbster and Warmuth, 1998] M. Herbster and M. K. Warmuth. Tracking the Best Regressor. In *Computational Learing Theory*, pages 24–31, 1998.

[IBM, 2005] IBM. Quest Synthetic Data Generation Code by IBM, 2005. available at www.almaden.ibm.com/software/quest/Resources/datasets/syndata.html#assocSynData.

[Kifer *et al.*, 2004] D. Kifer, Sh. Ben-David, and J. Gehrke. Detecting Change in Data Streams. In *VLDB*, pages 180–191, 2004.

[Klein *et al.*, 2007] A. Klein, H.-H. Do, G. Hackenbroich, M. Karnstedt, and W. Lehner. Representing Data Quality for Streaming and Static Data. In *ICDE Workshop on Ambient Intelligence, Media, and Sensing (AIMS07), Istanbul, Turkey*, pages 3–10, 2007.

[Kleinberg, 2003] J. M. Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7(4):373–397, 2003.

[Kolokotsa *et al.*, 2005] D. Kolokotsa, A. Pouliezos, and G. Stavrakakis. Sensor fault detection in building energy management systems. In *5th International Conference on Technology and Automation ICTA'05*, pages 282–287, 2005.

[M.-Ch. Chena and Chang, 2005] A.-L. Chiub M.-Ch. Chena and H.-H. Chang. Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, 28(4):773–781, 2005.

[Manku and Motwani, 2002] G. S. Manku and R. Motwani. Approximate Frequency Counts over Data Streams. In *VLDB 2002, Hong Kong, China*, pages 346–357, 2002.

[Motegi and Piette, 2002] N. Motegi and M. Piette. Web-based Energy Information Systems for Large Commercial Buildings. In *National Conference on Building Commissioning*, 2002.

[Österlind *et al.*, 2007] F. Österlind, E. Pramsten, D. Roberthson, J. Eriksson, N. Finne, and T. Voigt. Integrating Building Automation Systems and Wireless Sensor Networks. Technical report, SICS, 2007.

[Ramesh *et al.*, 2005] G. Ramesh, M. J. Zaki, and W. Maniatty. Distribution-Based Synthetic Database Generation Techniques for Itemset Mining. In *IDEAS*, pages 307–316, 2005.

[Rozsypal and Kubat, 2005] A. Rozsypal and M. Kubat. Association mining in time-varying domains. *Intelligent Data Analysis (IDA)*, 9(3):273–288, 2005.

[Schlimmer and R. H. Granger, 1986] J. C. Schlimmer and Jr. R. H. Granger. Incremental Learning from Noisy Data. *Machine Learning*, 1(3):317–354, 1986.

[Scholz and Klinkenberg, 2007] M. Scholz and R. Klinkenberg. Boosting Classifiers for Drifting Concepts. *Intelligent Data Analysis (IDA), Special Issue on Knowledge Discovery from Data Streams*, 11(1):3–28, 2007.

[Widmer, 1997] G. Widmer. Tracking Context Changes through Meta-Learning. *Machine Learning*, 27(3):259–286, 1997.

[Zhu and Shasha, 2003] Y. Zhu and D. Shasha. Efficient Elastic Burst Detection in Data Streams. In *Proceedings of SIGKDD 2003, Washington, D.C., USA*, pages 336–345, 2003.

Workshop Information Retrieval 2007
# of the Special Interest Group Information Retrieval (FGIR)

24.-26. September 2007, Martin-Luther-Universität Halle

Thomas Mandl, Norbert Fuhr, Andreas Henrich

Web search engines have become a part of daily life for many users. They make information retrieval technology visible for everyone. Often they help users solve their information needs and sometimes they lead to frustration. The ubiquity of search systems has led to the application of information retrieval technology in many new contexts (e.g. mobile and international) and for new object types (products, patents, music). In order to develop appropriate products, basic knowledge on information retrieval needs to be revisited and innovative approaches need to be taken. The quality of information retrieval needs to be evaluated for each context. Large evaluation initiatives respond to these challenges and develop new benchmarks.

The workshop Information Retrieval 2007 provides a forum for scientific discussion and the exchange of ideas. The workshop of the Special Interest Group for Information Retrieval within the Gesellschaft für Informatik (GI) takes place in the week of workshops LWA "Learning, Knowledge and Adaptivity“ (LWA, 24.-26. Sept. 2007 at the Martin Luther University in Halle, Germany). This workshop continues a successful series of conferences and workshops of the Special Interest Group on Information Retrieval (http://www.fg-ir.de). The workshop attracted both practitioners from the industry as well as researchers from universities.

The workshop received 15 submissions from four countries. Eight submissions were accepted as full papers. We would like to thank the members of the program committee for providing constructive reviews

We received submissions on the following topics:

- Retrieval for structured and und multimedia documents
- Text mining and information extraction
- Digital libraries
- Machine learning in information retrieval
- Question Answering
- Search Engine Optimization
- Information Retrieval and the Semantic Web

Program Chairs:

- Prof. Dr. Norbert Fuhr, Universität Duisburg-Essen
- Dr. Sebastian Goeser, IBM Germany Development
- PD Dr. Thomas Mandl, Universität Hildesheim

Program Committee:

- Prof. Dr. Martin Braschler, University of Applied Science and Technology, Zürich
- Prof. Dr. Norbert Fuhr, University of Duisburg-Essen
- Dr. Sebastian Goeser, IBM Germany Development
- Prof. Dr. Andreas Henrich, University of Bamberg
- Prof. Dr. Gerhard Knorz, University of Applied Science and Technology Darmstadt
- Dr. Johannes Leveling, University of Hagen
- PD Dr. Thomas Mandl, University of Hildesheim
- Prof. Dr. Marc Rittberger, DIPF, Frankfurt
- Dr. Ralf Schenkel, Max-Planck-Institute for Computer Science, Saarbrücken
- Dr. Peter Schäuble, Eurospider AG, Zürich, Switzerland
- Dr. Ulrich Thiel, FhG-IPSI, Darmstadt
- Prof. Dr. Christian Wolff, University of Regensburg
- Prof. Dr. Christa Womser-Hacker, University of Hildesheim

A Modified Information Retrieval Approach to Produce Answer Candidates for Question Answering

Johannes Leveling

Intelligent Information and Communication Systems (IICS)
University of Hagen (FernUniversität in Hagen)
58084 Hagen, Germany
johannes.leveling@fernuni-hagen.de

Abstract

This paper describes MIRA, a modified information retrieval approach to produce answer candidates for a question answering system. MIRA is part of IRSAW (Intelligent Information Retrieval on the Basis of a Semantically Annotated Web), a question answering framework that combines information retrieval (IR) with a linguistic analysis of texts to obtain answers to natural language questions. In IRSAW, different techniques for finding answers are employed to produce streams of answer candidates, which are then merged to produce a final answer.

MIRA is capable of producing a stream of answer candidates for definition questions and for a range of factual questions, based on shallow natural language processing and heuristics. MIRA builds on information from a tagged newspaper corpus and on extensive name lexicons. In contrast to other approaches, it does not employ full named entity recognition (NER) or pattern matching, but relies on finding the longest sequence of tags for a given expected answer type (EAT). This paper describes finding the expected answer type for a question, annotating the TüBa-D/Z newspaper corpus, applying MIRA to identify tag sequences representing answer candidates, and its evaluation on the QA@CLEF data set.

The evaluation shows that MIRA is a highly recall-oriented producer of answer candidates, returning the major portion of answer candidates and correct answer candidates in IRSAW. It achieves a large coverage of questions as used in the QA@CLEF test data: For 520 of 600 questions in the test set, answer candidates were produced.

1 Introduction

IRSAW[1](Intelligent Information Retrieval on the Basis of a Semantically Annotated Web) is a framework for building question answering systems. It comprises modules for creating answer candidates (answer streams) based on deep and shallow natural language processing (NLP) methods, answer validation and merging, and natural language generation. IRSAW is primarily based on semantically-oriented NLP. It utilizes the concept-oriented knowledge representation paradigm MultiNet [Helbig, 2006] for representing the meaning of questions and documents.

Figure 1 shows an overview of the core components of the IRSAW architecture. In contrast to other question answering (QA) systems, IRSAW includes a) a full semantic interpretation of questions and documents on which logical inferences are based, b) resolution of linguistic phenomena, including idioms, coreferences, and temporal and spatial aspects in questions and documents (e.g. deictic expressions), and c) generation of answers in natural language (instead of text extraction).

Currently, IRSAW employs answer streams produced by three different methods: InSicht [Hartrumpf, 2005], which is based on logical inferences and matching the semantic network representation of questions and documents; QAP (Question Answering by Pattern matching, [Leveling, 2006]), and MIRA (Modified Information Retrieval Approach). A final, concise answer is produced by a validation module (MAVE), which merges, ranks, and logically validates answer candidates and selects the best candidate [Glöckner *et al.*, 2007].

The main motivation for using several streams of answer candidates originates from characteristics of InSicht. InSicht performs best when applied to texts that are free of syntactical or grammatical errors, because its parser often fails to produce a meaning representation for malformed sentences. InSicht's syntactico-semantic parser (WOCADI, see [Hartrumpf, 2003]) is able to produce a complete semantic network for about 48.7% and a partial semantic network for 20.4% of all sentences in a newspaper corpus [Hartrumpf, 2005]. Common parser errors in WOCADI are caused by the limited robustness of the parser and by missing lexical knowledge. Thus, InSicht produces highly precise answer candidates, but will not find answers when parsing fails. In order to overcome this problem, two modules were added to IRSAW to produce additional answer candidates, QAP and MIRA. To emphasize robustness, these modules employ only shallow natural language processing (NLP) methods or simple heuristics.

Most traditional QA systems are based on shallow NLP only, i.e. they employ no semantic representation of a text or question. Typical methods utilized include adapting information extraction or pattern matching, as, for example in the QA system Webclopedia [Hovy *et al.*, 2001;

[1] The research presented in this paper was funded by the DFG – Deutsche Forschungsgemeinschaft as part of the project *Intelligent Information Retrieval on the Basis of a Semantically Annotated Web* (IRSAW; GZ: LIS 4 – 554975(2) Hagen, BiB 48 HGfu 02-01).

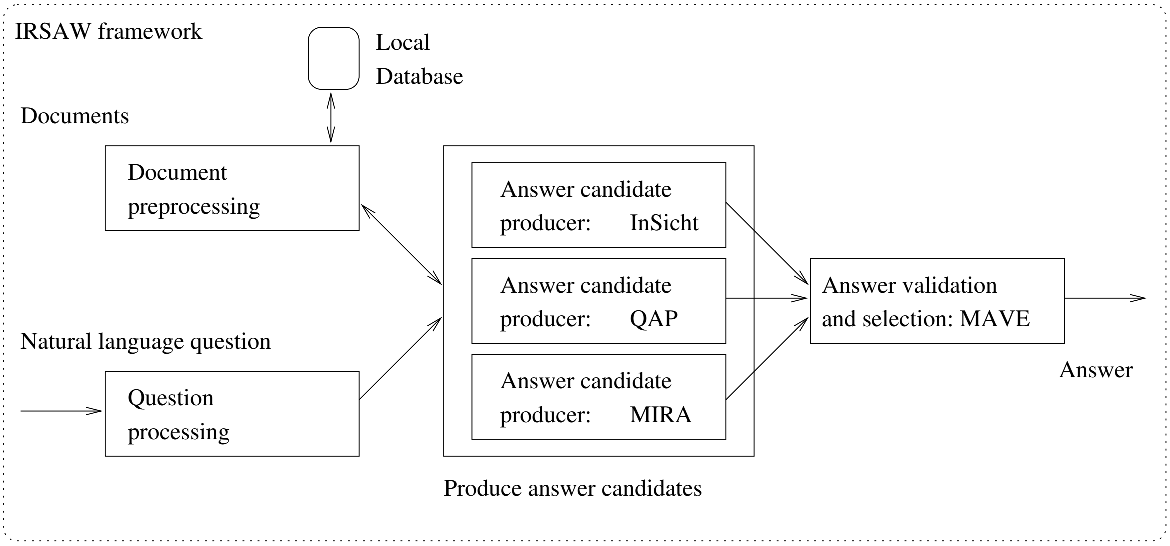

IRSAW framework
Local Database
Documents
Document preprocessing
Answer candidate producer: InSicht
Answer candidate producer: QAP
Answer candidate producer: MIRA
Answer validation and selection: MAVE
Answer
Natural language question
Question processing
Produce answer candidates

Figure 1: Basic architecture of IRSAW and embedding of the MIRA answer candidate producer

Ravichandran and Hovy, 2002]. Other approaches at merging answer streams typically use the same NLP approach with different parameters (see, for example, [Ahn *et al.*, 2005]) to produce answer candidates.

2 Processing Steps in MIRA

MIRA relies on two resources:

1. A manually annotated corpus of German newspaper articles (TüBa-D/Z)[2] allows to a) collect all tokens associated with a subclass of an EAT (e.g. all tokens which are annotated with the EAT *PERSON* and with the subclass first-name can be extracted), and b) generating tag sequences for a given expected answer type (e.g., a sequence of first name and last name is a tag sequence for the EAT *PERSON*). Table 1 shows examples for each EAT.
2. A lookup function returns all possible subclasses (and thus, the implicit class) for a given token. Note that this approach is different from a simple pattern matching approach as employed for the QAP answer stream [Leveling, 2006], in that it does not employ any context and does not rely on hand-coded rules or patterns. The lexica that were collected from the annotated TüBa-D/Z corpus (e.g. lexica containing first names, last names, etc.) were extended with large-scale resources for named entities, including gazetteer information for locations or data from the US census for persons.

Seeing MIRA as part of the question answering system IRSAW, processing includes the following steps:

1. preprocessing documents (text segmentation) and indexing text segments;
2. preprocessing questions and finding the expected answer type;
3. retrieving relevant text segments from a database containing the documents;
4. matching documents and question by finding the longest sequences of tokens in a text segment of the expected answer type; marking up the match as the answer string;
5. returning the top-N answer candidates; and
6. validating answer candidates and selecting the best one as an answer.

In comparison with to the highly precision-oriented InSicht answer stream [Hartrumpf and Leveling, 2006], MIRA is a highly recall-oriented answer producer, relying largely on the answer validation module (MAVE) to select the best answer.

1. Document preprocessing In the context of an open-domain question answering system, the document base typically consists of web pages. For efficiency, these web documents are harvested from the web in advance and indexed in a local database (using the Nutch web crawler[3], which includes the Lucene database system). Web page contents are normalized (e.g. contents are mapped to the same character set and SGML tags are removed) and transformed into text.

The texts are then split into text segments of the same type and of comparable length (in this case: single sentences), by applying a tokenizer and detecting sentence boundaries. Tokenization and sentence boundary detection follow an approach proposed by Grefenstette and Tapanainen [1994], recognizing punctuation inside numeric expressions and punctuation which is part of a token, e.g. in abbreviations. A major difference to this approach is that multi-word proper nouns are merged and represented as a single token (e.g. the proper nouns *"New York"* and *"San Antonio"* count as a single token).

2. Question preprocessing Questions are treated as a single sentence. In contrast to other question answering systems, MIRA employs the WOCADI parser to resolve coreferences including anaphoric references and ellipsis. For example, there may be references to preceding questions or to answers given by a QA system in a real-world dialogue between human and computer. Given

[2] `http://www.sfs.uni-tuebingen.de/en_tuebadz.shtml`

[3] `http://lucene.apache.org/nutch/`

Table 1: Overview over expected answer types with corresponding questions, answer candidates and tag sequences (questions and answers are taken from the QA@CLEF corpus).

EAT	Example question (ID: German/English)	Example answer	Tag sequence
LOCATION	n56R82F7: *"In welchem Land liegt die Stadt Osaka?"* In which country is the city of Osaka located?	in Japan	prep country
PERSON	n72R80F9: *"Wer wurde 1948 erster Ministerpräsident Israels?"* Who became the first Prime minister of Israel in 1948?	David ben Gurion	person-first person-part person-last
ORGANIZATION	n93R31F10: *"Welches deutsche Museum ist weltweit das einzige Museum für interaktive Kunst?"* Which German museum is the only museum for interactive arts world-wide?	ZKM Karlsruhe	organization org-loc
TIME	n29R14F11: *"In welchem Jahr endete offiziell die Besetzung Deutschlands?"* In what year did the occupation of Germany officially end?	im Jahr 1955	prep year num-card
MEASURE	n34R49F4: *"Wie viele Zeilen hat ein Limerick?"* How many lines does a Limerick have?	fünf	num-word
SUBSTANCE	qa03_163: *"Woraus ist die Skulptur 'Chicken Boy' gemacht?"* What is the sculpture 'chicken boy' made of?	Fiberglas	substance
OTHER	qa04_011: *"Wie wird der Ebolavirus übertragen?"* How is the Ebola virus transmitted?	Übertragen werden die Ebolaviren durch direkten Körperkontakt und bei Kontakt mit Körperausscheidungen infizierter Personen per Kontaktinfektion bzw. Schmierinfektion.	– (other entity type)
DEFINITION	qa04_073: *"Was ist ALDI?"* What is ALDI?	Aldi ist der Kurzname der beiden weltweit operierenden deutschen Handelsunternehmen Aldi Nord und Aldi Süd (eigene Schreibweise ALDI SÜD). Der Firmenname Aldi ist eine Abkürzung und steht für Albrecht-Discount.	– (multiple sentences)

```
<DOC>
<DOCID>FR940724-001243</DOCID>
<SENTENCE>Vor 25 Jahren betrat Neil
  Armstrong als erster Mensch den
  Mond, doch heute stagniert die
  bemannte Raumfahrt. </SENTENCE>
</DOC>
```

Figure 2: Example document representation (single sentence).

the question *“Where was Dijkstra born?”*, both the ellipsis *“When?”* and the personal pronoun in *“When did he die?”* are identified by the WOCADI parser. It rewrites these follow-up questions as *“When was Dijkstra born?”* and *“When did Dijkstra die?”*.[4]

A naïve Bayes classifier is employed to find the expected answer type. It uses the first five word forms and their corresponding part-of-speech tag (from STTS, the Stuttgart-Tübingen tag set) as features to determine the expected answer type (EAT) for a question. It was trained on data from the QA©CLEF questions [Magnini *et al.*, 2006], and from manually annotated questions from the SmartWeb question corpus [Neumann and Xu, 2003]. The classifier differentiates between the answer types *LOCATION*, *PERSON*, *ORGANIZATION*, *TIME*, *MEASURE*, *SUBSTANCE*, and *OTHER* (the rest class).

3. Retrieving text segments The natural language question is transformed into an IR query. Stopwords are eliminated and a German stemmer is applied. Query terms are weighted in the order of their importance, starting with proper nouns (which are most important and get the highest weight), nouns[5], numeric expressions, adjectives, and adverbs. The top-N relevant documents are retrieved from the local database for further processing (for evaluation purposes, N was set to 50).

4. Matching documents and questions The matching process in MIRA consists of collecting the longest tagged sequences associated with the EAT. For example, to find answer candidates for the natural language question *“Who was the first man on the moon?”/“Wer war der erste Mensch auf dem Mond?”*, an IR query consisting of the search terms *“first”/“erste”*, *“man”/“Mensch”*, and *“moon”/“Mond”* is processed, with the first term getting the lowest and the last term getting the highest weight. The retrieved text segments include the sentence *“Twenty-five years ago, Neil Armstrong was the first man to step onto the moon, but today manned space flight stagnates”/“Vor 25 Jahren betrat Neil Armstrong als erster Mensch den Mond, doch heute stagniert die bemannte Raumfahrt”*. The corresponding indexed text segment is shown in Figure 2. The expected answer type of the question is *PERSON*, so tokens in a document which constitute parts of person names are seen as (parts of) answer candidates, including the tokens *“Neil”* (tagged with the EAT subclass person-first) and *“Armstrong”* (person-last). The longest sequence of tagged tokens for the answer type *PERSON* is the string *“Neil Armstrong”*, which is returned by MIRA as an answer candidate.

Table 2: Sentence tagged with EAT and subclasses.

Token	EAT tag	Subclass tag
Vor	*TIME*	prep
25	*TIME*	num-card
Jahren	*TIME*	year
betrat	–	
Neil	*PERSON*	person-first
Armstrong	*PERSON*	person-last
als	–	
erster	–	
Mensch	–	
den	–	
Mond	*LOCATION*	other
,	–	
doch	–	
heute	*TIME*	deictic
stagniert	–	
die	–	
bemannte	–	
Raumfahrt	–	
.	–	

Table 2 shows the example sentence tagged with EATs and the corresponding subclasses for tokens. The tag sequences that can be extracted from this annotation include *“prep num-card year”* and *“deictic”* for the *TIME* answer type. Temporal deictic expressions were annotated in the TüBa-D/Z corpus as well (e.g., token sequences like *“vor zwei Jahren”/“two years ago”*, *“gestern”/“yesterday”*) and therefore are included as answer candidates, too. However, relevance assessment would judge these answers as incomplete or partially correct, because references to the document creation time are not given as answer support, yet.

5. Returning answer candidates Answer candidates are returned as a set of quadruples. A candidate includes the answer string, a document identifier, the text segment (sentence) supporting the answer, and the score as determined by the IR system. An obvious improvement to the scoring system would be to include the frequency of the matching tag sequence from the annotated TüBa-D/Z corpus. For instance, a sequence for the class *PERSON* with two leading first names (e.g. *“William Henry Gates”*) is less frequent than a sequence of a first name followed by the last name (e.g. *“Bill Gates”*).

6. Answer merging and validation The last step in question answering with MIRA is handled by the validation module MAVE. It employs inferences and entailments as well as several scores for validation indicators to compute a ranking of answer candidates. MAVE is not explained in greater detail here (see [Glöckner, 2006; Glöckner *et al.*, 2007] for a detailed description).

3 Question Classification and Expected Answer Types

3.1 Recognizing question classes

There is usually some correspondence between named entity classes and expected answer types in QA, which are

[4] Examples have been translated from German into English.

[5] In German, the first word of a sentence, proper nouns, and nouns start with a capital letter. Thus, nouns are relatively easy to spot in a text.

shown in Table 1. Early QA systems concentrated on returning named entities as answers, because they were relatively easy to detect and no parsing was required. Typically, shallow approaches at question answering rely on some form of named entity recognition (NER), possibly including a further classification of proper nouns (NERC). There are a number of software tools with either pre-learned models for NERC or with freely adaptable models. First experiments to adapt these techniques for MIRA showed that pre-learned taggers were not adequate for distinguishing between a number of name classes. Furthermore, a simple adaptation (i.e., replacing the part-of-speech tags for named entities with EAT tags or with subclass tags) did not return accurate results. About 80% accuracy for *LOCATION* and less than 60% for *ORGANIZATION* were achieved in experiments with the Acopost taggers[6] and MBT[7]. The low performance may be due to the corpus size used for training and to the large number of subclasses that are to be recognized. Apart from this problem, most training corpora do not contain many questions. Thus, part-of-speech tagging for questions is often based on sparse data and unreliable.

In contrast to using a tagger for NER, answer entity recognition for MIRA is based on list lookup (different types of named entities comprise only a subset of all answer types). Mikheev et al. [1999] report an accuracy of about 90% for different name classes when not using a NER module.

Table 1 gives an overview over common answer types. Examples were taken from [Neumann and Xu, 2003] and from the QA@CLEF corpus. Note that two classes are treated differently from the rest: questions in the class *OTHER* are not answered at all, as they aim at answers of an expected answer type not covered. For *DEFINITION* questions, answer candidates are generated as follows: for all tokens in the candidate document occurring in the natural language question, both the left and right context of up to five tokens is extracted. Tokens representing punctuation and stopwords at the end are eliminated and auxiliary verbs at the beginning and coordinatives as well. In many cases, this simple approach returns adjective noun phrases which make up a good answer candidate for definition questions.

3.2 Annotating the TüBa-D/Z corpus

Experiments are based on a manual annotation of the TüBa-D/Z corpus, consisting of 27,067 sentences[8] (500,628 tokens) from the German newspaper *taz*. There exists the treebank data for TüBa-D/Z (a syntactic annotation), which was not exploited for MIRA.

The annotation of tokens includes assigning EAT tags and subclasses. The annotation started on the level of the EAT tags. In the next step, subclasses were added, e.g. for the class *PERSON* the distinction between first name, last name, title, and other parts of a name for person names. Table 3 shows the distribution of the classes corresponding to the EAT. The TüBa-D/Z corpus contains 8,274 *LOCATION* tokens. Table 4 shows the distribution of subclasses for the *LOCATION* class.

The TüBa-D/Z corpus annotation was checked with several techniques. First, the variation detection tool DECCA[9] was applied to spot inconsistent annotations. This variation analysis was performed on the level of part-of-speech tags, EAT class tags, and subclasses. Second, the corpus itself was refined using the corrections supplied by the corpus publishers. Furthermore, results from a frequency analysis were employed to find and correct common spelling errors.

Table 3: Tag frequency (EAT) in the annotated TüBa-D/Z corpus.

Name class	Corpus frequency
LOCATION	8,394
PERSON	14,527
ORGANIZATION	7,148
TIME	14,524
MEASURE	895
SUBSTANCE	293
MISC/OTHER	2,987

Table 4: Frequency of *LOCATION tokens* (8,274 tokens) annotated with the corresponding EAT subclass.

LOCATION	Subclass frequency
city	3,717
state	370
country	1,955
region	926
river	85
sea	17
island	55
mountain	11
street	613
streetno	124
building	195
other	206

4 Evaluation Results

The QA@CLEF test corpus contains about 277,000 newspaper articles and newswires from the *Frankfurter Rundschau*, *Der Spiegel*, and *SDA (Schweizerische Depeschenagentur, Swiss news agency)* from the years 1994 and 1995.

The evaluation of MIRA as an answer candidate producer in the IRSAW system is based on the QA@CLEF questions and corpus from 2004 to 2006. Answer candidates for the 600 questions of this data set obtained from the different answer streams were annotated manually for their correctness. The heterogeneity of the answer streams InSicht, QAP, and MIRA is reflected by the numbers on coverage of questions and precision shown in Table 5.

The MIRA stream was meant to produce a high recall, which is achieved by the use of a simple IR search and subsequent matching of EAT and subclasses. Table 6 shows results for the QA@CLEF 2006 data. Answer candidates were produced by InSicht, QAP, and MIRA. Results are based on 600 questions from the QA@CLEF collection for which answer candidates were validated by the MAVE module. For comparison, an upper baseline is given as well (i.e. always selecting a correct answer if such a candidate was found).

The best experiment for monolingual German question answering at QA@CLEF 2006 (200 of the 600 questions) was achieved by the DFKI group. Their experiment has

[6] `http://acopost.sourceforge.net/`

[7] `http://ilk.uvt.nl/mbt/`

[8] In some cases, sentences were annotated to be ignored for annotation (58 sentences out of the total 27,125).

[9] `http://decca.osu.edu/`

Table 5: Coverage and precision for the answer streams InSicht, QAP, and MIRA, based on 600 questions from QA@CLEF data from 2004 to 2006.

System	# Candidates	Coverage	# Correct answers	Precision
InSicht	1,212	226/600	625	51.6%
QAP	2,562	114/600	1,190	46.6%
MIRA	14,946	520/600	1,738	11.6%

Table 6: Results of creating answer candidates and answer selection for the IRSAW answer streams for 600 questions from QA@CLEF data from 2004 to 2006.

Run	# Correct answers	# Incorrect answers	# Wrong answers
InSicht+Mira+QAP	247.4	15.8	307.8
InSicht+Mira+QAP (optimal)	305.0	17.0	249.0

80 correct answers, six inexact answers, and eight unsupported answers (dfki061dede, see [Magnini *et al.*, 2006]).

Table 7 shows results for 200 questions from QA@CLEF 2006 and for from QA@CLEF 2003 to 2006 for the MIRA answer producer alone, omitting answer validation and selection. (Note that there may be more than one correct answer candidate per question.) The 800 questions contains 66 NIL questions (questions with no answer in the document corpus), of which eight questions would have been answered correctly, i.e. MIRA did not find a corresponding answer candidate. In total, MIRA produces answer candidates for 709 of the 800 questions.

Summarizing, MIRA works as expected. It produces a highly recall-oriented answer stream, covers more questions than the other answer producers in IRSAW, and returns the largest number of correct answer candidates.

The monolingual German QA@CLEF tasks are becoming more complex every year. For example, in 2006 list questions and questions with temporal restrictions were introduced. In 2007, the test set contained questions with anaphoric references to previous questions or answers. MIRA employs shallow NLP methods (i.e., very limited syntactic information and no semantic knowledge is required). However, MIRA may be able to produce answer candidates for these kinds of complex questions.

MIRA tag sequences are in some cases even too fine-grained or contain linguistic phenomena that are not targeted by the questions in the QA@CLEF data. For example, some *TIME* tag sequences allow temporal expressions for names for weekdays, month names, or the time of day. Although there are questions for which this kind of temporal expression would deliver an answer, this is never the case for the QA@CLEF data: Most *TIME* questions are concerned with dates of birth or death or dates for an event only. If the document creation time is given as additional answer support, for answers containing a temporal deictic expression, MIRA will support this kind of question, too.

There is one special type of question that is difficult to identify and difficult to answer: list questions. In MIRA, they are difficult to identify because there are too few examples in the training set. Therefore, the questions were annotated with the corresponding simple class, e.g. *"Name the three Baltic States"* was annotated as a *LOCATION* question. In some cases, a distinction of this kind is not even clear from the question form (*"What is the profession of your parents?"* → *"teacher"* or *"teacher and carpenter"*, *"Who won the FIFA World Cup in 1954, 1974, and 1990?"* → *"Germany"* or (possibly) *"Germany, Brazil, and Argentina"*).

A simple, but effective approach to process list questions in MIRA without introducing a special list question class for every EAT would be to automatically generate meta-sequences representing a coordination of simple EAT tag sequences (e.g. *PERSON* "," *PERSON* "," "and" *PERSON*; *PERSON* "and" *PERSON* ; where *PERSON* denotes a tag sequence for one item of a *PERSON* list question).

Answer validation is not discussed in detail in this paper, but obviously, it would profit from using knowledge implicit in the tag sequence. For instance, questions of the expected answer type *MEASURE* typically consist of a sequence of cardinal or ordinal numbers, followed by a unit and should be classified into a finer taxonomy. Thus, answer candidates can be filtered if the answer string contains an incompatible unit of measurement. This taxonomy might be derived directly from the EAT subclasses.

5 Conclusion and Outlook

This paper described the adaptation of an IR approach to create MIRA, a high-recall approach to produce answer candidates for a question answering system. In contrast to other shallow question answering approaches, MIRA relies on tag sequences from a manually annotated corpus. One limitation of MIRA is the coarse-grained classification of expected answer types, which allows for a higher recall, but also limits precision.

The MIRA answer candidate producer achieves a high coverage, i.e. it produces answer candidates for 709 out of 800 questions in the test set. MIRA is divided from the answer validation, i.e. it creates many answer candidates to choose from, but does not perform even simple tests for validity of answer candidates. An evaluation of MIRA (combined with the MAVE answer validation) would yield a theoretical ideal performance of 26% of correct answer candidates. This value lies in the range of shallow QA systems. However, MIRA is just one of several producers of answer candidates in IRSAW and in the combination with the QAP and InSicht answer candidate producers, a much higher performance can be achieved.

Future work will therefore include using a finer distinction of expected answer types and processing list questions by allowing regular patterns of tag sequences.

Table 7: Results for the MIRA answer stream for QA@CLEF questions.

	top-N					
	N=50	N=40	N=30	N=20	N=10	N=5
# Correct answer candidates (2006)	798	731	615	441	215	95
# Inexact answer candidates (2006)	56	54	53	39	20	12
# Wrong answer candidates (2006)	4,436	4,013	3,421	2,539	1,360	722
# Correct answer candidates (2003–2006)	1,864	1,744	1,503	1,137	609	263
# Inexact answer candidates (2003–2006)	287	275	248	191	103	54
# Wrong answer candidates (2003–2006)	17,326	16,095	14,102	10,620	5,694	3,013

References

[Ahn *et al.*, 2005] David Ahn, Valentin Jijkoun, Karin Müller, Maarten de Rijke, Stefan Schlobach, and Gilad Mishne. Making stone soup: Evaluating a recall-oriented multi-stream question answering system for Dutch. In Carol Peters, Paul Clough, Julio Gonzalo, Gareth J. F. Jones, Michael Kluck, and Bernardo Magnini, editors, *Multilingual Information Access for Text, Speech and Images: 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004*, volume 3491 of *Lecture Notes in Computer Science*, pages 423–434. Springer, Berlin, 2005.

[Glöckner *et al.*, 2007] Ingo Glöckner, Sven Hartrumpf, and Johannes Leveling. Logical validation, answer merging and witness selection – a case study in multi-stream question answering. In *Proceedings of RIAO 2007 (Recherche d'Information Assistée par Ordinateur – Computer assisted information retrieval), Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, Pittsburgh, USA, 2007. Le Centre de Hautes Etudes Internationales d'informatique Documentaire – C.I.D.

[Glöckner, 2006] Ingo Glöckner. University of Hagen at QA@CLEF 2006: Answer validation exercise. In Alessandro Nardi, Carol Peters, and José Luis Vicedo, editors, *Results of the CLEF 2006 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2006 Workshop*, Alicante, Spain, 2006.

[Grefenstette and Tapanainen, 1994] Gregory Grefenstette and Pasi Tapanainen. What is a word, what is a sentence? Problems of tokenization. In *3rd International Conference on Computational Lexicography*, pages 79–87, Budapest, Hungary, 1994.

[Hartrumpf and Leveling, 2006] Sven Hartrumpf and Johannes Leveling. University of Hagen at QA@CLEF 2006: Interpretation and normalization of temporal expressions. In Alessandro Nardi, Carol Peters, and José Luis Vicedo, editors, *Results of the CLEF 2006 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2006 Workshop*, Alicante, Spain, 2006.

[Hartrumpf, 2003] Sven Hartrumpf. *Hybrid Disambiguation in Natural Language Analysis*. Der Andere Verlag, Osnabrück, Germany, 2003.

[Hartrumpf, 2005] Sven Hartrumpf. Question answering using sentence parsing and semantic network matching. In Carol Peters, Paul Clough, Julio Gonzalo, Gareth J. F. Jones, Michael Kluck, and Bernardo Magnini, editors, *Multilingual Information Access for Text, Speech and Images: 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004*, volume 3491 of *Lecture Notes in Computer Science*, pages 512–521. Springer, Berlin, 2005.

[Helbig, 2006] Hermann Helbig. *Knowledge Representation and the Semantics of Natural Language*. Springer, Berlin, 2006.

[Hovy *et al.*, 2001] Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question answering in webclopedia. In *Proceedings of the TREC-9 Conference, NIST*, Information Sciences Institute University of Southern California, 2001.

[Leveling, 2006] Johannes Leveling. On the role of information retrieval in the question answering system IRSAW. In *Proceedings of the LWA 2006 (Learning, Knowledge, and Adaptability), Workshop Information Retrieval*, pages 119–125. Universität Hildesheim, Hildesheim, Germany, 2006.

[Magnini *et al.*, 2006] Bernardo Magnini, Danilo Giampiccolo, Pamela Forner, Christelle Ayache, Valentin Jijkoun, Petya Osenova, Anselmo Peñas, Paulo Rocha, Bogdan Sacaleanu, and Richard Sutcliffe. Overview of the CLEF 2006 multilingual question answering track. In Alessandro Nardi, Carol Peters, and José Luis Vicedo, editors, *Results of the CLEF 2006 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2006 Workshop*, Alicante, Spain, 2006.

[Mikheev *et al.*, 1999] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL 1999)*, pages 1–8, Bergen, Norway, 1999.

[Neumann and Xu, 2003] Günter Neumann and Feiyu Xu. Mining answers in German web pages. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence (WI-2003)*, pages 125–131, Halifax, Canada, 2003.

[Ravichandran and Hovy, 2002] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 41–47, Philadelphia, USA, 2002.

Discovering concepts via semantic expansion of keyword queries

Markus Lorch, Andrea Elias, Thomas Hampp-Bahnmüller, Benjamin Sznajder
IBM
lorchm@acm.org, baader@de.ibm.com, thampp@de.ibm.com, benjams@il.ibm.com

Abstract

The problem of finding relevant information with high precision and recall in the masses of unstructured data is a grand challenge to information retrieval software. Semantic search mechanisms pose promising approaches to meet with this challenge. Semantic search can disambiguate results by interpreting concepts by use of contextual information and hence, improve precision. It can also locate instances of concepts that are not easily enumerable and for this reason may not be findable through basic synonym mechanisms or the like. Finding concept instances can, hence, also improve recall.

Traditional semantic search uses powerful query languages that are too complex to be useful for the regular information retrieval system user. This paper describes a mechanism to transform standard keyword queries into semantic queries through semantic expansion. It contrasts this approach to related work and describes an implementation of this mechanism.

1 Introduction

Keyword search is widely used and accepted. It is easy to use and does not require deep technical skill. Naïve users can formulate simple and short queries to search for information of interest. However, keyword search has its limitations:

Low precision makes it difficult for the user to locate applicable results. Simple keyword queries are frequently very ambiguous and do not allow to differentiate between the meanings of words. For example when processing a query for “rock” the search engine will not know if the user looks for rock (music) or rock (stone), hence returning results for both concepts.

Low recall may also limit basic keyword query effectiveness as alternative representations (synonyms for example) of the same expressions may be ignored.

Automatic synonym expansion can be used to alleviate some of the low recall problems by automatically expanding the query with alternative expressions [Ekmekcioglu, 1992]. However, synonym expansion has its limits. Precision typically declines due the increased ambiguity of the expanded query [Bernstein, 2002]. This is especially true for large synonym lists which can also have a significant performance impact on query processing. Furthermore, synonym expansion is inapplicable for concepts which cannot be enumerated, for example the finite but huge number of symbol combinations that represent a phone number cannot be searched for through this mechanism.

Semantic search can overcome the limitations of keyword search as context information and background knowledge are leveraged both, at query and indexing time to improve the search results. Unfortunately query languages with semantic search capabilities, like for example SPARQL [Prud’hommeaux & Seaborne, 2005], are frequently complex and therefore unsuitable for an unskilled user. In addition, experience with query logs from keyword search has shown that users typically only specify a few keywords and only very seldom use operators to refine their query [Holsher & Strube, 2000]. People expect the information retrieval system to *“understand what they mean and not what they say”* and they are not willing or unable to specify complicated queries to express their intent.

Interpreting the user's intent is a difficult problem and many natural language processing (NLP) approaches have been proposed and are being developed to address this issue (see section five). NLP remains a very difficult problem and NLP systems that can partially interpret one particular human language typically require extensive adaptation when a new human language has to be supported by the system.

This paper describes an approach to understand the user intent by interpreting selected query keywords as synonyms for semantic concepts. In contrast with the classical synonym query expansion, the query terms are not expanded to other natural language terms but to a representation of their analogous semantic concept. The keyword query is mapped to a more complex semantic search expression suitable for the underlying retrieval system. In case of the keywords “phone number” this translates to a search for the occurrence of an actual phone number instead of, or in addition to the keywords “phone” and “number”. The approach is simple but effective and is easily adapted to new languages and environments. The approach is also search engine and query language independent.

The remainder of this paper is organized as follows. Section two describes our approach to expand simple keyword queries to also include instances of semantic concepts in the query. Section three provides an overview of the XML Fragments query language and how it can be used to specify semantic queries. Section four presents an implementation of our approach in the context of the IBM Enterprise Search product OmniFind[1]. Section five puts

[1] http://www.ibm.com/software/data/enterprise-search/omnifind-enterprise/

this work in context by identifying related work and contrasting alternative ways to aid the user in the creation of semantic queries. Section six concludes the paper with a summary and an outlook to future work.

2 Automatically locating concept instances

The approach we present in this section enables end users to leverage the power of semantic queries through a standard keyword search interface. A user's keyword query is automatically expanded to include relevant concepts. These concepts must be expressed in a language understood by the information retrieval back-end (e. g., the search engine) and information about instances of relevant concepts must be available to the search engine.

The overall mechanism consists of two analysis steps. The focus of this discussion will be on step two occurring at query time. But the query time processing requires index time analysis during document processing in step one.

In step one the unstructured information that is to be entered in the information retrieval system is analyzed to detect instances of concepts of relevance. This analysis step can e.g., discover entities and their names (e.g., persons, places, and organizations), relationships between entities (e.g., which person is CEO of what organization), structured information like telephone numbers, and the meaning of expressions (e.g., if positive or negative sentiment is expressed in a sentence). This kind of analysis could use statistics or rule/knowledge-based methods as discussed in the literature under topics like semantic role labeling, information extraction or named entity detection. For the purpose of this discussion it is irrelevant how the additional semantic meta-data information is computed. We assume that appropriate methods for semantic analysis and disambiguation in the context are available and can be integrated with the document processing and indexing of the search engine. Information about the instantiations of these concepts is stored as meta-data and made available to the information retrieval engine.

The unstructured textual content, as well as the discovered and stored meta information can be queried for through advanced query languages like XML Fragments (an overview is given in section 3 of this paper). This query language serves a task in the advanced information retrieval world that is comparable to the task of SQL in the world of relational databases. Powerful queries that identify relevant content by specifying a combination of keywords, concepts, and their relationship are possible. Formulating such queries is a solvable task for an information retrieval specialist but typically is cumbersome and error-prone. Non-specialists are often unable to leverage the power of such query mechanisms to their full extent.

Step two, which is the focus of this method, aims at enabling the regular user to leverage the power provided by the text analysis from the first step without the need to formulate a complex query and is performed at query time. The simple keywords query provided by the user is parsed and analyzed for keywords that identify one or more of the concepts understood by the system. If a keyword identifying one of the understood concepts is found, then the query provided by the user is transformed into a semantic query. This transformation results in an expansion of the query to not only search for the keywords identifying concepts (and other keywords) but also for instances of the corresponding concepts. Revisiting the query given in the introduction -'phone number'- the expanded query will result in a disjunction of the keyword terms 'phone' and 'number' with a representation of the semantic concept of a phone number. Results for the query will, hence, be composed of documents containing the terms 'phone' and 'number' as well as documents containing instances of the phone number concept, e.g., "+1 234 5678".

This mechanism is generally search engine agnostic. It only requires a search engine that allows incoming information to be analyzed and augmented with meta-data which in turn is attached to the tokens or spans of tokens of the original document (e. g., "position 5 to 15 is a telephone number"). The search engine must be able to index such named spans for later efficient retrieval.

Furthermore the search engine must provide the necessary query language expressiveness to also search for the meta-data generated by the text analysis steps and ideally be able to highlight the original text (the text covered by the named span) that led to a match on the meta-data in the result set. The semantic query expansion logic can either be implemented directly in the search engine's query parser or can be located in the search application as a preprocessing step. By implementing it in the search application the mechanism can be added to any search engine system that fulfills the requirement of indexing named spans.

Many search engines support field based indexing and retrieval. Field search can be an acceptable retrieval technology for the implementation of semantic synonym expansion. E.g., all instances of phone numbers could be indexed as belonging to a field "phone_nbr" and at search time an underlying query for documents with a non-empty "phone_nbr" field could be generated. In this paper, we discuss a more powerful retrieval technology for semantic synonym expansion that can deal with phenomena like adjacency, highlighting, nesting of concepts (e.g., a phone number as part of an address) and relations, and more phenomena which are all hard or impossible to implement based on field search.

3 XML Fragments for Semantic Queries

The XML Fragments query language is used to express semantic queries in the implementation discussed in section four. In this section the language and its benefits are briefly explained. An exhaustive definition can be found in [Mass *et al.* , 2007] and [Broder *et al.*, 2004].

Following the QBE (Query By Example) paradigm [Zloof, 1977], Broder *et al.* [Broder *et al.*, 2004] defined a language where pieces of XML are used for querying XML collections. XML Fragments are thus portions of valid XML, possibly combined with free text. In addition to this basic definition, some very classical IR operators like '+', '-' or quoting the phrases were added [Broder *et al.*, 2004]**.**

'`<music_genre> Rock </music_genre>`' query will retrieve, for example, documents containing a term 'Rock' appearing under a tag `<music_genre>`, when '`<mineral> Rock </mineral>`' will retrieve the same term under a tag `<mineral>`.

In [Mass *et al.*, 2007], the language has been augmented, among others, with the Boolean operators `<.and>`, `<.or>` and the ability to define *Target Elements*: pre-pending a tag name in a query with the # symbol will retrieve the subtree rooted by this twig instead of the whole document.

The described approach can be implemented independent of a specific query language as long as the query language provides a means to query for concepts in addition to keyword terms. XML Fragments appeared as a powerful language for implementing our approach that provided the required semantic search capability.

XML Fragments easily gives the ability to combine free text and semantic constraints:

'`<.or> "phone number" <phone/> </.or>`', for example, will retrieve any documents containing the phrase "phone number" or a subtree rooted by a tag labeled with "phone". In our approach terminology, we will say that the query will retrieve the terms 'phone' and 'number' as well as the text labeled with the semantic concept of an actual phone number marked up by the tag '<phone>'. In comparison, expressing the same query in XPath [Berglund, *et. al,* 2003] is significantly more complex. XPath lacks a simple mechanism to connect individual query terms through an OR operator.

Furthermore, in XPath, the target element is implicitly the last node of the query path. XML Fragments, allows to define any query tag and a multiplicity of them as the target elements. The ability to explicitly define the target element appeared very useful to specify highlighting candidates in the search results.

4 Easy Semantic Search in IBM OmniFind

A version of the semantic expansion mechanism together with a customizable text analytics module are implemented in IBM OmniFind 8.4 Enterprise Edition. In this implementation, three distinct components work together to achieve what is called "Easy Semantic Search".

First, a customizable text analytics engine is added to OmniFind's Unstructured Information Management Architecture (UIMA) [Ferrucci & Lally, 2004] processing pipeline to detect instances of concepts in the document text. Examples include enumerable entities like product names or structured items like phone numbers. The detection rules can be extended by the enterprise search administrator. The discovered meta information is stored as named spans, identifying the concept instances, in the index. This corresponds to step one described in section two. The open architecture of UIMA allows to plug-in any compliant text analysis module for the semantic analysis task and in fact it has been used with modules implementing a variety of text analysis mechanisms from simple regular expressions over knowledge based named entity detectors to machine-learning based modules (e.g. OpenNLP taggers). The semantic expansion of the query is not concerned with how the meta information about concept instances in the documents got computed. The query side step is independent of how the index side task is actually accomplished. It is a necessary part for the overall system but not the focus here.

Second, OmniFind's synonym dictionary mechanism is used to define the semantic synonyms for keywords. The semantic synonyms take the form of XML Fragments query expressions and can be used together with ordinary synonyms. For example, the keywords "phone number" have the synonym entry "telephone number" as well as the XML Fragments expression '`<#phonenumber/>`'.

Third, a component located in the search application builds the complex semantic query. The component first retrieves synonyms from the search engine back-end for the query terms. If the synonym list contains one or more semantic concepts, then the query is replaced by an XML Fragments query that contains the original keyword terms as well as semantic terms for applicable synonym concepts. We should stress that the expansion logic does not replace the original keyword but rather creates an OR expression to locate either the keyword or a concept instance relating to the keyword. E.g., supposing the synonym entries of the paragraph upper, a keyword query for 'IBM phone number' becomes the complex XML Fragments:

'`ibm <.or> phone "telephone number" <#phonenumber/> </.or> <.or> number "telephone number"<#phonenumber/> </.or>`'

As the original keyword consists of two terms for which a semantic synonym exists the expansion logic follows Boolean algebra rules to create two OR parts connected by AND. `(A ^ B) v C` becomes `(A v C) ^ (B v C)` which can easily be represented in an XML Fragments expression.

5 Alternative Approaches & Related Work

In a system where the search engine allows to discover entities and store them in the index, semantic expansion is a novel approach to allow the user to easily search for the semantic information. However, it is not the only available option to accomplish this. Other mechanisms can be used instead of or in combination with semantic synonym expansion.

This section will give an overview of some of the more common alternatives. It will link the approaches to related work and compare them with semantic synonym expansion.

Querying with XML over annotated collections is an active research area. Most of the works like [Fuhr & K. GrossJohann, 2001, Xquery, 2006] for example, focus on defining semantic query languages that enable the user to express constraints on the context in addition to the content. Exposing a user to a powerful semantic query language gives the user full control and allows him to use the query language to its full extend. Unfortunately, as pointed out earlier, the user then needs to master the complex query syntax. Typing complex queries in a strict syntax is error-prone. In addition to understanding the query syntax the user must also know the XML schema of the annotated data to be able to formulate actual queries. Given these restrictions this approach is limited to the skilled administrator or power user. Its role can be seen as similar to the role of SQL in the structured world which is rarely typed by its end users.

One of the prominent front ends to SQL in structured applications are form-based interfaces where the user fills out a form on the screen and the application converts that into a complex query when the form is submitted. This approach can also be used in the world of unstructured text search as an alternative to semantic query expansion, when a form can be used to formulate a query request.

While forms are familiar and easy to use for most end users, their GUI elements (labels, fields, drop down lists etc.) have to match the concepts available for search (i.e., the type system or schema). Any change in the schema requires a corresponding form change in the GUI. Doing these changes manually is time consuming and costly, doing them automatically requires a complex underlying system. In contrast, our semantic synonym approach does not require any GUI support and therefore no adjustments to changes at all.

Complex natural language query understanding and preprocessing is another type of approach to address the complex query problem. This requires linguistic analysis of the user query that goes beyond the simple substitution approach used by semantic synonym expansion. It can range from "heavy weight" analysis using full NLP logic for question answering that translates full questions like "Who is CEO of IBM?" into queries like '<company> IBM <#CEO/> </company>' to "middle weight" analysis that can translate a query like "Barbara phone" into queries like:

<person_with_phone><person> Barbara</person> </person_with_phone>.

An example of this "heavy weight" NLP analysis approach is the PIQUANT [Prager *et al.*, 2007] system in which not only the documents but also the query is processed by a set of advanced text analysis engines that add semantic meta information. In addition to deeper query preprocessing, PIQUANT also has a post-processing step which tries to extract an actual answer from the documents that match the XML query generated from the natural language user question.

"Middle weight" analysis may not need full NLP query parsing. It may use information about synonyms for concepts (e.g., phone for <phonenumber/>) and in that respect it is similar to semantic synonym expansion. But that approach can only account for the translation of concept words like "phone" to their corresponding concept queries. To be able to do instance-based translations, a "middle weight" approach needs to have access to information about which token instances correspond to which concepts. For example, the query token "Barbara" can be an instance of both the concept <person/> and the concept <place/> in a corpus where both <person>Barbara Miller</person> and <place>Santa Barbara</place> occur. Knowing the conceptual context in which a token occurs helps disambiguating the query to the sole ambiguous terms present in the queried corpus. Acquiring this information is more difficult and goes beyond the semantic synonym expansion approach discussed in this paper.

An example of this "middle weight" query analysis approach is AVATAR Semantic Search [Kandogan *et. al.*, 2006, Jayram *et. al.*, 2006]. The AVATAR query preprocessing system expands a keyword query at query time by automatically suggesting query *interpretations*. Interpretations are generated by analyzing the paths where the keyword appears as a data value in the data set. Since multiple interpretations can be generated by this process, a pruning step drops some unrealistic interpretations and offers the top interpretations to the user. Each query interpretation can be a XML Fragments query.

All these deeper analysis approaches can deal with more complex cases than simple semantic synonym expansion. They can give much better search results if and when they are available in a search application. But they require significant extensions to the search engine and the query parsing that cannot be easily implemented on top of existing search engines. In that sense they are still experimental, complex to create, and they typically need non-trivial effort to be adjusted to a new schema, domain or language. Semantic synonym expansion can only do a subset of what can be done with more elaborate analytics, but it is easy to implement, easy to adjust to a new language or domain, and easy to maintain over the live time of an application. All maintenance is carried out by modifying or exchanging the semantic synonym dictionaries which link keyword terms of the desired language to semantic concepts marked-up in the indexed documents..

6 Summary and future work

The semantic expansion of keyword queries combines powerful query languages like XML Fragments with easy to use keyword search. Semantic search capabilities of information retrieval engines are made accessible to the end user by transforming keyword queries in complex semantic search expressions. The approach is simple and practical and offers significant advantages over ordinary synonym expansion and fielded search.

The method is largely independent of the underlying search engine or query format used and does not require expensive processing at query time. It has been implemented in an enterprise search product and first user experiences show positive results. Recall is especially improved by the ability to transparently search for semantic concept instances through a standard keyword query interface, as pointed by first customer feedbacks. Supplementary experiments should be done including use of known benchmarks like TREC.

The performance impact of running a semantically expanded query, while significant when compared to a simple keyword query, is similar to the execution of other OR-connected multi-term queries in the implementation and is well within typical complex query processing times. The current implementation performs the query expansion on the client side requiring an additional client-server communication step to retrieve the semantic synonyms, this adds some fixed communication overhead to the query execution time and doubles the number of client requests the server has to handle. Other implementations may choose to co-locate the expansion code with the query parser avoid the additional communication at the cost of having to modify the search server side of the system. (In the discussed implementation all semantic synonym expansion code is located in the search application, requiring no modification to the enterprise search server.) It is important to consider the performance impact of a larger number of complex (semantic) queries in systems where a larger number of semantic synonyms are configured and hence semantically expanded queries are expected to be numerous.

Future work may include the definition of a list of query keywords that correspond to the detected concept in the configuration of the text analysis engines. This would allow the system to automatically extend keyword queries to semantic queries without the need for the search administrator to define a binding between keywords and detected concepts through synonyms. The actual synonyms for a concept could also be delivered by an online lexical database like WordNet[2] or openCYC[3].

[2] http://wordnet.princeton.edu/

[3] http://www.opencyc.org

Future versions should also go beyond the current limitation of synonyms to include other semantic relations like hyponymy and meronymy.

Of course an existing domain ontology can be an extremely helpful source of this kind of information. One point of this system though is to provide something that can have much better search quality than classic keyword search without necessarily requiring the complexity of a full fledged ontology.

A promising approach is the combination of semantic synonyms with more "heavy weight" NLP steps like those discussed in section five. The integration would allow for the analysis steps to be optional modules in an overall query analysis framework. It would create a system which can start with synonym expansion but add the deeper processing for even better results when resources (e. g., instance information or full NLP parsers) become available for a given domain and language.

References

[Berglund, *et al,* 2003] A. Berglund, *et. al* "XML Path Language (XPath) 2.0 . *W3C Working Draft*, 12 Nov 2003.

[Bernstein, 2002] A. Bernstein and M. Klein, Towards high-precision service retrieval. In Semantic Web---ISWC 2002, Lecture Notes in Computer Science, Vol. 2342, pp. 84-101, 2002

[Broder *et al.*, 2004] A. Broder, Y.S. Maarek, M. Mandelbrod & Y.Mass "Using XML to Query XML – From Theory to Practice". *In Proceeding of RIAO,* Avignon, France*,* 2004.

[Ekmekcioglu, 1992] F.C. Ekmekcioglu. Effectiveness of query expansion in ranked-output document retrieval systems. Journal of. Information Science, Vol. 18, pp. 139-147, 1992.

[Ferrucci & Lally, 2004] D. Ferrucci and A. Lally. UIMA: An architectural approach to unstructured information processingin the corporate research environment. *J. of Natural Language Engineering*, 10(3-4):327–348, 2004.

[Fuhr & K. Grossjohann, 2001] N. Fuhr & K. Grossjohann "XIRQL: A Query Language for Information Retrieval in XML Documents", in *"Proceedings of SIGIR'2001,* New Orleans, LA, 2001.

[Jayram *et al.*, 2006] T.S Jayram *et. al*, "Avatar Information Extraction System", in "*Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*", 2006.

[Kandogan *et al.,* 2006] E. Kandogan *et. al*, **"**Avatar semantic search: a database approach to information retrieval", in *"Proc, 2006 ACM SIGMOD international conference on Management of data"*, Chicago, USA, 2006.

[[Mass *et al.*, 2007] Y. Mass, D. Sheinwald. B. Sznajder and S. Yogev, "XML Fragments extended with database operators", in *Proceeding of RIAO*, Pittsburgh, USA, 2007.

[Prager *et al.*, 2007] J. Prager, J. Chu-Carroll, E. Brown, and K. Czuba, "Question answering using predictive annotation", In Adv. in Open-Domain Question Answering, T. Strzalkowski & S. Harabagiu, editors, Kluwer Academic Publishers.

[Prud'hommeaux & Seaborne, 2005] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/, Feb 2005 .

[Xquery, 2006] XQuery-FT – XQuery 1.0 and XPath 2.0 Full-Text, W3C Working Draft 1 May 2006 http://www.w3.org/TR/xquery-full-text/

[Zloof, 1977] Zloof M. (1977). Query by example. *IBM Systems Journal*, 16(4):324-343, 1977) .

The Effectiveness of Concept Based Search for Video Retrieval

Claudia Hauff **Robin Aly** **Djoerd Hiemstra**
Computer Science
University Twente
P.O. Box 217
7500 AE Enschede
The Netherlands
e-mail: {c.hauff, r.aly, d.hiemstra}@ewi.utwente.nl

Abstract

In this paper we investigate how a small number of high-level concepts derived for video shots, such as *Sports*, *Face*, *Indoor*, etc., can be used effectively for ad hoc search in video material. We will answer the following questions: 1) Can we automatically construct concept queries from ordinary text queries? 2) What is the best way to combine evidence from single concept detectors into final search results? We evaluated algorithms for automatic concept query formulation using WordNet based concept extraction, and we evaluated algorithms for fast, on-line combination of concepts. Experimental results on data from the TREC Video 2005 workshop and 25 test users show the following. 1) Automatic query formulation through WordNet based concept extraction can achieve comparable results to user created query concepts and 2) Combination methods that take neighboring shots into account outperform more simple combination methods.

1 Introduction

Bridging the semantic gap is a key problem in multimedia information retrieval [Sebe, 2003]. This gap exists between the well understood extraction methods of low level features from media files (e.g. color histograms or audio signals) and the high level concepts users express their information needs with (e.g. *Find me pictures of a sunrise*). This problem applies especially to video retrieval where the detection of the semantic concepts has become a research focus in recent years [Naphade and Smith, 2004]. This paper investigates the problem of identifying the relevant concepts given the user's text query, and it investigates how multiple concepts need to be combined to retrieve the best video shots on data from the TREC video (TRECVID[1]) search task of 2005.

The goal of the TRECVID search task is the retrieval of video shots which are relevant to the user. We adopt the definition of a semantic concept from Snoek at al. [Snoek *et al.*, 2006b] where a concept is defined as something which must appear clearly in the static key frame of the video shot to return. Thus the expression does not cover concepts which are only represented in the audio content or more abstract concepts such as *World Peace*.

The generally used approach to detect concepts in video data is to train several so called detectors through positive and negative examples in order to recognize the appearance of a concept. The problem of how to determine the set of required detectors applies even for limited domains. The most commonly used metaphor for this problem is to define an infinite *semantic space*. The objective is to create detectors for a certain set of concepts which should allow to answer all possible queries [Naphade *et al.*, 2005]. Besides the issue of selecting appropriate concepts for a particular domain, another question is how to handle requests for concepts which are not directly present in the set of available concepts or require the usage of more then one concept. For example, a user might search for *Condoleezza Rice* but the search system only has the concepts *Face* and *Women* available. Due to the lack of knowledge about the structure of the *semantic space*, it is not an option to simply increase the number of detectors up to the point where all requested concepts are covered. Thus, some concepts have to be expressed as a combination of concepts for which detectors exist. This problem has not been satisfactory addressed yet [Snoek *et al.*, 2006a].

Users searching an image or video collection cannot be expected to know the concepts that have been used in the concept detection step. User queries usually either consist of a few keywords (e.g. *Beach*) or more elaborate natural language requests (e.g. *Find me pictures of a beach with people.*). In the best case the query contains one or more of the concept names and syntactic matching is sufficient, however often this will not be the case (for instance, in TRECVID available concepts include *Outdoor*, *Waterscape* and *People* but not beach). Hence, the first task is the extraction of the concepts underlying the queries. The query concepts and the concepts available for the collection are then matched and a ranking of relevant concepts is derived that shall resemble the information need expressed in the query as closely as possible . When viewing the relevant concepts analogously to relevant documents in the information retrieval setting, the quality of concept extraction can be evaluated with information retrieval methods. In order to create relevance judgments we performed a user study and evaluated our automatic concept query formulation algorithms against the queries formulated by the users. This has the advantage that automatic query formulation can be evaluated independently.

In previous work [Aly *et al.*, 2007], we evaluated approaches to combing two concepts to form a new composite concept. In this paper we extend this work to the combination of more than two concepts. Prior to this we introduce a framework of formulas which simplifies the process of building new scoring formulas. Given the ranked list of concepts from our approaches above there is still the open question of which of those to choose for the actual combi-

[1] http://www-nlpir.nist.gov/projects/t01v/

nation. We introduce a number of mechanisms to solve this task.

The rest of this paper is organized as follows: In Section 2 we briefly give an overview of related work. Section 3 describes the methods utilized for mapping user queries to a ranked list of concepts. In Section 4 the scoring of composite concepts is described, and ways to choose concepts from a ranked list are evaluated. The following section describes the experiments performed to evaluate to presented methods (Section 5). Finally, Section 6 concludes and proposes future work.

2 Related Work

A lot of work on concept detection has been done in the context of the TRECVID Workshop [Smeaton *et al.*, 2006]. The search task in TRECVID requires the participants to return for each topic the first 1000 entries of a ranked list of shots. Each topic consists of a textual part and example multimedia material. Together with the main raw collection the participants get the results of the high-level feature extraction task and speech transcripts from an Automatic Speech Recognition System (ASR). We use the data of TRECVID 2005 to evaluate our methods.

The query concept extraction process is aided by background knowledge in the form of thesauri or more general ontologies. These are hierarchical and associative structures usually created manually by experts that capture domain-independent or domain-dependent knowledge. The most widely used knowledge base for the general domain today is WordNet [Fellbaum, 1998], a semantic dictionary whose content expresses common-sense world knowledge which is also a popular knowledge source for TRECVID participants. WordNet is applied in two directions: expansion of queries, documents and concept descriptions and the determination of relatedness scores between concepts. In [Koskela *et al.*, 2006; Sjóberg *et al.*, 2006] the given concepts were located in WordNet and the concept description was expanded by the concept names' synonymous terms. The topics were then syntactically matched against the concept descriptions. A more general approach was adopted in [Campbell *et al.*, 2006], namely the Adapted Lesk algorithm [Banerjee and Pedersen, 2002]. It also relies on the overlap between topic text and the concept descriptions, but furthermore takes advantage of WordNet's graph structure and considers related concepts and their descriptions as well. Instead of matching queries and concepts based on their descriptions, their relatedness can also be measured directly on WordNet's graph structure. [Snoek *et al.*, 2006a] link all possible query nouns and the concepts to entries in WordNet and determine their relatedness by applying Resnik's Information-based algorithm [Resnik, 1995]: the relatedness between two concepts equals the information content of their most specific common parent.

Exploiting the relationship between concepts is related to the creation detectors for combined concepts. The idea behind the multi-concept relationships is to let the scoring process for a concept be influenced by the relationship with other related concepts, for example the likelihood of observing the concept *Bus* decreases when observing an *Indoor* setting with a high score. Rong Yan [Yan, 2006] give a good overview of the available techniques to do this. The link to our proposed method is that the multi-concept relationship approach tries to improve detectors by considering the presence of related concepts and the presented approach creates new concepts. Thus both consider multiple concepts.

The MediaMill Group [Snoek *et al.*, 2006b] evaluated several ways of combining *low-level* features, namely color-histograms and associated text generated by performing ASR, into high-level concept detectors. Each strategy is based on a vector of a number of low-level features. The detector relies on support vector machines (SVM) [Vapik, 1998] and is trained on designated training data in order to accurately assign scores to shots from other data sources. In their experiments, they investigated different types of low-level features and their respective impacts: 1) video features only, 2) associated text only, 3) video features and associated text (early fusion), 4) a combination of the output of 1) and 2) (late fusion) and finally 5) a combination of the output of methods 1)-4). For TRECVID 2005 they trained and evaluated 101 concept detectors on approximately 30,000 shots. On average method 1) was performing the best on the TRECVID dataset. Hence textual features were not particularly beneficial. The reason for this was not explicitly researched. It seems plausible that the data was not suitable for speech recognition - because ASR and machine translation from Chinese and Arabic speech introduced too much noise. The output of their set of concept detectors are rankings for the search data together with the ground-truth and rankings on the test dataset that was used for the the high level feature extraction evaluation. We use the scores of the 101 concept detectors on the search data to verify our ideas and employ the results from the test data to judge the quality of a detector.

3 Concept Extraction

3.1 WordNet

WordNet [Fellbaum, 1998] is an online lexical database developed at Princeton University that was inspired by psycholinguistic theories. It is continuously enlarged and updated by human experts and as already been pointed out can be viewed as a general domain knowledge base. WordNet's building blocks are sets of synonymous terms[2] (so-called *synsets*) each representing one lexical concept that are connected to each other through a range of semantic relationships. Relations between terms instead of synsets exist as well but are not very frequent.

A small part of WordNet is shown as a graph in Figure 1: the synsets are represented by nodes and an edge exists between two synsets if they are semantically related. A synset can consist of several terms and each term is contained in m synsets with m being the number of senses the term has (identified by the sense numbers $\#X$ in Figure 1).

Relationships exist only between synsets of the same word type, hence there are separate structures for nouns, verbs, adjectives and adverbs with nouns make up the largest fraction of WordNet. We restrict ourselves to a short overview of the relations that we utilized in our approach. In the following paragraphs, s_1 and s_2 represent synsets and t_1 and t_2 represent terms.

Hypernymy, hyponymy (nouns): s_1 is a *hypernym* of s_2 and s_2 is a *hyponym* of s_1 if s_1's meaning contains s_2's, e.g. $\{$*vessel, watercraft*$\}$ is a hypernym of $\{$*ship*$\}$.

Hypernymy, troponymy (verbs): s_1 is a *hypernym* of s_2 and s_2 is a *troponym* of s_1 if s_2 is a certain manner of s_1 e.g. $\{$*walk*$\}$ is a hypernym of $\{$*stroll, saunter*$\}$.

[2] A term can be a single word, a compound or a phrase.

Holonymy, meronymy (nouns): s_1 is a *holonym* of s_2 and s_2 is a *meronym* of s_1 if s_2 is a member of or part of s_1, e.g. {*fleet*} is a holonym of {*ship*}.

Sibling (nouns, verbs): s_1 and s_2 are *siblings* if they share a direct hypernym e.g. {*ship*} and {*yacht, racing yacht*} are siblings as they share the hypernym {*vessel, watercraft*}.

Entailment (verbs): s_1 *entails* s_2 if s_1 implies s_2 e.g. {*buy, purchase*} entails {*pay*}.

Verb group (verbs): s_1 and s_2 belong to the same *verb group* if they have a similar meaning (manually grouped by human experts)

Derivationally related form (nouns, verbs): the noun t_1 has a derivationally related noun (or verb) form t_2 if they are morphologically related and semantically linked e.g. the noun *machine* and the noun *machinist* or the verb *cook* and the noun *cook*.

Similar to (adjectives): s_1 and s_2 are similar to each other if one is more general than the other e.g. {*yellow, yellowish, xanthous*} and {*chromatic*} are similar.

Furthermore, WordNet also provides glosses for all synsets, which consist of definitions or sentences that show the synset's usage. Examples are here the gloss "a craft designed for water transportation" for {*vessel, watercraft*} and the gloss "of the color intermediate between green and orange in the color spectrum; of something resembling the color of an egg yolk" for {*yellow, yellowish, xanthous*}.

3.2 Extracting Concepts from Queries

In order to determine the ranking of concepts that best describes a query, the relatedness scores between the concepts and the query need to be determined. This can be done on two levels: on the synset level or on the term level. A number of algorithms have been proposed [Banerjee and Pedersen, 2002; Patwardhan and Pedersen, 2006; Resnik, 1995; Jiang and Conrath, 1997; Lin, 1998; Lea, 1998; Wu and Palmer, 1994] that differ in what part of WordNet they utilize - glosses, synset terms and various relationship types with different weighting schemes. In the next two sections, the approaches chosen for our experiments are explained in greater detail. First, a term-level gloss-based approach is presented, then three synset-level graph-based approaches are introduced. In both cases, it is assumed that the concepts have been (manually) linked to the correct corresponding synsets in WordNet.

Using WordNet's glosses

This approach does not require extensive preprocessing of the queries as the graph-based approaches do. Furthermore, it is not restricted by word types: if a query noun is found in a gloss of a verb for example, the verb concept is deemed likely to be relevant. Graph-based approaches on the other hand usually cannot cross part-of-speech boundaries. The gloss of a concept's synset as well as the glosses of related synsets are used to create a *concept document*. The type of relations used, the maximum depth and the glosses' weightings are freely settable parameters. A depth of 0 means that only the gloss of the synset itself is added to the concept document, a depth of 1 includes the directly related synsets as well, etc. Possible weighting schemes include uniform weighting of every gloss and linear weighting, which linearly decreases the weight of the glosses the larger the depth. The concept documents are then treated as a document collection and keyword-based text retrieval is performed - the document ranking corresponds to the concept ranking.

> head of state chief of state the chief public representative of a country who may also be the head of government chancellor premier prime minister the person who is head of state (in several countries) Prime Minister PM premier the person who holds the position of head of the government in England president the chief executive of a republic President of the United States United States President President Chief Executive the person who holds the office of head of state of the United States government; "the President likes to jog every morning" sovereign crowned head monarch a nation's ruler or head of state usually by hereditary right

Figure 2: A concept document for the concept *government leader* which was mapped to WordNet's {*head of state, chief of state*} synset. Hyponym-related synset glosses up to a depth of 1 were added.

An example of a concept document is shown in Figure 2. It also demonstrates one of WordNet's drawbacks: WordNet is updated and altered by human experts, hence inevitably there will be a bias in what concepts make it into WordNet and what concepts do not. While the Prime Minister of the United Kingdom and the President of the United States occur as concepts, the heads of almost all other countries cannot be found.

Using WordNet's Graph Structure

Determining the semantic relatedness scores requires a number of preprocessing steps as depicted in Figure 3. First of all, the word types of the query terms need to be found with a part-of-speech (POS) tagger. Since most terms have more than one sense their meaning in this particular context needs to be determined. This step is called word sense disambiguation (WSD) and can also utilize WordNet. In the simplest case the most common sense (which is provided by WordNet) is chosen.

Having located the query terms' corresponding WordNet synsets makes it possible to use graph theoretic measurements to determine the semantic relatedness between the query concepts and the given concepts. A very simple measure is the hierarchical shortest path measurement $rel_{HS}(s_1, s_2)$: how many hypernymy/hyponymy edges[3] len of the WordNet graph need at least to be traversed to reach a synset s_2 from synset s_1? There are problems though, as WordNet is a small-world network [Sigman and Cecchi, 2002], hence within the connected part of the graph two nodes can always be reached within a few steps. Another issue is that the synsets close the root node are quite dissimilar from each other (e.g. {*object, physical object*} is a direct hypernym of emph{ice}) whereas deep in the hierarchy they tend to be very similar (e.g. {*cab, hack, taxi, taxicab*} is a direct hypernym of {*minicab*}). For

[3] If only the hypernymy/hyponymy relationship is utilized, the noun graph becomes hierarchical and the measures are often called *semantic similarity* instead of the more general *semantic relatedness* which considers all types of relationships.

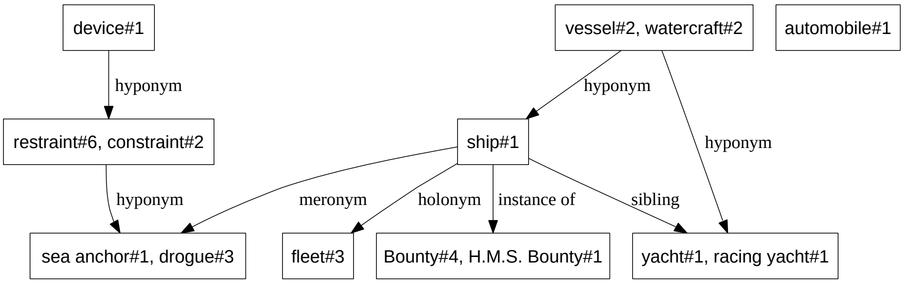


Figure 1: A part of WordNet's noun graph

these reasons, measurements usually prohibit edge walks along certain relationship types [Hir, 1998], they include information about WordNet's depth [Lea, 1998; Wu and Palmer, 1994] or exploit information drawn from analysing large corpora [Resnik, 1995; Jiang and Conrath, 1997; Lin, 1998]. An overview of five WordNet-based relatedness measurements and their performances is given in [Budanitsky and Hirst, 2006].

[Lea, 1998] determine the relatedness score of two synsets also solely based on the hypernymy/hyponymy relationship. In contrast to rel_{HS}, the number of edges between s_1 and s_2 is scaled by the maximum depth of the WordNet hierarchy.

$$rel_{LC}(s_1, s_2) = -\log \frac{len(s_1, s_2)}{2 \times \underset{s \in WordNet}{max}[depth(s)]} \quad (1)$$

[Wu and Palmer, 1994] exploit not the global depth of WordNet but instead the depth of the *lowest super-ordinate* (lso) of the two synsets, that is the most specific synset that subsumes both synsets (e.g. the lowest super-ordinate for {*ship*} and {*yacht, racing yacht*} is their common hypernym {*vessel, watercraft*}). Let $z = lso(s_1, s_2)$, then the relatedness is given by

$$rel_{WP}(s_1, s_2) = \frac{2 \times depth(z))}{len(s_1, z) + len(s_2, z) + 2 \times depth(z)} \quad (2)$$

4 Concept Combination for Search

This section studies the possibilities to combine multiple concepts in video retrieval. Our previous studies revealed that the combination of two concepts improves performance. We believe the extension to multiple concepts is beneficial because many concepts stand in a inherent *leads directly to* relationship to others. For example, the correct detection of the concept *Face* directly leads to the presence of the the concept *Person*. Thus, if searching for *A person in the street* the search for the concepts *Person Face Street Outdoor* will be beneficial in case we have a good *Face* and *Outdoor* detector. This, of course, assumes that persons are mainly shown with their face into the camera and that all streets are outdoor.

The rest of this section proceeds as follows: first the basic operations used are introduced (Section 4.1), followed by an overview of the new ranking formulas tested (Section 4.2). In Section 4.3 methods on how to identify the concepts to use in a query, given the list of concepts selected by the concept extraction approach presented in the previous section.

$$\begin{aligned} \chi(c, s_j) &= \text{score } s_j \text{ using } c && (3)\\ \psi(C, s_j) &= \text{score } s_j \text{ using } C && (4)\end{aligned}$$

χ Functions:

$$\begin{aligned} r(c, s_j) &= \text{original score} && (5)\\ factor(c, s_j) &= \log(r(c, s_j)) && (6)\\ smooth(c, s_j) &= \frac{\sum_{i=j-nh}^{j+nh} \delta(|i-j|) r(c, s_i)}{\sum_{i=j-nh}^{j+nh} \delta(|i-j|)} && (7)\\ weighted(c, s_j) &= ap(c) \cdot r(c, s_j) && (8)\end{aligned}$$

Combination Functions:

$$\begin{aligned} sum(\chi, C, s_j) &= \frac{\sum_{c \in C} \chi(c, s_j)}{|C|} && (9)\\ sumC(\chi, \psi, C, s_j) &= \sum_{c \in C} \chi(c, s_j) \frac{\psi(C \setminus c, s_j)}{|C| - 1} && (10)\end{aligned}$$

Figure 4: Basic Functions

4.1 Basic Operations for Ranking Functions

We refined the list of scoring functions and extended them to handle more then two base concept detectors. Formally their task is to calculate the score for a shot s_j based on the detectors of a set of concepts C. We identified two different classes of basic operations: 1) functions which only use one concept c for their score calculation χ and 2) combination functions which operate on a set of concepts ψ.

$r(c, s_j)$ (5) simply returns the score of the shot s_j as calculated by the detector for concept c. Instead of introducing a combination function that sums the scores of two concept ranking functions, and another that multiplies them, we define a function $factor(c, s_j)$ (6). Summed logarithmic scores produce the same ordering of shots as multiplied original scores. Using the function $factor$ is beneficial due to less numerical precision loss in case of a multiplication. Another reason is to keep the set of combination operations small.

The function $smooth$ (7) assumes that it is more likely that a concept c appears in the shot s_j if it also appears in previous or following shots. Similar approaches have been investigated using the text from automatic speech recognition associated with shots [Hauptmann *et al.*, 2006]. We define the surrounding neighborhood as a fixed number nh of shots before and after the actual shot s_j that contribute to the score of s_j. We expect that shots which are further away from s_j to rank, to have less influence on the likelihood of the presence of concept c. We model this fact in

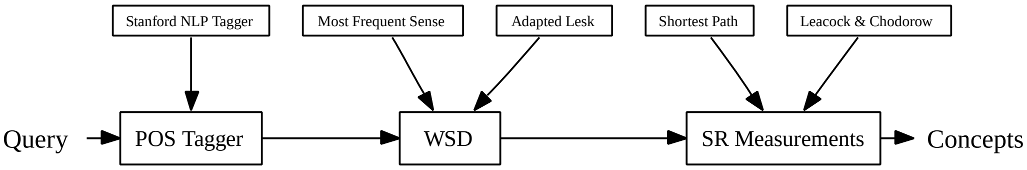


Figure 3: Converting queries to concepts.

$$
\begin{array}{rcll}
add(C, s_j) & = & sum(r, C, s_j) & (11) \\
cbw(C, s_j) & = & sum(weighted, C, s_j) & (12) \\
mult(C, s_j) & = & exp(sum(factor, C, s_j) & (13) \\
n(C, s_j) & = & sumC(r, sum(smooth), C, s_j) & (14) \\
\psi_{log} & = & log(sum(smooth)) & (15) \\
\psi_{logs} & = & log(sum(smooth)) & (16) \\
nm(C, s_j) & = & exp(sumC(factor, \psi_{log}, C, s_j)) & (17) \\
nw(C, s_j) & = & sum(weighted, \psi_{logs}, C, s_j) & (18) \\
sa(C, s_j) & = & sum(smooth, C, s_j) & (19) \\
sm(C, s_j) & = & exp(sum(\psi_{log}, C, s_j)) & (20)
\end{array}
$$

Figure 5: Concrete Functions

a weighting function δ which takes the absolute distance $[0 \ldots nh]$ of a shot as an argument and returns a weight in the interval $[0 \ldots 1]$ to weight the score of the shot. We define the smooth function as follows:

Let $\delta(x)$ be a function of the distance from the shot that determines the influence of the neighboring shots. We created three alternatives for δ. The first version $\delta_{constant}(x) = 1$ weights all shots uniformly. This will serve as a base line. The function $\delta_{linear}(x) = 1 - \frac{x}{nh+1}$ lowers the weight of a shot linearly in its distance to s_j. The furthest shots on both sides will still have a small positive weight. The version $\delta_{exp}(x) = -exp(-x)$ lowers the weight of shots in an exponential fashion.

As ranking functions differ in their precision of detecting their base concepts [Snoek *et al.*, 2006b] we created a possibility to weight their output. $weighted(c, s_j)$ (8) weights the outcome of the ranking function for concept c by the average precision $ap(c)$ achieved for this concept on the test dataset.

We identified two basic combination methods. The first, $sum(\chi, C, s_j)$ (9) takes as χ a function which should be executed for each particular concept. C is the set of concepts which it should perform the function for. It then sums up the results from execution of χ on each concept from C. To keep the scores within $[0..1]$ it divides the result from the sum by the number of concepts. The function $sumC(\chi, \psi, C, s_j)$ (10) allows each summand to be calculated from two parts: A score from a function using the current concept on the shot (Class χ) and a function which operates on all other concepts in the set passed to the function ψ. In order to assure range intervals the output of this function is divided by the number of concepts the calculation is build on. The rational is that the score for a concept on a certain shot could be influenced by the performance of other ranking functions.

4.2 Ranking With Combined Concepts

Based on these basic operations we extended methods we already studied in [Aly *et al.*, 2007] for the use of two concepts. The function $add(C, s_j)$ (11) is a simple summation of the base scores. The derived version $cbw(C, s_j)$ (12) weights the summands by their AP in the test set. In the function $mult(C, s_j)$ (13) first the sum of the logarithms out of each base score is calculated. At the end a $exp()$ function is applied to get the scores again in the interval [0..1].

The Neighbor function $n(C, s_j)$ (14) considers all base scores multiplied with the average of the smoothed scores of the other concepts. $nm(C, s_j)$ (17) is an extension of the $mult$ function which is weighting the individual scores by the $log()$ of averaged smoothed scores of other concepts.

As described above it is the case that some concept detectors are less precise then others. Therefore, we create versions of the n function, namely $nw(C, s_j)$ (18) which additionally weights the score of the individual scoring function by the AP of the detector in the test data.

A new class of scoring functions that operate only on smoothed values. The $sa(C, s_j)$ (19) takes the average of all smoothed scores. The function $sm(C, s_j)$ (20) does the equivalent but with the described method to effectively multiply summands.

4.3 Selection of Concepts

The input for the concept combination algorithms are ordered lists of concepts. The problem now is what concepts to employ during the ranking. The most obvious is of course to combine the whole list of concepts. However there are a lot of concepts which were only chosen once or have very little effect on the search performance. Out concept extraction method could return all available concepts, in which case some of them will definitely have negative impact on search performance. To overcome this problem we use a Top-N approach to only select the first n concepts of a list.

5 Evaluation

5.1 Concept Extraction

Not every concept could be attached to exactly one synset in WordNet, some concepts were linked to several synsets. The concept *natural disaster* for instance does not occur as such in WordNet 2.1 but instead was represented by the synsets {*flood, inundation, deluge, alluvion*}, {*earthquake, temblor, seism*}, {*storm, violent storm*} and {*volcanism*}. Another problem arose for several person concepts like *A. Sharon* or *E. Lahoud* which have no representation in WordNet. In those cases, the concepts were added to WordNet as instances of an appropriate synset such as {*head of state, chief of state*} and their gloss consists of the concept name alone.

Run	Depth	MAP	P@5
uniform/sibling	0	0.268	0.246
uniform/noSibling	4	0.296	0.225
linear/sibling	4	0.286	0.200
linear/noSibling	4	0.297	0.225

Table 1: Results for the gloss-based approach. 4 types of runs were performed with varying depth: uniform weighting with siblings, uniform weighting without siblings, linear weighting with siblings and linear weighting without siblings. The best performing run for each type in terms of P@5 is shown together with its depth parameter.

Golden Standard

In order to evaluate the different concept extraction algorithms separately from the concept combination part, we developed a *golden standard* for the TRECVID 2005 topics. 25 users were given the topics and asked to return those concepts of the 101 available concepts that best describe the information need expressed in the topics. No restriction was given on the number of concepts the users could choose. On average users chose 6.96 concepts per topic. The spread between the users was quite large: the lowest average was 4.42, the maximum average was 10.67. Furthermore, for many topics the agreement between the users was surprisingly low. 11 of the 25 topics had more than 20 different concepts returned at least once. We derived a concept ranking for each topic from this survey by considering all concepts of a topic that more than 50% of the participants had chosen and ranked them accordingly. The average number of concepts per topic was reduced 3.2, the minimum number of concepts is 2 (for topics 155, 157, 164, 166, 170) and the maximum number is 7 (for topic 159). As mentioned before, viewing the relevant concepts as relevant documents in the information retrieval setting, allows us to evaluate our approaches with information retrieval performance measures, namely mean average precision (MAP) and precision at 5 documents (P@5).

Gloss-Based Approach

For the gloss-based conversion, we utilized the *hyponymy*, *meronymy*, *entailment*, *sibling*, *verb group*, *derivationally related form* and *similar to* relationships described in Section 3.1. The depth was varied between 0 and 5 and the uniform and linear weighting schemes were tested. Finally due to the large volume of siblings for a number of concepts we also considered the influence they have and run the algorithms with and without the inclusion of this particular relationship. For the retrieval experiments we employed the Lemur Toolkit for Information Retrieval[4]. The documents and topics were stemmed and stopwords were removed. The language modeling approach with Jelinek-Mercer smoothing was used for retrieval purposes. We report the results in Table 1. Since the concept combination step relies on the Top-N ranked concepts, P@5 was deemed a more important measure than MAP.

Surprisingly, using the synset glosses without adding related concepts performs best for P@5. The sibling relationship hurts the performance across all runs tested except for one. The weighting scheme does not have a large influence on the results.

Run	MAP	P@5
rel_{HS}	0.370	0.217
rel_{LC}	0.366	0.217
rel_{WU}	0.345	0.200

Table 2: Results for the graph-based approaches.

Graph-Based Approaches

POS tagging the queries was performed with the Stanford NLP Tagger[5]. The relatedness measures rel_{HS}, rel_{LC} and rel_{WU} introduced in Section 3.2 were investigated in two variants: 1) word sense disambiguation of the query concepts was reduced to choosing the most common sense and 2) the query terms q_i were tested with all their senses against the concepts and the maximum relatedness score was returned:

$$rel(q_i, s_j) = \underset{q_i \in s_k}{max}[rel(s_k, s_j)]. \tag{21}$$

The differences between 1) and 2) proved to be small, in Table 5.1 the results of the most common sense approach are presented. While MAP considerably increases over the gloss-based approach, $P@5$ is harmed. Thus, for the TRECVID 2005 topics, using only the glosses of the synsets corresponding to the concepts is the best approach among all tested ones.

5.2 Combined Concept Search

Most of our combinations methods depend on the $smooth$ method. There are two free parameters which will affect the performance: The degrading function δ and the size of the neighborhood nh. We first evaluate the best parameter setting performance in order to justify which of the combinations to employ in the later combination. First we evaluated which δ function was the best. We did this by taking the average AP of the first top-n concepts for each query with this δ. We did this for top-n following $\{2, 4, 6, 8\}$. The results are shown in Table 3 (a).

δ_{const} performs for all top-n values best, thus it is used to evaluate the variations of nh. Results with other δ functions are not shown due to space limitations but did not yield other conclusion. Table 3 (b) shows the MAPs of the r and the $smooth$ function with δ_{const} for $nh \in \{2, 5, 8\}$. We stopped at $nh = 8$ because it was the first drop in the MAP. As we deem it to be realistic that this will not improve with higher nh we limit ourselves to this sample. The setting $nh = 2$ is always worse than the other results. The setting $nh = 5$ improves MAP compared to r by 24% in average. With $nh = 8$ the improvement is a bit lower with 18%. This brings us to the conclusion that we use for our combination functions $smooth$ with the parameter setting $\delta = \delta_{const}$ and $nh = 5$.

After evaluation of the $smooth$ method we use the best settings to test the concept combination methods. We evaluated them against the Golden Standard and our concept extraction methods. The parameter N, the Top-N extracted concepts was tested in a range from $2, 4, 6, 8$. It turned out that Top-2 showed the best MAPs. Table 4 shows the results for the experiments with $N = 2$. The *linear / sibling d=4* performed surprisingly similar but a bit stabler than the Golden Standard. The other gloss based methods were always worse in terms of MAP. The graph based concept extraction methods show very poor performance for all com-

[4] http://www.lemurproject.org/

[5] http://nlp.stanford.edu/software/tagger.shtml

top-n	δ_{const}	δ_{lin}	δ_{exp}
2	0.0190	0.0182	0.0156
4	0.0175	0.0168	0.0144
6	0.0165	0.0160	0.0138
8	0.0161	0.0156	0.0135

δ functions, averaged over all neighborhoods
(a)

top-n	r	nh=2	nh=5	nh=8
2	0.0173	0.0114	0.0233	0.0223
4	0.0175	0.0119	0.0209	0.0196
6	0.0163	0.0113	0.0197	0.0186
8	0.0156	0.0110	0.0191	0.0181

$smooth$ with δ_{const} and different nhs against base score r
(b)

Table 3: Evaluation of smoothing parameters on MAP

bination methods. Looking at the performance of the combination methods using the Golden Standard one can see that the best methods are $mult$ and nm. Surprisingly $mult$ is in average of all concept extraction methods the best. It should be noted that in average the variance of all combination methods is rather low. The results for the Golden Standard with Top-8 can be found in the lowest row of the table. Here the sm method performs best, which leads to the conclusion that the number of included concepts influences the performance of the combination method.

6 Conclusion & Future Work

In this paper we presented our approach to TRECVID's video retrieval search task and focused on two particular problems: 1) the conversion from queries in natural language format to a ranked list of concepts and 2) the combination of the returned concepts to improve the ranking of the shots. WordNet was chosen as mediator between the queries and concepts. Several algorithms utilizing WordNet's content (gloss-based approaches) and structure (graph-based approaches) were investigated. Given the ranked list of concepts, several combination and scoring methods were applied in order to gain insights into the optimal number of concepts to combine and the optimal scoring function.

While gloss-based concept extraction scored not considerably worse than the golden standard in the search task (with one gloss-based run regularly outperforming the golden standard), the graph-based approaches performed about 50% worse. Using only the Top-2 concepts for scoring proved to be the most effective. For the Top-2, our method $mult$ and nm performed the best. The latter one uses the timely smoothed values of shots. sm, a method which only considers smoothed values, performed best for the Golden Standard using the 8 best concepts

There are several directions for future work. One important issue is how to deal with named entities - topic 149 (*Find shots of Condoleezza Rice*) is such an example. The graph-based concept extraction algorithms did not return a single concept, since *Condoleezza Rice* does not appear in WordNet. It would therefore be beneficial to have access to an alternative knowledge source like Wikipedia or a newspaper corpus as a fall-back option. Furthermore, classifying the queries into several different types [Volkmer *et al.*, 2006] and creating query type dependent concept extraction and scoring algorithms can also help to alleviate the current problems.

The score combination algorithms presented so far do not adequately take into account the quality of the concept detectors. While some concepts are detectable with high precision (e.g. *Face* or *Sports*), others pose great difficulties (e.g. *Police Security*). This additional information can be exploited by weighting the concept scores according to the quality of the concept detectors.

References

[Aly *et al.*, 2007] Robin Aly, Djoerd Hiemstra, and Roeland Ordelmann. Building detectors to support searches on combined semantic concepts. In *Proceedings of the SIGIR-MMIR: Multimedia Information Retrieval Workshop - New Challenges in Audio Visual Search -*, 2007. to be published.

[Banerjee and Pedersen, 2002] S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145, 2002.

[Budanitsky and Hirst, 2006] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[Campbell *et al.*, 2006] Murray Campbell, Alexander Haubold, Shahram Ebadollahi, Dhiraj Joshi, Milind R. Naphade, Apostol Natsev, Joachim Seidl, John R. Smith, Katya Scheinberg, Jelena Tesic, and Lexing Xie. IBM research TRECVID-2006 video retrieval system. In *Proceedings of the 4th TRECVID Workshop*, 2006.

[Fellbaum, 1998] Christiane Fellbaum. *Wordnet: An Electronic Lexical Database*. The MIT Press, 1998.

[Hauptmann *et al.*, 2006] A.G. Hauptmann, R. Baron, M. Christel, R. Conescu, J. Gao, Q. Jin, W.-H. Lin, J.-Y. Pan, S. M. Stevens, R. Yan, J. Yang, and Y. Zhang. Cmu informedias TRECVID 2005 skirmishes. In *Proceedings of the 3rd TRECVID Workshop*, 2006.

[Hir, 1998] *Wordnet: An Electronic Lexical Database*, chapter Lexical chains as representations of context for the detection and correction of malapropisms, pages 305–332. The MIT Press, 1998.

[Jiang and Conrath, 1997] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings on International Conference on Research in Computational Linguistics*, pages 19–33, 1997.

[Koskela *et al.*, 2006] Markus Koskela, Peter Wilkins, Tomasz Adamek, Alan F. Smeaton, and Noel E. O'Connor. TRECVid 2006 experiments at dublin city university. In *Proceedings of the 4th TRECVID Workshop*, 2006.

[Lea, 1998] *Wordnet: An Electronic Lexical Database*, chapter Combining local context and WordNet similarity for word sense identification, pages 265–283. The MIT Press, 1998.

[Lin, 1998] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning*, 1998.

Top2 of	add	cbw	$mult$	n	nm	nw	sa	sm
Golden Standard	0.0715	0.0615	**0.0801**	0.0588	**0.0801**	0.053	**0.0621**	**0.0719**
uniform / sibling d=4	0.0525	0.0564	0.0548	0.0553	0.0548	0.055	0.051	0.0566
linear / noSibling d=4	0.0515	0.0541	0.053	0.0567	0.053	0.0556	0.0481	0.0465
linear / sibling d=4	**0.0736**	**0.0741**	0.0754	**0.0713**	0.0754	**0.0710**	0.0606	0.064
uniform / noSibling d=0	0.0443	0.048	0.0413	0.0477	0.0387	0.048	0.0433	0.0341
rel_{HS}	0.0342	0.0335	0.0316	0.0317	0.0316	0.0314	0.0334	0.0278
rel_{LC}	0.0339	0.0332	0.0312	0.0315	0.0312	0.0312	0.0334	0.0278
rel_{WU}	0.0339	0.0334	0.031	0.0316	0.031	0.0311	0.0329	0.0262
Average:	0.0494	0.0493	***0.0498***	0.0481	0.0495	0.047	0.0456	0.0443
Golden Standard Top8	0.0600	0.0372	0.0593	0.0464	0.0593	0.0266	0.0408	***0.0681***

Table 4: MAP values for each Concept Extraction and Combination Method. **bold**: best for this combination method.

[Naphade and Smith, 2004] Milind R. Naphade and John R. Smith. On the detection of semantic concepts at trecvid. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 660–667, New York, NY, USA, 2004. ACM Press.

[Naphade *et al.*, 2005] M.R. Naphade, L. Kennedy, J.R. Kender, S.-F. Chang, J.R. Smith, P. Over, and A. Hauptmann. A light scale concept ontology for multimedia understanding for trecvid 2005. Technical report, IBM T.J. Watson Research Center, 2005.

[Patwardhan and Pedersen, 2006] Siddharth Patwardhan and Ted Pedersen. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *EACL*, 2006.

[Resnik, 1995] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[Sebe, 2003] Nicu Sebe. The state of the art in image and video retrieval. In *Image and Video Retrieval*, volume Volume 2728/2003, pages 1–8. Springer Berlin / Heidelberg, 2003.

[Sigman and Cecchi, 2002] Mariano Sigman and Guillermo A. Cecchi. Global organization of the wordnet lexicon. *PNAS*, 99(3):1742–1747, 2002.

[Sjóberg *et al.*, 2006] Mats Sjóberg, Hannes Muurinen, Jorma Laaksonen, and Markus Koskela. PicSOM experiments in TRECVID 2006. In *Proceedings of the 4th TRECVID Workshop*, 2006.

[Smeaton *et al.*, 2006] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[Snoek *et al.*, 2006a] Cees G. M. Snoek, Jan C. van Gemert, Theo Gevers, Bouke Huurnink, Dennis C. Koelma, Michiel van Liempt, Ork de Rooij, Koen E. A. van de Sande, Frank J. Seinstra, Arnold W. M. Smeulders, Andrew H. C. Thean, Cor J. Veenman, and Marcel Worring. The MediaMill TRECVID 2006 semantic video search engine. In *Proceedings of the 4th TRECVID Workshop*, Gaithersburg, USA, November 2006.

[Snoek *et al.*, 2006b] Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430, New York, NY, USA, 2006. ACM Press.

[Vapik, 1998] V. N. Vapik. *Learning Theory: Inference from Small Samples.* Wiley, 1998.

[Volkmer *et al.*, 2006] Timo Volkmer, S.M.M. Tahaghoghi, and James A. Thom. RMIT university video retrieval experiments at TRECVID 2006. In *Proceedings of the 4th TRECVID Workshop*, 2006.

[Wu and Palmer, 1994] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, 1994.

[Yan, 2006] Rong Yan. *Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval.* PhD thesis, Canegie Mellon University, 2006.

Retrieval of technical drawings in DXF format - concepts and problems

Nadine Weber, Andreas Henrich
University of Bamberg
Chair of Media Informatics
D-96052, Bamberg, Germany
{nadine.weber, andreas.henrich}@wiai.uni-bamberg.de

Abstract

Nowadays, launching new products in short intervals is a critical factor for success to persist on the global market. At the same time many enterprises call for cost reduction in all of their divisions. Especially development departments have to increase their effectiveness and efficiency by lowering development time and costs to preserve the competitiveness of the company. Since development processes are affected by a plethora of information and knowledge, a starting point for time and cost reduction is to reuse existing design knowledge. This knowledge consists of both text documents and special kinds of artifacts such as 3D models, technical specifications, technical drawings, or bills of material. While the retrieval of text documents and 3D models is already explored widely in the research area, retrieving technical drawings is still a complex and interesting field of investigation. Therefore, this paper deals with existing search techniques for technical drawings and the problems emerging from implementing such techniques for *Drawing Interchange Format* (DXF) drawings. Moreover, we propose a general procedure for the extraction of features from such drawings to solve these problems.

1 Introduction

Today, the development of new products or product versions in the domain of mechanical engineering takes place by designing 3D models on modern Computer-Aided Design (CAD) systems. By automatically generating a drawing from a designed 3D model, these systems facilitate the creation of technical drawings needed for the manufacturing process. This leads to thousands of digital drawings stored in Product Data Management systems or on file systems. Thus, a tremendous source of information is available which is not further used in many cases; most of the data lies idle because their existence is unknown or they are not retrievable. Consequently, the reuse of such useful knowledge can contribute to reduce time and costs in the product development process.

For this purpose, a retrieval system is necessary to enable searching for technical drawings. But searching should not be seen as searching by name or drawing number only; it is the content of technical drawings that has to be addressed. Hence, both text-based retrieval methods (like, e.g., the vector space model or the boolean model) and content-based methods have to be applied to find appropriate results for a designer with a specific information need.

The first required step is to analyze the existing file formats. Since many different CAD systems are available, there is also a multitude of proprietary file formats which are not manageable without expensive converters. The only format with open access that can be handled by almost every CAD system is Autodesk's DXF format. DXF is a vector file format, expanded to a quasi-standard exchange format for technical drawings amongst nearly all companies. As a consequence, it makes sense to support this format with preference. Although the specification of this format is open and it is widely used, some aspects make implementing search techniques for it difficult.

For that reason, we want to consider some existing search techniques for technical drawings and their applicability for DXF drawings. Therefore, the paper is organized as follows. Section 2 examines our definition of a technical drawing. To this end, the potential information content of such a drawing is considered more precisely. In section 3 we give an overview of state-of-the-art search techniques for technical drawings in general. Before we sum up the paper by discussing our conclusions and presenting directions for further research in section 6, we point out the problems emerging from implementing a similarity concept for DXF drawings (cf. section 4) and thereupon make a proposal for how a feature extraction process can be configured (cf. section 5).

2 Definition of a technical drawing

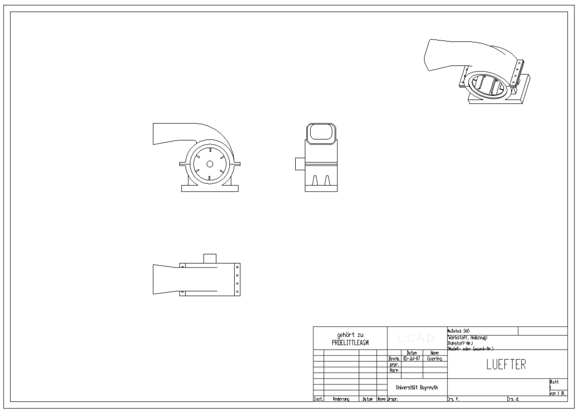


Figure 1: Simple example drawing of an exhauster.

For a better understanding of the following sections, we want to highlight our understanding of technical drawings first. According to [Conrad, 2005], we define a technical drawing as a line-based representation of scale of a part or assembly which consists of different views, slices and other additional information. Since technical drawings are an important means of communication between the design and manufacturing step in the product development process, it is necessary to consider them in searching for relevant information.

Therefore, we have to take into account that there are different types of drawings dependent on content, purpose, kind of representation, or type of creation. In [DIN, 1962] a diversity of terms such as sketch, plan, original, parent drawing, or preliminary drawing is defined. Considering these occurrences shows that a differentiation between a general drawing and a single part drawing is essential.

A single part drawing, as depicted in figure 1, predominantly describes the geometry of an individual part by displaying different views and slices of the product normally completed by dimensions. Additionally, such a drawing contains information about the product designation, the material, or tolerances integrated in form of one or more text fields. Moreover, bills of material or alternatives for the product can be contained. [Eigner and Maier, 1991]

General drawings in contrast are representations of assemblies consisting of multiple parts. The illustration of an assembly presents its mounting state to demonstrate the arrangement of the associated parts with their interdependencies. This kind of drawing is also denoted as exploded drawing or exploded view. As illustrated in figure 2, the parts are ordered in the way they would disrupt if the assembly detonated. [Vajna *et al.*, 1994]

Given the fact that we want to support engineering designers with relevant information, considering the information content of these drawings is an important task. General drawings provide mainly the structure of an assembly which can be used to create a topological representation of the product. Since this is the only information that can be extracted from such an information source, a precise study of the information content of a single part drawing has to be carried out. [Conrad, 2005] divides the content of a single part drawing into three categories: geometry data, technological information, and organizational information. While the former one gives a complete and detailed description of the geometry of primitive elements of the product such as lines, circles, or splines, the technological information contains, for example, dimensions, information about used material, or quality features. Furthermore, there is organizational information that can be divided into two groups. While the first group includes factual details such as designation and part numbers for identification and classification of the product, the second group comprises information referring to the drawing like scale, drawing format, charge number, draftsman, or creation date. Consequently, a search based on metadata as well as on geometry / topology is possible. Obviously, this requires different kinds of similarity concepts. For that reason, we give an overview of existing retrieval methods for engineering drawings in the next section.

3 Retrieval concepts for technical drawings

In general, technical drawings are illustrations of a designed product which can be stored in pixel or vector formats. Thus, a differentiation between pixel-based and vector-based retrieval methods has to be made. Obviously, both types need corresponding representations for the documents in the knowledge base and the query (the information need). The latter one can be posed in form of keywords, an example document (Query-by-Example QbE), or a sketch. While vector-based methods are rather suitable for keyword- and QbE-queries, pixel-based concepts are useful for sketches in form of scanned technical drawings or example documents.

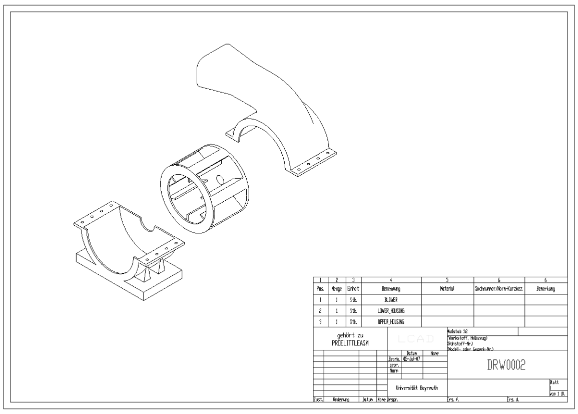


Figure 2: Exploded view of an exhauster.

3.1 Pixel-based methods

The following paragraphs give a review of existing retrieval concepts developed for image (pixel)-based engineering drawings.

Applying the Hough transform to extract global line features from a drawing is part of a retrieval method proposed by Fränti et al. [2000]. Thereby, the authors assume that engineering drawings are binary (black-and-white) images mainly consisting of line segments. For that reason, the process of line detection starts with the determination of the set of black pixels in the image. Then, each pixel is transformed into a parametric curve in the parameter domain which is also called the accumulator space. In doing so, each pixel (x, y) is described by means of the line equation $d = x \cdot \cos\theta + y \cdot \sin\theta$, where d is the distance from the origin to the line, and θ is the angle between the x-axis and the line's normal. Dependant on these two parameters, an accumulator matrix can be computed where each row corresponds to one value of the distance d and each column to one value of the angle θ. Thus, every pixel is arranged in this matrix, before a feature vector is generated. For this purpose, the authors suggest two variants. The first method reduces the matrix using a threshold value and sums up the significant coefficients in each column of the accumulator matrix. Hence, a global description of the image based on angular information is given. Since the use of this kind of information is not sufficient for large and more complex drawings, the authors present a second variant which includes positional information of the lines. This variant uses the full accumulator matrix for the generation of the feature vector and therefore allows more accurate image matching. The matching itself is done by applying a distance measure to the feature vectors of the query image and the database image. In [2001], Tabbone et al. present a method for indexing technical drawings based on the notion of F-signatures. In this approach, every binary graphical object in a drawing is represented by such an F-signature which is defined as a specific kind of histogram of forces. This histogram is generated by calculating all the forces exerted between the pixels of a same object. Therefore, a

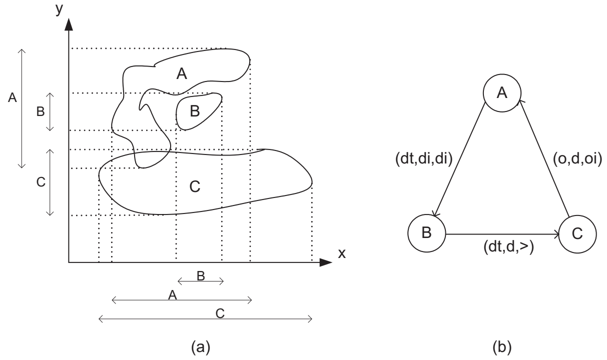


Figure 3: 2D-PIR of an example picture according to [Nabil *et al.*, 1996].

mapping function is used that is defined as $\varphi_r(d) = 1/d^r$, with d denoting the distance between two points (pixels) of an object. Dependant on the parameter r, different kinds of forces can be determined. For $r = 1$ for example, the attraction force between two points a_1 and a_2 is defined as $\varphi(a_1 - a_2)$. Thus, calculating the forces between all pairs of pixels of an object results in a force histogram describing the object. The matching between two objects is done by computing a similarity ratio of the two associated F-signatures. Although this kind of representation is characterized by low time complexity and invariance to scaling, translation, symmetry and rotation, it was developed for recognizing special kinds of graphical objects. Hence, this approach is rather suitable for characterizing technical drawings of the architectural domain by identifying objects such as a shower or a washbasin.

Another image-based approach was developed by Nabil et al., supporting the retrieval process by generating a representation called 2D Projection Interval Relationship (2D-PIR) [Nabil *et al.*, 1996]. Based on the 2D-string representation of Chang et al. [1987], a 2D-PIR is a symbolic representation of directional as well as topological relationships among spatial objects in a picture. In general, this concept adapts three existing representation formalisms, namely Allen's temporal intervals [Allen, 1983], 2D-strings, and topological relationships, and combines them in a novel way. As a result, a connected labeled graph is constructed, with nodes representing the objects of a picture in form of symbols (e.g. names) and edges illustrating the positional relationships between them (cf. figure 3(b)). Thereby, a positional relationship is described in form of a triple consisting of one topological and two interval relationships. While the topological relationship describes the correlation between the positions of two objects, the interval relationship constitutes a 'temporal' relationship between the objects. Therefore, an object is projected along the x and y axes resulting in an x-interval and an y-interval. On this basis, two objects are compared with regard to their 'temporal' appearance in the picture having, for example, a 'before', 'during', 'start', 'finish', or 'after' relationship. Figure 3 gives an example for a drawing with its corresponding graph consisting of three objects A, B, and C. The triple $(dt, d, >)$, for example, represents the 2D-PIR of the two objects B and C. The first parameter dt describes that the objects are 'disjoint'; the second parameter d illustrates that with regard to the objects' interval on the x-axis, object B appears 'during' object C. Finally, the third parameter $>$ specifies that the y-interval of object B lies 'after' the y-interval of object C. In this way, 2D-PIRs between all spatial objects in a picture are determined and a digraph is computed. The comparison of such two graphs consists in solving the graph isomorphism problem by applying similarity metrics both for topological as well as interval relationships.

Müller and Rigoll [1999] also present an approach for the description of image-based engineering drawings. Based on the use of stochastic models they represent a drawing image with a pseudo 2-D Hidden Markov Model (P2DHMM) which is surrounded by filler states. Thereby, a P2DHMM is defined as a stochastic automata with a two-dimensional arrangement of the states where states in horizontal direction are denoted as superstates. Moreover, each superstate is defined as a one-dimensional Hidden Markov Model in vertical direction. Since the generation of this kind of representation depends on a learning phase in which the P2DHMM is trained from specific graphic objects, this approach serves mainly for recognizing the learned objects in a drawing. As a consequence, applying this (pattern recognition) method for the retrieval of engineering drawings in real companies is not feasible because there is a huge amount of predefined objects that would have to be trained. For that reason, the concept of Müller and Rigoll is not further contemplated.

3.2 Vector-based approaches

Considering the pixel-based methods described in subsection 3.1 demonstrates that they are unsuitable for vector-based drawings because too much information gets lost or is not taken into account. Accordingly, retrieval approaches explicitly addressing the content of a technical drawing are needed.

The potential information content of a drawing as discussed in section 2 illustrates that searching for technical drawings should be based on both text and geometry / topology data. A method that tries to take into account both data types is described in [Love and Barton, 2004]. It is based on GT coding and was integrated in a commercial retrieval system called CADFind[1] supporting the CAD systems AutoCAD and SolidWorks. In this system, each drawing is represented automatically by a GT code. GT is the abbreviation for Group Technology and means that a part's geometry, material, and production process information is encoded into a string of digits or alphanumeric characters. Since an engineering drawing normally consists of several simple views of a part, the drawing is separated into these views at first. Afterwards, each view is extracted free of additions like title blocks, dimensions, or textual comments and serves as input to a feature extraction program. Given the resulting features, a GT code is generated. Since the authors did not explain their coding process in detail, a conventional coding and classification scheme such as the Opitz scheme [Opitz, 1970] can be used to form the basis of such a procedure. The Opitz classification scheme characterizes a part with respect to predefined properties (CAD features) of the part. For example, a part described by the partial code '01312' has the following geometrical characteristics:

- '0' = Rotational part with L/D $<$ 0.5 (first digit describes the component class)
- '1' = Stepped to one end, no shape elements (second digit specifies the external shape)

[1] www.sketchandsearch.com

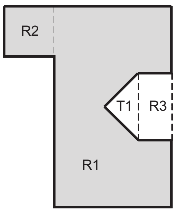


Figure 4: Example block according to [Park and Um, 1999].

- ’3’ = Smooth or stepped to one end with functional groove (third digit describes the internal shape)
- ’1’ = External planar surface (fourth digit gives information about plane surface machining)
- ’2’ = Axial holes related by a drilling pattern, no gear teeth (fifth digit specifies auxiliary holes and gear teeth).

Besides this geometrical description, the coding process includes also other additional information about the product such as material or production process information. Once the GT code is determined, both the code and an image of the view are stored in a record. Finally, two drawings (views) are supposed to be similar if they have similar GT codes and with it similar properties. However, implementing a retrieval system based on automatic coding and classification of drawings is no trivial task.

Except for this approach, normal text-based retrieval methods mainly provide no satisfying results. Consequently, the engagement in developing content-based retrieval concepts based on geometry and/or topology especially for technical drawings becomes a relatively new field of interest. For about ten years researchers from different countries have dealt with this subject.

Park and Um [1999] propose a method for content-based retrieval of technical drawings based on so-called dominant shapes. Here, the authors describe the contour of a complex graphic object by recursively decomposing its shape into dominant and auxiliary shapes. Moreover, they take into account topological information of the drawing by distinguishing between two types of spatial relationships: inclusion and adjacency. For this purpose, the authors remove dimension lines and characters from the drawing in the first step. Then, they partition the drawing into a set of dominant blocks, i.e. outstanding polygons formed along consecutive line segments describing, e.g. the views of a product. Furthermore, each block is separated into a set of shapes that can contain both polygons (blocks) and predefined primitives such as rectangles or circles. Thus, the description of a block results from adding or subtracting auxiliary shapes from a dominant shape. Figure 4 illustrates a simple example block consisting of a rectangle $R1$ (the dominant shape) extended by a rectangle $R2$ and reduced by the union of triangle $T1$ and rectangle $R3$. In this way, every arbitrary shape of a complex object can be described. Finally, the blocks are organized into a graph structure according to their inclusion and adjacency relationships to each other. The similarity between two drawings results from solving the graph matching problem. Furthermore, the recursive procedure applied in this approach enables partial matching of drawings, i.e. parts of a query drawing contained in other drawings can also be found.

Another approach, developed by Fonseca et al., draws on and expands the idea of Park and Um to improve the retrieval of technical drawings. In [Fonseca *et al.*, 2005], the authors perform two steps for the comparison of those artifacts. These steps consist in generating two representations which are used afterwards for similarity measurement. First of all, a representation based on topology information according to Park and Um’s method described in the previous paragraph is created. Hence, a drawing is partitioned into blocks by isolating polygons. These polygons are described by their spatial relationships to each other and are represented in form of a topology graph. Since graph matching as used in [Park and Um, 1999] is an NP-complete problem, Fonseca et al. use graph spectra instead to solve the matching problem. As a result, for each topology graph a descriptor is computed by determining the eigenvalues of the graph’s adjacency matrix. These values are stored in a multidimensional vector whose dimension depends on the complexity of the graph. Consequently, very complex drawings will result in vectors with high dimensions, while simple drawings will yield rather low dimensions. For that reason, the topological representations of the drawings are finally stored in an indexing structure called NB-tree which supports indexing of vectors with variable dimensions [Fonseca and Jorge, 2003]. By computing the euclidean norm a multidimensional vector is mapped to a 1D line and inserted in a B^+-tree. Moreover, a drawing is described on the basis of its contained geometry information. Fonseca et al. give two possibilities for extracting this data. On the one hand, a general shape recognition library called CALI, also developed by the authors, can be used. On the other hand, a computation of geometric features such as area and perimeter ratios from polygons such as the convex hull, the largest area triangle inscribed in the convex hull, or the smallest area enclosing rectangle amongst others is implementable. Applying the latter method to each polygon of a block gives a complete description of the block’s geometry. On the basis of these representations, the matching procedure proves to be as follows. Searching the k nearest neighbors that have similar topological descriptors works as a filtering step to narrow down the result set. Afterwards, the geometrical information is used to refine the remaining drawings. The advantage of this method arises from the fact that a multilevel searching approach is possible. Generating different topology graphs for various levels of detail of the drawing provides searching for complete drawings as well as for subparts of these. However, it has to be taken into account that using graph spectra does not ensure the uniqueness of topology descriptors. Thus, more than one graph can have the same spectrum and with it the same descriptor.

Extracting shape appearances for the retrieval of engineering drawings is also used by Liu et al. [2004]. This method represents a drawing by an attributed graph where a

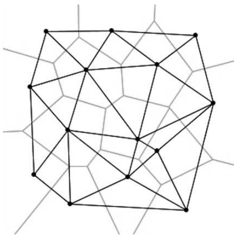


Figure 5: Voronoi diagram (gray) with corresponding Delaunay graph (black) (cf. http://de.wikipedia.org/wiki/Bild:Voronoi_delaunay.jpg, access: 2007-07-04).

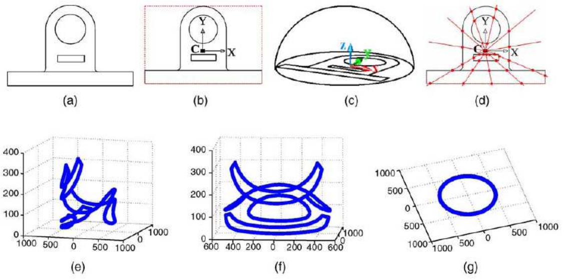


Figure 6: 2.5D spherical harmonics representation of a 2D drawing according to [Pu and Ramani, 2006].

node corresponds to a meaningful primitive extracted from the original drawing such as line or curve. Furthermore, for characterizing the content of an engineering drawing graph attributes are used which are divided into node attributes and edge attributes. While the former ones depict the appearance of the primitives such as circular, straight, or angular, the latter ones define the spatial relationships between these primitives such as parallel, or intersectant. The graph construction consists of four steps. First, each primitive is evenly sampled into multiple points which are adopted as input to a Delaunay tesselation in the second step. Thereby, a Delaunay graph is calculated which is defined as the dual graph of the Voronoi diagram of the set of points. Hence, after building the Voronoi diagram the graph can be constructed as follows: if two cells of the Voronoi diagram share an edge, the points located in the cells are connected. Figure 5 depicts both a Voronoi diagram (marked in gray) and the corresponding Delaunay graph (drawn in black). Afterwards, the resulting graph is simplified by merging nodes sampled from the same primitive into one node. Finally, as fourth step, the graph attributes are determined. By carrying out a Fourier transform with a direction histogram that describes the appearance of a primitive, the resulting coefficients of this transform are used as node attribute. On the other hand, the edge attribute contains elements such as the relative angle, the relative length, or the relative distance between two primitives. For graph matching the authors propose the application of the mean field theory which measures the similarity by calculating both the costs for matching graph edges and the costs for matching graph nodes.

Pu and Ramani deal with the problem of 2D drawing retrieval, too. In [Pu and Ramani, 2006], the authors propose two options, namely 2.5D spherical harmonics and 2D shape histogram, to find similar drawings for a query object as illustrated in figure 6(a). The first method draws on the successful application of the spherical harmonics representation in 3D shape matching. Thus, a drawing is described as a spherical function in terms of the amount of energy it contains at different frequencies. Therefore, the authors define a sphere whose center corresponds to the center of the drawing's bounding box and whose radius ensures to enclose the drawing completely (depicted in figure 6(b) and (c)). After that, a set of rays starting from the sphere center and locating in the plane where the 2D drawing lies, is generated (figure 6(d)). Determining the intersection points between these rays and the drawing serves as input to define a spherical function which is transformed from the 2D space into the 3D space. Finally, to compare two of these representations, a rotation-invariant descriptor is calculated by applying a fast spherical harmonics transformation method (figures 6(e)-(g) show the representation from different perspectives). The second approach of 2D shape histograms is a statistics-based representation originally developed by Osada et al. [2002]. For this purpose, a drawing composed of basic geometrical entities is transformed into a set of line segments. Afterwards, random points on these line segments are generated uniformly. The more points are sampled the more accurate is the approximation of the original shape. Once there are enough random points, the euclidean distance of every possible pair of randomly selected points is calculated. This distance is inserted into a histogram describing the distance distribution for the drawing. In the end, to measure the similarity between two histograms, the Minkowski distance is used.

Furthermore, technical line drawings can also be indexed by semantic networks. Yaner and Goel present in [2002] a retrieval process consisting of the two stages *reminding* and *selection*. In the first step, every drawing is represented by a feature vector, i.e. a vector of attribute-value pairs. Since drawings consist of different object types such as lines, circles, or ellipses, a feature vector is simply defined as a mapping from object type to its appearance frequency in the drawing. Consequently, a drawing is assumed to be similar to a query drawing if its feature vector is a superset of the query's feature vector. Given the results of this reminding step, the authors refine them by taking into account the spatial structure of the drawing. Therefore, the arrangement of the various object types in the drawing is described by five relation types called 'left-of', 'right-of', 'above', 'below', and 'contains'. To represent this spatial structure, the authors use a semantic network with nodes defining the spatial elements and links illustrating the spatial relations along them. On this basis, similarity between two drawings is determined in terms of subgraph isomorphism using symbolic methods; i.e. if the semantic network of the query can be found in the semantic network of a stored drawing, the latter one is delivered as similar.

4 Problems of implementing search techniques for DXF drawings

The concepts examined in section 3 prove that there are efforts to improve the retrieval of engineering drawings. However, implementing such methods for drawings especially based on the widely used vector file format DXF raises some problems that have to be solved. For that reason, the following paragraphs identify some of the problems a programmer is confronted with when processing data of a real technical drawing as illustrated in figure 7.

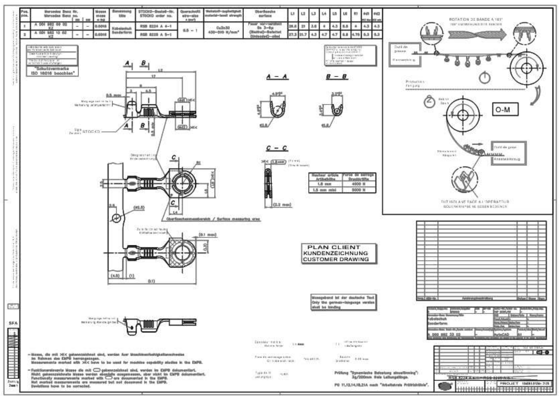
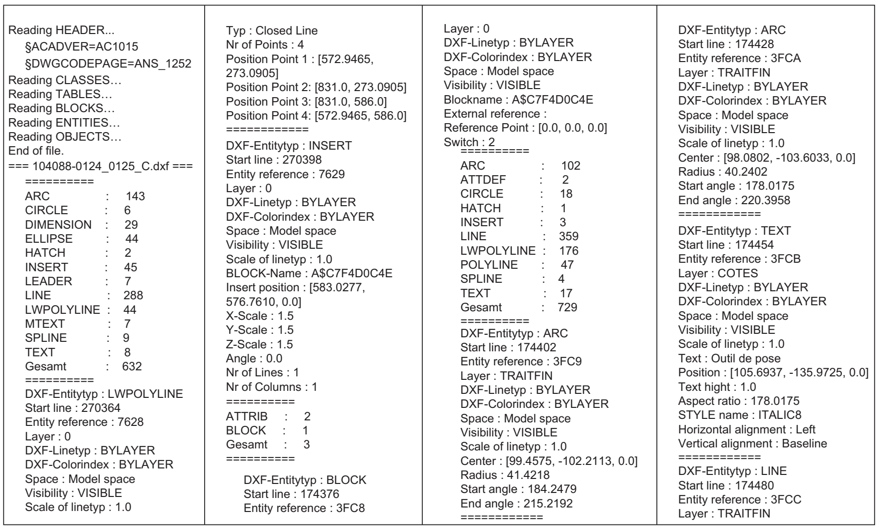


Figure 7: Example of a real technical drawing.



```
Reading HEADER...
    §ACADVER=AC1015
    §DWGCODEPAGE=ANS_1252
Reading CLASSES…
Reading TABLES…
Reading BLOCKS…
Reading ENTITIES…
Reading OBJECTS…
End of file.
=== 104088-0124_0125_C.dxf ===
    ==========
    ARC          :   143
    CIRCLE       :   6
    DIMENSION    :   29
    ELLIPSE      :   44
    HATCH        :   2
    INSERT       :   45
    LEADER       :   7
    LINE         :   288
    LWPOLYLINE   :   44
    MTEXT        :   7
    SPLINE       :   9
    TEXT         :   8
    Gesamt       :   632
    ==========
    DXF-Entitytyp : LWPOLYLINE
    Start line : 270364
    Entity reference : 7628
    Layer : 0
    DXF-Linetyp : BYLAYER
    DXF-Colorindex : BYLAYER
    Space : Model space
    Visibility : VISIBLE
    Scale of linetyp : 1.0

Typ : Closed Line
Nr of Points : 4
Position Point 1 : [572.9465,
273.0905]
Position Point 2: [831.0, 273.0905]
Position Point 3: [831.0, 586.0]
Position Point 4: [572.9465, 586.0]
============
DXF-Entitytyp : INSERT
Start line : 270398
Entity reference : 7629
Layer : 0
DXF-Linetyp : BYLAYER
DXF-Colorindex : BYLAYER
Space : Model space
Visibility : VISIBLE
Scale of linetyp : 1.0
BLOCK-Name : A$C7F4D0C4E
Insert position : [583.0277,
576.7610, 0.0]
X-Scale : 1.5
Y-Scale : 1.5
Z-Scale : 1.5
Angle : 0.0
Nr of Lines : 1
Nr of Columns : 1
==========
ATTRIB    :   2
BLOCK     :   1
Gesamt    :   3
==========
    DXF-Entitytyp : BLOCK
    Start line : 174376
    Entity reference : 3FC8

Layer : 0
DXF-Linetyp : BYLAYER
DXF-Colorindex : BYLAYER
Space : Model space
Visibility : VISIBLE
Blockname : A$C7F4D0C4E
External reference :
Reference Point : [0.0, 0.0, 0.0]
Switch : 2
    ==========
    ARC          :   102
    ATTDEF       :   2
    CIRCLE       :   18
    HATCH        :   1
    INSERT       :   3
    LINE         :   359
    LWPOLYLINE   :   176
    POLYLINE     :   47
    SPLINE       :   4
    TEXT         :   17
    Gesamt       :   729
    ==========
    DXF-Entitytyp : ARC
    Start line : 174402
    Entity reference : 3FC9
    Layer : TRAITFIN
    DXF-Linetyp : BYLAYER
    DXF-Colorindex : BYLAYER
    Space : Model space
    Visibility : VISIBLE
    Scale of linetyp : 1.0
    Center : [99.4575, -102.2113, 0.0]
    Radius : 41.4218
    Start angle : 184.2479
    End angle : 215.2192
    ============

DXF-Entitytyp : ARC
Start line : 174428
Entity reference : 3FCA
Layer : TRAITFIN
DXF-Linetyp : BYLAYER
DXF-Colorindex : BYLAYER
Space : Model space
Visibility : VISIBLE
Scale of linetyp : 1.0
Center : [98.0802, -103.6033, 0.0]
Radius : 40.2402
Start angle : 178.0175
End angle : 220.3958
============
DXF-Entitytyp : TEXT
Start line : 174454
Entity reference : 3FCB
Layer : COTES
DXF-Linetyp : BYLAYER
DXF-Colorindex : BYLAYER
Space : Model space
Visibility : VISIBLE
Scale of linetyp : 1.0
Text : Outil de pose
Position : [105.6937, -135.9725, 0.0]
Text hight : 1.0
Aspect ratio : 178.0175
STYLE name : ITALIC8
Horizontal alignment : Left
Vertical alignment : Baseline
============
DXF-Entitytyp : LINE
Start line : 174480
Entity reference : 3FCC
Layer : TRAITFIN
```

Figure 8: Part of the extracted content of the example drawing in figure 7.

This drawing contains a lot of useful information. On the one hand, the geometry of the product is shown from three views: one top view and two side views, depicted on the left side of the figure. In addition, slices according to the views are described, labeled as A-A, B-B, and C-C. Both views and slices are specified by dimensions. On the other hand, the drawing includes an amount of text information in form of text fields such as product designation, part number(s), material, project information or vendor data. Moreover, the right side of the drawing illustrates manufacturing information.

4.1 Missing file structure

Due to the mentioned information content above, the retrieval of technical drawings based on metadata is necessary. Thus, a designer should be able to search for a drawing by querying, for example, the part number or the project in which the drawing was created. Therefore, the available metadata has to be extracted. One main problem in doing so is the missing structure of a DXF file, which makes the extraction process quite difficult to accomplish. Since DXF was developed mainly as output format for plotters and as communication medium between designers and manufacturers, it is only suited for presentation. In correspondence to [Rudolph, 2000], examining a DXF drawing in more detail shows that it is a container of arbitrary objects with no interrelationships. In general, the objects are divided into two groups. The first group contains objects without any graphical embodiment such as dimensions, layers, line types, text types, or viewports. However, the second group, also referred to as entities, comprises objects with a graphical embodiment such as lines, circles, ellipses, polylines, splines, or blocks. The latter one combines a set of arbitrary objects into one object which can be used several times in a drawing. According to figure 8, a DXF file stores these objects in an arbitrary order, describing them mainly by their object type, their position on the drawing, and their geometrical data.

To extract metadata from a drawing, objects of type TEXT have to be identified. Although every textual information is stored in such an object, the understanding of its semantics is not ensured. Studying a text field on a drawing shows that it is composed of an abundance of LINE and TEXT objects. While the LINE objects possess no relevant information for metadata-based retrieval, we have to select all TEXT objects of a DXF file from which the textual data can be extracted. However, this procedure delivers the text, but not its meaning. Consequently, a part number, e.g. ’A2A00476’, can be extracted from the associated TEXT object. But there is no information that this text defines the part number. Instead, this information (the meaning) is also stored as text ’Part Number’ in a separate TEXT object which has no relationship to the number’s TEXT object. As a result, metadata extraction is possible, but taking into account the semantics of the extracted data is not supported. A possible solution for this problem could be the inclusion of the spatial proximity of the objects. Since objects that belong together, like ’Part Number’ and ’A2A00476’, are normally positioned close to each other on a drawing, finding adjacent TEXT objects (nearest neighbors) could help to find the matching ones. However, it has to be taken into account that there are often more than two objects in short distance to each other in a text field. Thus, identifying the right ones is a further challenge.

Furthermore, most of the concepts described in section 3 act on simple assumptions by using drawings consisting of only one block (a block defines a view of the product). Thus, these methods deal with technical drawings not comparable to real practical DXF drawings which contain normally more than one view. For example, Love and Barton define in [2004] a drawing as a simple view of a part that defines its essential geometry without any of the additions such as title blocks, dimensions, or textual comments, which are necessarily present on a normal engineering drawing. Hence, the different views of a real technical drawing have to be identified and separated. In doing this, the same problem of missing structure occurs. Although DXF provides so-called LAYER or VIEWPORT objects for defining several views of a product, this option is often not used by designers. All information – especially the view’s geometrical data – is mostly stored together in one LAYER object, hindering the identification of views. Moreover, this LAYER object contains the entities in an arbitrary order, i.e. the programmer has to find out which objects belong to which view. For example, a line can either be part of the geometry description of a view, or a dimensioning line, or it can be a boundary line of a text field. While in the first case the line is important for finding similar products based on geometry, a dimensioning line and a boundary line can be neglected for both a metadata and a geometry-based search. Hence, applying an appropriate algorithm for segmenting the drawing is needed which identifies the different views of the drawing together with their associated objects. Considering positional information can also be helpful for this task. However, identifying views with associated objects is a necessary step to be able to handle real technical drawings.

4.2 Different drawing layouts and format versions

Once the different views of a drawing are identified and separated, a further problem has to be solved. In general, the process of retrieving artifacts is based on posing a query. This can be done by either using a text describing the information need or by using an example object/document as input. The latter query-by-example option leads to the fact that all views of the drawings have to be compared. Thus, the occurring problem is that views used for a drawing are chosen variably from user to user, i.e. there is no standardization for views on a drawing. While one designer represents a product, for example, with a top view and two side views as in figure 7, another designer displays the same product with a front view, only one side view, and a back view. This leads to the fact that it is not ensured that all drawings have the same views. Thus, the emerging question is what views should be compared or, rather, how significant is the similarity measured between two views for the whole document. Consequently, retrieving technical drawings requires a form of indexing, and with it a filtering step that takes into account the kind of views contained in the drawing.

Another problem in handling technical DXF drawings arises from the fact that designers configure their drawings in different ways. Although the data which has to be on a drawing is predefined in most cases, there is no unique layout for the drawings. In general, the ISO norm 5457 [ISO, 1999] defines the sizes and the general layout of technical drawings, but, dependent on suppliers and customers, a company often has to adapt the layout to the specific guidelines of the supplier or customer. As a consequence, a company has to deal with a multitude of different drawing layouts. Besides, every designer has its own idea of presentation (i.e. there is no uniform use and notation of LAYER objects and no uniform placing of text blocks) what complicates working with this kind of CAD-specific artifact.

Finally, DXF is a format that is in constant development, i.e. with every new version of the CAD system AutoCAD Autodesk also provides an improved DXF version. Consequently, the retrieval of technical drawings has to take into account different format versions, and with it different kinds of objects dependent on the used version. While the older objects are well documented, the new ones are not. Thus, considering all possible objects of a DXF format is not feasible. For this reason, a restriction to the core objects according to [Rudolph, 2000] makes sense. Concentrating on objects which are of capital importance and which are most commonly used, such as lines or circles, is a first possible solution for this problem. Nonetheless, a lot of information gets lost by disregarding the new objects. This can lead to the fact that products are not described completely and a retrieval system might generate false results.

5 Feature Extraction Process

Solving the problems described in section 4 requires a feature extraction process consisting of two parallel paths. Consequently, we propose the procedure depicted in figure 9 to generate both a metadata and a geometry representation for a DXF drawing. One path of this process consists in extracting the metadata of a technical DXF drawing. For this, a *Metadata Extractor* is needed which selects all the TEXT objects from the DXF drawing. Afterwards, relationships between the TEXT objects have to be identified. This task is carried out by a *Metadata Correlator* which has to find metadata that belong together such as ’Designer’ and ’John Q. Public’. Therefore, the *Metadata Correlator* has to implement methods that consider the spatial proximity of textual objects. However, it is also conceivable to include concepts from the domain of Optical Character Recognition (OCR). These methods enable the identification of single text objects and give the possibility of identifying their relationships and semantics. Since there are OCR approaches that take into account the context of a text object, a differentiation between typical text attributes, which are contained in every drawing (e.g. drawing number, creation date, or the designer’s name), and other additional text information can be conducted.

The other path of the feature extraction process generates a geometry representation for a drawing. First, a *Layout Eliminator* identifies and eliminates all the elements that determine the layout of the drawing. Moreover, all dimensions contained in the drawing have to be rejected. To this end, a *Dimension Eliminator* is used. After these two operations, the drawing only contains the real geometrical information in form of product views and slices. Hence, the different views of the drawing have to be determined. This function is realized by a *Drawing Decomposer* which applies a segmentation method to partition the drawing.

Finally, the *Representation Generator* has to transform the extracted information (metadata and geometry) into suitable representations (e.g. feature vectors) and has to store them in index structures.

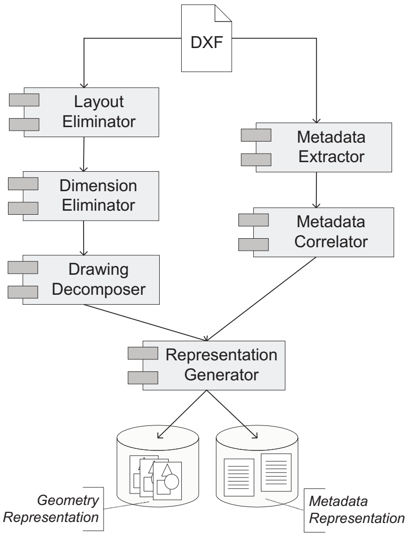


Figure 9: Proposed feature extraction process for DXF drawings.

6 Conclusion and Future Work

Retrieving technical drawings is an eminent help for engineering designers in doing their everyday work. Thus, a retrieval algorithm has to be implemented which considers both the textual information in form of metadata, and the geometrical data describing the geometry of the illustrated product. Since text-based retrieval methods are widely explored in research, our paper presents two groups of geometry / topology-based concepts: pixel-based and vector-based approaches. While the former do not take into account the real information content of a drawing and technical drawings are actually generated in vector formats, the latter group of similarity methods should be used for a retrieval system. However, there is a diversity of proprietary file formats (one for each available CAD system) that cannot be handled without having expensive converters. Since Autodesk's DXF format is the only open format in this domain, we decided to base our ideas for retrieving technical drawings on this file format. But using DXF implicates several problems as demonstrated in this paper. Thus, implementing a search functionality for technical drawings is no trivial task.

Therefore, we propose a procedure for extracting both metadata and geometry information from a DXF drawing. We started our implementation of this process by extracting the textual information of a drawing. We created a *Metadata Extractor* that simply selects all TEXT objects from the drawing. The results are indexed by a *Representation Generator* using an Apache Lucene[2] index. Hence, our future work has to concentrate on building the connections between the metadata and its meaning to improve the index. In addition to this task, we also dealt with content-based retrieval methods for DXF drawings. Concerning this matter, we implemented two algorithms from section 3. Currently, these algorithms work on simple drawings containing only one view of a product and no additional information. Thus, in our next working steps we have to evaluate the algorithms in regard to their suitability for practical use. Therefore, a concentration on identifying the different views of a real technical drawing by eliminating both layout elements and dimensions is necessary.

Acknowledgments

The work described in this paper is part of the Bavarian research alliance FORFLOW consisting of six Bavarian universities working in the fields of engineering design and computer science. It collaborates with 21 companies and is promoted by the Bavarian Research Foundation (BFS). [3]

References

[Allen, 1983] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[Chang *et al.*, 1987] S. K. Chang, Q. Y. Shi, and C. W. Yam. Iconic Indexing of 2D Strings. *IEEE Transactions of Pattern Analysis and Machinable Intelligence*, 9(3):413–428, 1987.

[Conrad, 2005] Klaus-Jörg Conrad. *Grundlagen der Konstruktionslehre: Methoden und Beispiele für den Maschinenbau*. 3. aktualisierte und erweiterte Auflage, Carl Hanser Verlag, München/Wien, 2005.

[DIN, 1962] Deutsches Institut für Normung. *DIN 199: Technische Zeichnungen - Benennungen*

[Eigner and Maier, 1991] M. Eigner and H. Maier. Einstieg in CAD: Lehrbuch für CAD-Anwender. 8. Nachdruck, Carl Hanser Verlag, München/Wien, 1991.

[Fonseca and Jorge, 2003] Manuel J. Fonseca and Joaquim A. Jorge. Towards Content-Based Retrieval of Technical Drawings through High-Dimensional Indexing. *Computers and Graphics*, 27(1):61–69(9), February 2003.

[Fonseca *et al.*, 2005] Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge. Content-Based Retrieval of Technical Drawings. *Int. Journal of Computer Applications in Technology*, 23(2-3):86–100, March 2005.

[Fränti *et al.*, 2000] Pasi Fränti, Alexey Mednonogov, Ville Kyrki, and Heikki Kälviäinen. Content-based matching of line-drawing images using the Hough transform. *Int. Journal on Document Analysis and Recognition*, 3(3):117–124, 2000.

[ISO, 1999] International Organization for Standardization. ISO 5457:1999, Technical product documentation - Sizes and layout of drawing sheets. 1999.

[Liu *et al.*, 2004] R. Liu, T. Baba, and D. Masumoto. Attributed Graph Matching Based Engineering Drawings Retrieval. In *Document Analysis Systems VI - Proc. of the 6th Int. Workshop*, pages 378–388, Florence, Italy, September 2004. Springer.

[Love and Barton, 2004] Doug Love and Jeff Barton. Aspects of design retrieval performance using automatic GT coding of 2D Engineering Drawings. *4th Int. Conf. on Integrated Design and Manufacture in Mechanical Engineering*, Bath, April 2004.

[Müller and Rigoll, 1999] Stefan Müller and Gerhard Rigoll. Searching an Engineering Drawing Database for User-specified Shapes. In *Proc. of the Fifth Int. Conf. on Document Analysis and Recognition*, pages 697–700, Washington, DC, USA, 1999. IEEE Computer Society.

[Nabil *et al.*, 1996] Mohammad Nabil, Anne H. H. Ngu, and John Shepherd. Picture Similarity Retrieval Using 2D Projection Interval Representation. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):533–539, August 1996.

[Opitz, 1970] Herwart Opitz. A Classification System to Describe Workpieces. Pergamon Press, Oxford, 1970.

[Osada *et al.*, 2002] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape Distributions. *ACM Transactions on Graphics*, 21(4):807–832, October 2002.

[Park and Um, 1999] Jong H. Park and Bong S. Um. A new approach to similarity retrieval of 2-D graphic objects based on dominant shapes. *Pattern Recognition*, 20:591–616, 1999.

[Pu and Ramani, 2006] Jiantao Pu and Karthik Ramani. On visual similarity based 2D drawing retrieval. *Computer-Aided Design*, 38(3):249–259, March 2006.

[Rudolph, 2000] Dietmar Rudolph. AutoCAD-Objekte. SYBEX-Verlag GmbH, Düsseldorf, 2000.

[Tabbone *et al.*, 2001] S. Tabbone, L. Wendling, and K. Tombre. Indexing of Technical Line Drawings Based on F-Signatures. In *Proc. of the Sixth Int. Conf. on Document Analysis and Recognition*, pages 1220–1224, Los Alamitos, CA, USA, 2001. IEEE Computer Society.

[Vajna *et al.*, 1994] S. Vajna, Chr. Weber, J. Schlingensiepen, and D. Schlottmann. CAD/CAM für Ingenieure: Hardware - Software - Strategien. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1994.

[Yaner and Goel, 2002] Patrick W. Yaner and Ashok K. Goel. Using Spatial Structure in the Associative Retrieval of 2-D Line Drawings. Technical Report GIT-CC-02-70, Artificial Intelligence Laboratory College of Computing, Georgia Institute of Technology, Atlanta, USA, December 2002.

[2] http://lucene.apache.org

[3] For more details see http://www.abayfor.de/forflow/en

Modelling a Summarisation Logic in Probabilistic Datalog

Jan Frederik Forst, Thomas Roelleke, Anastasios Tombros
Queen Mary, University of London
London E1 4NS, United Kingdom
{frederik, thor, tassos}@dcs.qmul.ac.uk

Abstract

The automatic summarisation of data is an important information retrieval task, since summaries help users to efficiently access information and judge their relevance. For enabling the flexible and customisable generation of summaries, we propose a new abstraction layer: a summarisation logic. In this paper, we introduce and investigate a Datalog-based approach for the processing of POLIS (Probabilistic Object-oriented Logic for Information Summarisation). The main finding of this paper is that POLIS expressions can be translated into probabilistic Datalog, thus allowing the evaluation of POLIS to take advantage of existing probabilistic Datalog engines. We report on the retrieval quality and efficiency of POLIS and its Datalog-based implementation, observed for experiments carried out with the DUC (Document Understanding Conference) collection.

1 Introduction

Research in information retrieval is motivated by the need for methods which allow users to find information relevant to their information needs. The importance of efficient automatic document summarisation approaches as a tool for handling ever larger amounts of data was recognised as early as 1960 [Luhn, 1958; Edmundson, 1969].

Document summaries can be either *extracts*, consisting of material verbatim present in the original document, or *abstracts*, coherent documents generated from –but not present verbatim in– the source document. The generation of abstracts is usually a postprocessing step following the extraction of "extract worthy" material. Over the years, several methods where proposed for extracting the most salient information from documents: Sentence scoring strategies try to calculate a "score" for each document sentence, based on term- and sentence-frequencies in documents; heuristic sentence scoring strategies, incorporating information about the relative position of sentences within paragraphs; scores for "bonus" and "malus" words and information about sentence length. Natural Language Processing approaches try to determine the informativeness of bits of documents based on the syntactical structure of sentences, and individual user background. Machine learning approaches attempt to determine the most relevant parts of documents by learning from given examples – documents with their associated, human generated abstracts or extracts.

Furthermore, probabilistic summarisation models based on the above strategies were developed, and some of those models were implemented in standalone summarisers, which can be included in other information retrieval systems (e.g. SUMMARIST [Hovy and Lin, 1997]). However, both (i.e. mathematical models and complete summarisation systems) represent extremes of approaches to document summarisation: probabilistic models define how a summary should be constructed, but need to be implemented. Summarisers, on the other hand, represent a "black box" approach to summarisation, where user interaction and control is minimal.

In this paper, we present a new development in the field of document summarisation, which tries to integrate both extremes: a summarisation logic, which allows users to control the summarisation of (structured) documents, but hides details of how the summarisation approach is implemented.

A native interpreter for this logic has not been implemented yet. Instead, expressions in our logic are translated into probabilistic Datalog expressions, which can be interpreted by existing IR frameworks. In addition, the generated Datalog program itself is meaningful to experts in knowledge engineering; it thus allows for inspection of the summary generation process. In this paper, we show how summarisation expressions of a summarisation logic are translated into equivalent probabilistic Datalog expressions.

2 Background

Early work on summarisation tried to establish the importance of sentences using heuristic features, such as term frequencies in sentences, the location of sentences in the document, and sentence length. Overall sentence scores were derived by a linear combination of individual features [Luhn, 1958; Edmundson, 1969]. In later work, this combination of features was optimised using machine learning systems, that would derive ideal weighted combinations of features from a given knowledge base, consisting of full texts, and human generated extracts [Kupiec *et al.*, 1995].

Later research shifted the focus from term weighting strategies based on linear combinations of features to probabilistic sentence scoring strategies. For example, Saravanan *et al.* [Saravanan *et al.*, 2005] present a multi-document summariser based on a K-mixture model for term distribution. Knight and Marcu [Knight and Marcu, 2002] present a probabilistic model for sentence compression, which goes beyond conventional probabilistic models for sentence extraction trained on given document / summary pairs.

While probabilistic summarisation models provide a high level of detail to users, detailing the exact process of sentence weighting, extraction and content generation, they need to be implemented, and require users to control every aspect of a summarisation system, from document parsing and representation, to the actual summary generation.

The other extreme are summarisation systems, such as SUMMARIST [Hovy and Lin, 1997], which implement complex summarisation models that take into account several sentence scoring strategies to detect the most salient parts of documents. The output of such summarisation systems is either an extract of the input document(s), or some conceptual representation for the later generation of summaries (e.g. SUSY [Fum *et al.*, 1985], TOPIC [Reimer and Hahn, 1988], SCISORS [Rau *et al.*, 1989]). Although such systems try to combine the best summarisation strategies available at any one time, they represent a black-box approach to summarisation, as the inner workings might not be accessible to a user, and the process of summary generation is fixed and cannot be tailored towards user needs.

Summarisation of structured documents adds another level of complexity to the summarisation task, as the explicit structure provides additional information. Recent approaches to structured document summarisation (here: XML summarisation) have mainly taken a data-centric point of view, where summarisation is seen as a form of data compression. Alam *et al.* [Alam *et al.*, 2003] present a summarisation approach in which the original structure of the original document is retained. Litkowski tested a summarisation system for structured documents at the Document Understanding Conference (DUC) (e.g. [Litkowski, 2004]). However, neither of these systems makes it explicitly clear *how* sentences are scored, and how structure is used in the summarisation process.

The summarisation logic discussed here provides an abstraction layer to the task of document summarisation, and implements the summarisation strategy in probabilistic Datalog. Probabilistic Datalog is the probabilistic extension of stratified Datalog with negation [Rölleke and Fuhr, 1998; Fuhr, 2000], and thus combines the rule based modelling of Datalog with a probabilistic framework necessary for implementing IR models.

3 Summarisation

In this section we will highlight how a summarisation logic would benefit users as an abstraction layer to the task of document summarisation. We will sketch a summarisation strategy in Datalog, to show where problems and complications arise. The modelling in Datalog will then be compared to an equivalent expression in the summarisation logic used for the present discussion, POLIS, which is implemented in probabilistic Datalog, but hides implementational details using a simplifying syntax.

3.1 Summarisation in Datalog

Suppose a user would like to generate document summaries by extracting individual sentences from the source documents. To provide some level of abstraction, and to not burden users with the task of implementing summarisation models, the summarisation strategy should be implemented in a functional language, such as Datalog. An initial idea for a Datalog expression to this effect might look like

```
summary(S)   :-   sentence(S,D);
```

where *S* denotes a sentence, and *D* is a document. The above expression makes the implicit assumption that a predicate "Sentence" exists, which provides information on the logical markup of documents. This would usually not be the case. For the above statement to be processed correctly, it would thus be necessary for a user to also provide a definition of "Sentence". If we assume that a sentence is part of a document, and that every sentence is known to be an instance of *type* sentence, this could be expressed as

```
sentence(S,D)   :-   part_of(S,D),
                     instance_of(S,'sentence');
```

Starting to formulate a summarisation strategy in this fashion, it becomes evident that a great deal of complexity emerges fairly quickly. Such an approach would also require users to be familiar with Datalog. Furthermore, the above program would not provide a ranking of sentences, so it would not be possible to easily distinguish sentences representative of a document from non-representative sentences. Finally, Datalog does not provide an operator to allow numbered access to predicates, such as an operator to access to first ten instances of a predicate (some kind of a *top-K* operator). This, combined with a ranking of sentences, however, would be necessary to generate summaries of documents by extracting the *n* most representative sentences.

3.2 Summarisation Logic

The above example of a summarisation strategy in Datalog highlights some of the main problems of any such approach: standard Datalog does not provide the measures necessary for the ranking of textual elements, and furthermore, users need to be familiar with the implementation layer to express their strategies. Additionally, the implementation burdens users with a level of complexity not immediate to their need for document summaries. The main motivation for developing a summarisation logic thus is the flexibility with which it provides users for developing their own summarisation systems, by adding a layer of abstraction to hide implementational details.

As an initial example, we use a system for the task of expert profiling. The purpose of such profiles is to provide details of an expert candidate's expertise. Profiles in this context are summaries of all documents associated with an expert candidate. The profiling tasks thus resembles a multi-document summarisation task. A summarisation logic here provides users with a way to flexibly customise profiles to their needs [Forst *et al.*, 2006].

Summaries can also be used beneficially in retrieval systems. User studies conducted by Wolf et al. show that more sophisticated summaries of documents retrieved in response to a user query reduce the time it takes for participants to complete given tasks [Wolf *et al.*, 2004]. Wolf et al. also note that their "work demonstrates that using information about the subcomponent structure of documents to guide selective extraction can result in more useful document summaries" [Wolf *et al.*, 2004]. The summarisation logic used for the present discussion was specifically developed for the summarisation of structured documents, and allows the explicit selection of the granularity at which document components should be extracted for summarisation by use of logical expressions. It is thus possible to not only summarise documents in response to a user query, but to also incorporate user preference in the summarisation model.

As an example, consider a retrieval system, where each returned document is represented by up to three most important sentences occurring in it. An initial idea on how to allow a user to express this could look like **/sentence:<=3**. However, this expression adds an additional layer of abstraction, by not only hiding the internal of the summarisation process itself, but additionally simplifying the internal structure of documents to be summarised. Processing this instruction would require a fair amount of insight into the structure of documents, which could potentially be achieved through the use of schema mappings from user input to the actual structure of documents. A user path-expression like **/sentence** would be mapped probabilistically onto different document path-expressions (such as **/sec/sentence**, **/sec/para/sentence**, or **/sec/para/sent**), where a distance criterion – e.g. edit-distance, or node-distance – between user expression and actual document structure could be utilised as a weighting function. However, this syntactic feature has not been implemented yet. Instead, the actual logical expression to carry out this operation is: **/*/body/sec/para/sent:{:3}**, for a document that has sentences as parts of paragraphs, which are children of section elements inside 'body' elements.

The remainder of this section provides an overview over the summarisation logic which provides the basis for the present discussion. Emphasis is placed on only those aspects which affect the modelling in probabilistic Datalog, i.e. the underlying probabilistic model and the syntax. Other aspects of the logic, such as its semantics and an evaluation of its efficiency, have been presented elsewhere and will be covered only briefly.

3.3 Syntax

The summarisation logic used for the present discussion, POLIS, employs an XPath-like syntax. XPath allows the easy selection of parts of structured documents. It also reflects the structure of documents in such a way that non-experts of information retrieval can develop an understanding of the meaning of the expressions. To allow for summarisation, additional constructs were added, to allow for restrictions on the number of elements selected, and to include weighted query terms.

A POLIS-expression starts with a root-definition, followed by a specification of the axis of the structured document to be summarised.

```
expression    ::=  | root-element axis-def
root-elem.    ::=  | '/'
axis-def      ::=  | axis-element
                   | summary-definition
axis-elem.    ::=  | node-def. sum.-def.
                   | node-def '/' axis-def
node-def      ::=  | '*'
                   | '*' '{' node-restr. '}'
                   | NAME
                   | NAME '{' node-restr. '}'
node-restr.   ::=  | context-ident.
                   | q.-term
context-id.   ::=  NUMBER
q.-term       ::=  | NAME | NAME q.-term
                   | w. NAME | w. NAME q.-term
```

An axis-definition can either be a path-element (called *axis-element* here), or a summary-definition, restricting the number of elements to appear in the summary. Nodes in the axis can be specified explicitly by giving the number of the axis-element to be used, or by specifying query-terms to appear in the node-element.

```
summary-def.     ::=  ':' '{' sum.-res. '}'
summary-restr.   ::=  | range-restriction
                      | element-restriction
                      | vague-range-restr.
range-restr.     ::=  ':' NUMBER
element-restr.   ::=  | NUMBER
                      | NUMBER ',' elem.-res.
vague-range-r.   ::=  | 'full'
                      | 'large'
                      | 'medium'
                      | 'small'
                      | 'tiny'
```

The important elements forming the summary can be range-restricted (i.e. allowing a certain number of elements to be returned), element-restricted (i.e. asking for specific elements to be returned) or vaguely range restricted (i.e. not giving an explicit range, but using vague predicates for the range). Vague predicates allow to associate the number of elements used in a summary with a certain predicate. This association could be user-specified, or could be learned from user-provided examples.

NAME	::=	$[a-z][a-z\_A-Z0-9]^*$
NUMBER	::=	$[0-9]^+$

Finally, a name is defined as any character sequence starting with a lower case letter, followed by any number of letters, digits or underscores. A number is a character sequence consisting of a digit, followed by any number of digits.

4 Probabilistic Summarisation Model

The probabilistic model implemented by POLIS is motivated by earlier sentence scoring strategies outlined in the introduction. The task of any summarisation system is to extract the most salient textual elements from either a single document, or a collection of documents, such that the extracted elements are as representative as possible of their origin. We will refer to this property as the "aboutness" of textual elements; any textual element that participates in the formation of a summary should be as much "about" its source document(s) as possible. It follows that the probability of any (random) text element being included in a summary should be proportional to the degree that that textual element is "about" its surrounding document or context.

With a view on general research in IR, the aboutness of textual elements to their surrounding contexts is very similar to the idea that documents are "about" queries. In traditional IR, the relevance estimation between a document d and a query q can be seen as "the extent to which q might be inferred from d" [van Rijsbergen, 1986], expressed as

$d \rightarrow q$, where $\rightarrow$ is not the material implication $\supset$ (defined as $d \supset q$ = $\neg q \vee d$), but rather has a probabilistic interpretation such that the correspondence (or *aboutness*) of d and q is expressed as the degree to which $d \rightarrow q$ is true.
Thus,

$$\text{aboutness}(d, q) := P(d \rightarrow q).$$

Following van Rijsbergen, the probability P($d \rightarrow q$) can be defined as the conditional probability P($q|d$), which can be rewritten such that

$$P(q|d) := \frac{P(d \cap q)}{P(d)}.$$

Assuming term independence, the joint probability P($d \cap q$) and the document probability P(d) can be calculated as the total probability over a set of disjoint terms, with

$$P(d \cap q) = \sum_t P(d \cap q|t)P(t)$$

and

$$P(d) = \sum_t P(d|t)P(t).$$

Using these definitions, it is possible to rewrite P($d \rightarrow q$) as

$$\sum_t P(t|d)P(q|t).$$

In terms of traditional IR models, P($t|d$) can be interpreted as the term-frequency *tf*, while P($q|t$) is a measure of term specificity, given by the collection frequency weight (CFW), or inverse document frequency *idf*.

The sentence weighting model implemented by POLIS directly builds upon this IR model. Terms in textual elements occurring at the granularity specified by a user are weighted using the *idf* collection weight, while terms in the surrounding document or context are weighted using the document frequency *tf*. The sentence weighting model thus implements a measure of "aboutness" in terms of the implication from subcontexts to supercontexts. The weighting of terms in both the user specified elements and the overall contexts is additionally influenced by terms occurring in their sub-components.

5 Modelling POLIS in Datalog

This section will provide details how the summarisation model outlined in 4 can be expressed as a Datalog program, consisting of a series of Datalog expressions. These expressions can be broadly classified as belonging to one of two functional categories: expressions for extracting textual elements at the user specified level of granularity, and expressions for implementing the probabilistic model (i.e. expressions for *weighting* predicates / tuples). The last processing step in any summarisation model is the combination of the weighted predicates, to produce an actual summary. Figure 1 shows how the different functional units stack up to model the summarisation function.

5.1 User-specified Extract Granularity

In section 3.2, we mentioned that POLIS allows the specification of the granularity at which textual elements should be extracted for the generation of summaries. To model the selection of elements in probabilistic Datalog, the system relies on a relational schema known as the object-oriented content representation (OOCR), or *object-relational* schema. This will be illustrated by a short example. We will again use the user request shown in 3.2: **/*/body/sec/para/sent:**{**:3**}. Figure 2 shows a short document which follows the schema outlined in this expression. The text is a short haiku (attributed to Shiki), which for the purpose of the discussion was transformed into an XML document with a *body*, which consists of sections (*sec*) with paragraphs (*para*), which in turn hold sentences (*sent*).

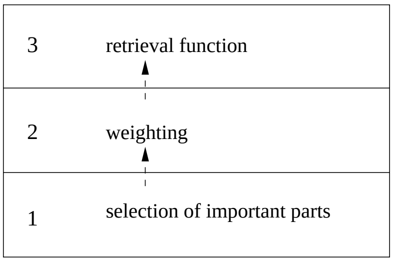
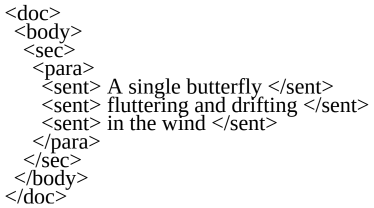


Figure 1: Layers implementing the summarisation function



```
<doc>
 <body>
  <sec>
   <para>
    <sent> A single butterfly </sent>
    <sent> fluttering and drifting </sent>
    <sent> in the wind </sent>
   </para>
  </sec>
 </body>
</doc>
```

Figure 2: Sample document

This document can be entirely represented by four relations: *instance_of* which stores information about the type of tags, *part_of* which provides details about the structure of the document, *attribute* which holds information about tags such as identifiers, and *term*, to represent the actual content of the document. Figure 3 shows how the sample document is represented.

POLIS steps sequentially through the given logical expression to define an intensional relation that contains all those elements at the user specified granularity. The first element in the expression is *"/*"*, denoting any type of element that is a part of the root. POLIS creates an intentional predicate (named after the type of element, or *p* (for *predicate*) if none is provided, followed by a sequential number) which holds all those elements:

```
p0(A)    :-    instance_of(A,B,"/");
```

Note that the selection is performed via the *instance_of* here, while it would have been possible to carry out the selection equally well on the *part_of*. This is for consistency reasons, as expressions involving the type of tags would need to be translated using the *instance_of*.

Expanding the POLIS expression further, one arrives at *"/*/body"*. Here, the type of element is specified (*body*), and all relevant elements need to be part of those elements selected for the first part of the expression. Accordingly, the Datalog expression is:

```
body1(B)    :-    p0(A), part_of(B,A),
                    instance_of(B, "body", X);
```

attribute

type	object	value	context
id	/doc[1]	1	/
id	/doc[1]/sec[1]	1.1	/doc[1]
⋮	⋮	⋮	⋮
runningID	/doc[1]	1	/
runningID	/doc[1]/sec[1]	1	/doc[1]

part_of

subcontext	supercontext
/doc[1]	/
/doc[1]/body[1]	/doc[1]
/doc[1]/body[1]/sec[1]	/doc[1]/body[1]
/doc[1]/body[1]/sec[1]/para[1]	/doc[1]/[...]/sec[1]
⋮	⋮

instance_of

object	class	context
/doc[1]	doc	/
/doc[1]/body[1]	body	/doc[1]
/doc[1]/body[1]/sec[1]	sec	/doc[1]
⋮	⋮	⋮

term

term	context
single	/doc[1]/body[1]/[...]/sent[1]
butterfly	/doc[1]/body[1]/[...]/sent[1]
fluttering	/doc[1]/body[1]/[...]/sent[1]
⋮	.

Figure 3: Relational representation of document

The *"X"* in the above expression represents "any value", as the value of the attribute is irrelevant for the operation, but a variable still needs to be given for a correct schema mapping. The syntax definition highlights how it is also possible to specify query terms in certain tags. For example, suppose that the POLIS expressions would have been **/*/body{"butterfly"}/sec/para/sent:{:3}**, the Datalog expression for predicate *body1* would have been:

```
body1(B)   :-   p0(A), part_of(B),
                  instance_of(B, "body", X),
                    term("butterfly", B);
```

Here, queryterms are interpreted such that they have to occur strictly in those elements specified in the logical expression. It would also be possible to allow for terms to occur in an augmented context, rather than a strict context, by replacing *term* in the above expression with *about* (see subsection 5.2). The steps outlined above are repeated for all path-elements of the logical expression. We will only show it here for the last level, which is the user-specified level of granularity. Similarly to the above examples, the expression for selecting elements at the right granularity is:

```
sent4(H)   :-   para3(G), part_of(H,G),
                  instance_of(G, "sent", X);
```

Elements at this user-specified level will also be called *important parts (impParts)*, which can be expressed as

```
impParts(I)   :-   sent4(I);
```

These important parts will later be used in the formulation of the retrieval strategy.

5.2 Knowledge Augmentation

In subsection 4, we noted that the weighting of terms in contexts is influenced by the presence of terms in those contexts' subparts. More generally, the *knowledge* of contexts is *augmented* with the knowledge of subcontexts. Traditionally, this kind of knowledge augmentation is modelled in Datalog via a "link" relation, such that a certain context is "about" a term if either that term occurs in the context, or it occurs in a context linked to the current one:

```
about(T,D)   :-   term(T,D);
about(T,D)   :-   link(D,D2), term(T,D2);
```

We follow this idea of modelling context augmentation, however, since dealing with structured documents, replace the "link" relation with a "part_of" relation denoting the structural markup of documents:

```
about(T,D)   :-   term(T,D);
about(T,D)   :-   part_of(D2,D), term(T,D2);
```

Under a deterministic interpretation of Datalog, the above definitions would influence the *presence* of terms in contexts. However, knowledge augmentation here forms part of a summarisation model evaluated on a probabilistic Datalog layer. The augmentation of contexts thus leads to a change in the *probabilities* that terms occur in contexts. Following the assumption that the occurrences of terms in supercontexts and in their subcontexts are independent, the term probability in an augmented contexts is formed by the joined term probabilities of all subcontexts, modelled by the *Inclusion-Exclusion principle*.

Note that the relation "about" defines the augmentation of contexts, and is thus different from the *concept* "aboutness" discussed in 4.

5.3 Weighting

For the summarisation model outline in 4, two probabilities are combined in the retrieval function: the document term probability $P(t|d)$, and the collection term frequency $P(t|c)$. While the document term probability (or *tf*) corresponds to the linear probability $P(t|d)$, defined as $\frac{n_L(t,d)}{N_L(d)}$, where $n_L(t,d)$ is the number of occurrences of term t in document d, and $N_L(d)$ is the total number of terms in d [Rölleke *et al.*, 2006], the collection term frequency (or *idf*) is interpreted as a measure of "informativeness"; the more often a term is present in a collection of documents, the less informative it is. This can be formally stated as $\text{idf}(t,c) := -\log P(t|c)$, where in this case $P(t|c)$ does correspond to the linear probability. These different interpretations of probabilities are similar to traditional assumptions used in probabilistic modelling. For modelling the document probability $P(t|d)$, a disjointness assumption on the individual term occurrences in the document, followed by a summation of those probabilities, yields the linear probability $P(t|d)$. The inverse document frequency *idf*, however, cannot be modelled in this way, as it is based on informativeness rather than strict probabilities. This can be overcome by interpreting informativeness as a new kind of probabilistic assumption. To model both kinds of probabilities in probabilistic Datalog, it would be necessary to introduce new operators into the language, which allow the specification of *tf*-based and *idf*-based probabilities via rules. Such an extension of probabilistic Datalog is beyond the scope of this paper. For the purpose of the present discussion, and for the purpose of processing POLIS, we view *tf* and *idf* as extensional, i.e. pre-computed relations.

5.4 Combination of Weights

To implement the actual summarisation model as defined in section 4, the two probability distributions outlined in the previous section need to be combined to form a function of "aboutness". As aboutness is defined as a function of user-specified elements in relation to their surrounding contexts, these two element levels first need to be combined with the two probability distributions. As mentioned previously, the user specified elements will be called "impParts" in the present discussion. The surrounding context is called "firstPred" for practical reasons, as in single document summaries the surrounding context will be the first predicate defined in the POLIS expression. However, "firstPred" will usually be "/" for multi-document summaries, and could be at any desired level. The explicit specification of the level of "firstPred" is currently not supported by the syntax of POLIS, but could conceivably be implemented, and does not affect the summarisation function, and thus the present discussion.

The first step in weighting the respective elements is to augment the contexts with all terms present in their subcontexts. This is achieved via the "about" predicates defined previously.

```
impParts_about(T,D)    :-   impParts(T,D), about(T,D);
firstPred_about(T,D)   :-   firstPred(T,D), about(T,D);
```

The now augmented contexts can now be joined with the probabilistic predicates:

```
w_impParts_about(T,D)   :-   impParts_about(T,D),
                               idf(D);
w_firstPred_about(T,D)  :-   firstPred_about(T,D),
                               tf(T,D);
```

The combination of both weighted predicates yields the overall summarisation function, which attaches a probability of "aboutness" to all elements present at the user-specified level of granularity. We will call it "sum" here, for brevity:

```
sum(D)   :-   w_impParts_about(T,D),
                w_firstPred_about(T,D2);
```

This predicate *sum* now contains all textual elements at the user specified granularity together with their probability of "aboutness". To actually generate the summary, it is necessary to collect all those terms occurring in the contexts listed in *sum*. Assuming that contexts in *sum* are sorted by probability, this can be modelled within Datalog by using a *top-k* operator. The above example limits the number of textual elements that can participate in the generation of a summary to "up to three". This can be expressed in probabilistic Datalog as

```
limited_sum(D)   :-   sum(D):3;
summary(T)       :-   limited_sum(D), term(T,D);
?- summary(T);
```

which produces all the terms in the three most important textual elements.

5.5 Application to Sample Document

To show what results the above declarations produce in practice, we applied the Datalog program to the sample document in figure 2.

```
0.875   /doc[1]/body[1]/sec[1]/para[1]/sent[1]
0.875   /doc[1]/body[1]/sec[1]/para[1]/sent[2]
0.875   /doc[1]/body[1]/sec[1]/para[1]/sent[3]
```

For this sample document, all sentences have the same probability of being representative, as all sentences contain three words, all words are unique, and no further evidence is available. For "real" documents, term and sentence probabilities would be influenced by sentences of unequal length, and non-unique terms, so that the sentence probabilities would be more representative of "aboutness".

6 Evaluation

To evaluate the effectiveness of POLIS, the summarisation model implemented by the above Datalog expressions was tested on the AQUAINT corpus as used by the Document Understanding Conference (DUC). The test corpus is split into 50 topics, each of which consists of 25 documents covering that topic. Documents in the collection are newswire texts from three different sources: the Xinhua News Service, the New York Times, and the Associated Press. For each of the 50 topics, a title and a narrative are provided (neither of which were used for the work presented here). For example, the first topic in the collection is "D0601A" with the title "Native American Reservation System - pros and cons". "D0601A" consists of eight documents from AP, and 17 documents from the NYT.

The aim of DUC is to generate 250 word summaries of each of the 50 topics. The machine generated summaries are compared to four human generated reference summaries using the ROUGE evaluation framework [Lin and Hovy, 2003; Lin, 2004]. ROUGE calculates co-occurrence statistics for words occurring in both automatically generated summaries and the provided reference summaries, and provides measures of the degree of co-occurrence as precision and recall.

6.1 Efficiency

All experiments were carried out on an Intel Pentium 4 2.6 GHz machine, with 2GiB of main memory. We timed the processing of our Datalog programs using the built-in Linux "time" command. All reported times are real time. To time the processing, we broke down the overall POLIS program into subcomponents for evaluating *w_firstPred_about*, *w_impParts_about*, and *sum*.

To give a feel for the overall performance of our approach, we timed the above processing steps for the smallest (D0609I), a medium (D0634G), and a large topic (D0601A). We measured the size of the topics as the number of terms present in the subcollections after indexing. The results are shown in Table 1. The system performs

Topic	No. Terms	*firstPred*	*impParts*	*sum*
D0601A	13225	0.952s	3.967s	40.515s
D0634G	9428	0.479s	2.600s	41.135s
D0609I	4616	0.169s	0.585s	8.889s

Table 1: Performance comparison for small, medium and large topics

best for the smallest subcollection, and slows down for the larger topics. However, the speed decrease from the medium to the large topic is lower than the loss in performance going from the small to the medium topic, indicating that even larger topics could be processed without severe performance issues.

Topic	No. Terms	total Time	Time per *k*-term
D0601A	13225	45.434s	3.435s
D0634G	9428	44.214s	4.689s
D0609I	4616	9.643s	2.089s

Table 2: Performance per 1000 terms

The impact of topic size on the overall efficiency of the approach is shown in Table 2. For the three different collections, we measured the time it took for the system to process 1000 terms. For the smallest collection, the system is able to process a thousand terms every two seconds. This drops to a thousand terms every 4.7 seconds for the medium collection, and increases to a thousand terms per 3.4 seconds for the large collection. The figures show that for larger collections, additional system factors affect performance beyond mere term count. With the current implementation, assuming a processing speed of about 3.5 seconds per thousand terms, it would take about 5 minutes for the system to process a 100,000 word document. We believe this performance to be reasonably good, especially considering that summarisation systems rarely need to perform in real time.

6.2 Effectiveness

We compared the performance of POLIS to the average performance of all DUC participant summarisers. Note that performance values for DUC were available for topics 1 to 35 only. We therefore provide POLIS average performance scores both at 35 and at 50 topics.

	Precision	Recall	$F_{0.5}$
DUC avg.	0.38584	0.37141	0.37791
POLIS @35	0.33622	0.33258	0.33437
POLIS @50	0.33484	0.33095	0.33286

All values were reported at or above 95% significance level.

The above figures show that POLIS generated summaries perform slightly below the DUC average. The precision of POLIS generated summaries compared to the references provided is 4.962 percentage points below the DUC average at 35 topics, whereas recall values are 3.883 percentage points below DUC average at 35 topics. The overall lower performance of POLIS compared to DUC was mainly caused by four topics where results were significantly lower than the DUC average (topics 13, 20, 30, 33).

We believe that the lower results for these topics are caused by the way the current POLIS summarisation model determines the importance of textual elements: the "aboutness" of an element is only determined in terms of overlapping terms between the element and the overall document. Elements are not compared to other elements at the same level of granularity. The DUC collection consist of news-stories which cover the same topic. Different texts for the same topic can therefore contain similar textual elements, which in turn will get similar POLIS-scores. The generated summaries will then contain redundant sentences, which would not be present in the human generated reference summaries.

A more refined future version of the summarisation model will need to penalise textual elements at the user specified granularity which are very similar to other elements at the same level to overcome this problem.

7 Conclusion

We have introduced in this paper a summarisation logic, POLIS, and have shown how it can be processed Datalog-based. POLIS is a new abstraction layer for IR: it is tailored to enabling users and developers to describe summarisation strategies and preferences. The main contributions of this paper are the introduction to POLIS, the translation of POLIS expressions to probabilistic Datalog, and the application of the summarisation framework to the DUC test collection. The application to the test collection proves the feasibility of the Datalog-based implementation. With the current simple, tf-idf based strategy POLIS already shows a summarisation effectiveness comparable to that of less abstract, more collection specific summarisers. Future improvements of the weighting model and the specification of alternative probabilistic models for summarisation should results in higher effectiveness. Furthermore, we aim at increasing the expressiveness by supporting vague path expressions. Overall, POLIS is a novel contribution to logic-based information retrieval, which has been proposed in the past for describing the retrieval of documents only. With POLIS, we address requirements of a new generation of information management systems, where many retrieval tasks (including retrieval, summarisation, classification, ontology-matching, XML schema translations, etc) are described in high-level, logic-based abstraction layers.

References

[Alam *et al.*, 2003] H. Alam, A. Kumar, M. Nakamura, A. F. R. Rahman, Y. Tarnikova, and C. Wilcox. Structured and unstructured document summarization: Design of a commercial summarizer using lexical chains. In *International Conference on Document Analysis and Recognition*, pages 1147–1152, 2003.

[Edmundson, 1969] H.P. Edmundson. New Methods in Automatic Extraction. *Journal of the ACM*, 16(2):264–285, 1969.

[Forst *et al.*, 2006] Jan Frederik Forst, Anastasios Tombros, and Thomas Roelleke. Solving the Enterprise TREC Task with Probabilistic Data Models. *The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)*, 2006.

[Fuhr, 2000] Norbert Fuhr. Probabilistic Datalog: Implementing logical information retrieval for advanced applications. *Journal of the American Society for Information Science*, 51(2):95–110, 2000.

[Fum *et al.*, 1985] Danilo Fum, Giovanni Guida, and Carlo Tasso. Evaluating importance: A step towards text summarization. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 840–844, 1985.

[Hovy and Lin, 1997] E. Hovy and C. Lin. Automated text summarization in SUMMARIST, 1997.

[Knight and Marcu, 2002] Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artif. Intell.*, 139(1):91–107, 2002.

[Kupiec *et al.*, 1995] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73, New York, NY, USA, 1995. ACM Press.

[Lin and Hovy, 2003] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram cooccurrence statistics. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[Litkowski, 2004] K. C. Litkowski. Summarization Experiments in DUC 2004. In *Proceedings of the Document Understanding Conference 2004*, 2004.

[Luhn, 1958] H. P. Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[Rau *et al.*, 1989] L. F. Rau, P. S. Jacobs, and U. Zernik. Information extraction and text summarization using linguistic knowledge acquisition. *Inf. Process. Manage.*, 25(4):419–428, 1989.

[Reimer and Hahn, 1988] U. Reimer and U. Hahn. Text condensation as knowledge base abstraction. In *Proc. IEEE-88*, pages 338–344, 1988.

[Rölleke and Fuhr, 1998] Thomas Rölleke and Norbert Fuhr. Information retrieval with probabilistic Datalog. In F. Crestani, M. Lalmas, and C.J. van Rijsbergen, editors, *Logic and Uncertainty in Information Retrieval: Advanced models for the representation and retrieval of information*, chapter 9, pages 221–245. Kluwer Academic Publishers, Boston et al., 1998.

[Rölleke *et al.*, 2006] Thomas Rölleke, Theodora Tsikrika, and Gabriella Kazai. A general matrix framework for modelling information retrieval. *Inf. Process. Manage.*, 42(1):4–30, 2006.

[Saravanan *et al.*, 2005] M. Saravanan, S. Raman, and B. Ravindran. A probabilistic approach to multidocument summarization for generating a tiled summary. In *Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '05)*, pages 167–172, 2005.

[van Rijsbergen, 1986] C. J. van Rijsbergen. A nonclassical logic for information retrieval. *Comput. J.*, 29(6):481–485, 1986.

[Wolf *et al.*, 2004] C. G. Wolf, S. R. Alpert, J. G. Vergo, L. Kozakov, and Y. Doganata. Summarizing technical support documents for search: expert and user studies. *IBM Syst. J.*, 43(3):564–586, 2004.

Informationsextraktion aus Stellenanzeigen im Internet

Sandra Bsiri & Michaela Geierhos
Centrum für Informations- und Sprachverarbeitung
Ludwig-Maximilians-Universität München
D-80538 München, Deutschland
{sandra.bsiri|michaela.geierhos}@cis.uni-muenchen.de

Abstract

Dieser Beitrag beschäftigt sich mit der Informationsextraktion aus Stellenanzeigen im französischsprachigen Web. Ziel dieser Arbeit ist es, unstrukturierte Dokumente in Repräsentationsvektoren anhand lokaler Grammatiken zu transformieren. Auf diese Weise wird es möglich, den Stellenmarkt für Jobsuchmaschinen transparenter zu gestalten, indem nur auf dem Inhalt der Anzeige in Form von Darstellungsvektoren anstatt auf unübersichtlichem Fließtext gesucht werden muss.

1 Einführung

Das Internet hat sich dank seiner hohen Verbreitungseffizienz zum zentralen Medium des Stellenmarktes entwickelt [Fondeur *et al.*, 2005]. Gemäß einer Untersuchung, die der Personalvermittler *Kelly Services*[1] im ersten Quartal 2006 mit 19.000 beteiligten Personen aus 12 europäischen Ländern durchgeführt hat, ist das Internet inzwischen das an erster Stelle genutzte Kommunikationsmittel auf der Suche nach einem Arbeitsplatz. Die Studie kommt unter anderem zu dem Schluss, dass das Wachstumspotential des Internets bei der Arbeitsvermittlung keineswegs ausgeschöpft ist, was eine weitere, von der *Focus RH Gruppe* durchgeführte Studie [Focus RH, 2006] über die 500 wichtigsten Internet-Jobbörsen in Frankreich bestätigt.

Das Spektrum konkurrierender Stellenbörsen im Netz hat allerdings zu einer hohen Redundanz der Daten und eingeschränkter Transparenz geführt. So wird ein Großteil der auf dem Arbeitsmarkt verfügbaren freien Stellen über andere Kommunikationswege wie Firmenwebseiten, spezialisierte Diskussionsforen oder auch Kleinanzeigenverzeichnisse [Fondeur *et al.*, 2005] veröffentlicht. Dabei wird eine erhebliche Anzahl an Stellenangeboten unter heterogenen Formaten gleichzeitig auf mehreren Jobbörsen publiziert. Außerdem erschwert die weite Streuung der Stellenanzeigen im Internet ihre Zugänglichkeit für Web-Suchmaschinen und somit für den interessierten Bewerber.

In den letzten Jahren lag der Schwerpunkt bei vielen Online-Anbietern vorwiegend auf der Quantität der zur Verfügung gestellten Daten (der publizierten Stellenanzeigen bzw. der in den Datenbanken gespeicherten Lebensläufe) [2], wobei die Qualität der Suchergebnisse teilweise aus dem Blick verloren ging. Jedoch beschränken sich Jobsuchmaschinen dieses Typs in der Transformationsphase der Suchanfrage auf die traditionellen, rein statistischen Indexierungs- und Suchmethoden [Glöggler, 2003; Kowalski, 1997], anstatt den Vorteil der bereits semistrukturierten Daten [Ferber, 2003] zu nutzen und somit präzisere Suchergebnisse zu ermitteln. Deshalb sollte es das Ziel einer jeden Jobsuchmaschine sein, eine gewisse Transparenz in den Arbeitsmarkt zu bringen und die Fülle an offenen Stellenanzeigen virtuell zu vereinigen. Somit sollten die Ergebnisse *einer* spezifischen Suche alle ausschließlich relevanten Arbeitsangebote enthalten. Dabei wäre es wünschenswert, dass dies über eine einzige graphische Oberfläche und in Echtzeit für alle aktuellen, freien, online-publizierten Stellen realisiert würde.

Ähnlich wie bei [Flury, 2005] wird in dieser Arbeit am Fall des frankophonen Stellenmarktes illustriert wie mit linguistisch basierten Methoden alle im Netz verfügbaren Stellenangebote erstmals auf einer zentralisierten Plattform gesammelt und gefiltert werden können. Ein System zur automatischen Erkennung und Klassifikation von Firmen-Homepages wurde implementiert, um eine Firmendatenbank zu erstellen und auf aktuellem Stand zu halten. Basierend auf diesem Verzeichnis werden die HTML-Strukturen auf Ankertexte durchsucht, die zu den offenen Stellen führen. Lokale Grammatiken [Gross, 1993; 1997; Geierhos, 2006] und elektronische Lexika [Courtois *et al.*, 1990; 1997] wurden ausgearbeitet, um in einer zweiten Phase jene Informationen zu extrahieren, die für die Umwandlung der reinen Textform der Stellenangebote in ein semantisch strukturiertes Dokument notwendig sind. Über diese linguistische Analyse der einzelnen Einträge wird die Datenbank automatisch gefüllt und höchste Selektivität sowie Benutzerfreundlichkeit bei Suchanfragen gewährleistet.

Zur Realisierung einer verbesserten Jobsuchmaschine wurde im Rahmen dieser Arbeit ein Multi-Level-System aufgebaut, das in all seinen Analyse- und Bearbeitungsphasen von linguistischen Theorien und besonders vom Konzept „lokaler Grammatiken“ getragen wird. Das gesamte System (siehe Abbildung 1) basiert auf zwei hier in Abschnitt 2 und 3 ausgearbeiteten, interagierenden Modulen (A und B) und erinnert auf den ersten Blick an den Aufbau einer gewöhnlichen Suchmaschine:

A Lokalisieren der Stellenanzeigen im Web

B Dokumentanalyse und Informationsextraktion

C Bearbeitung der Suchanfragen

Jedoch liegt der Schwerpunkt unserer Ausführungen verstärkt auf Teil B und der damit verbundenen Informationsextratkion aus Stellenanzeigen, sowie ihrer Trans-

[1] siehe auch http://management.journaldunet.com/repere/outils-recherche-emploi.shtml

[2] siehe auch http://edito.keljob.com/index.php?id=27#1595 http://edito.keljob.com/recruteurs/articles/1206/barometre-octobre.html

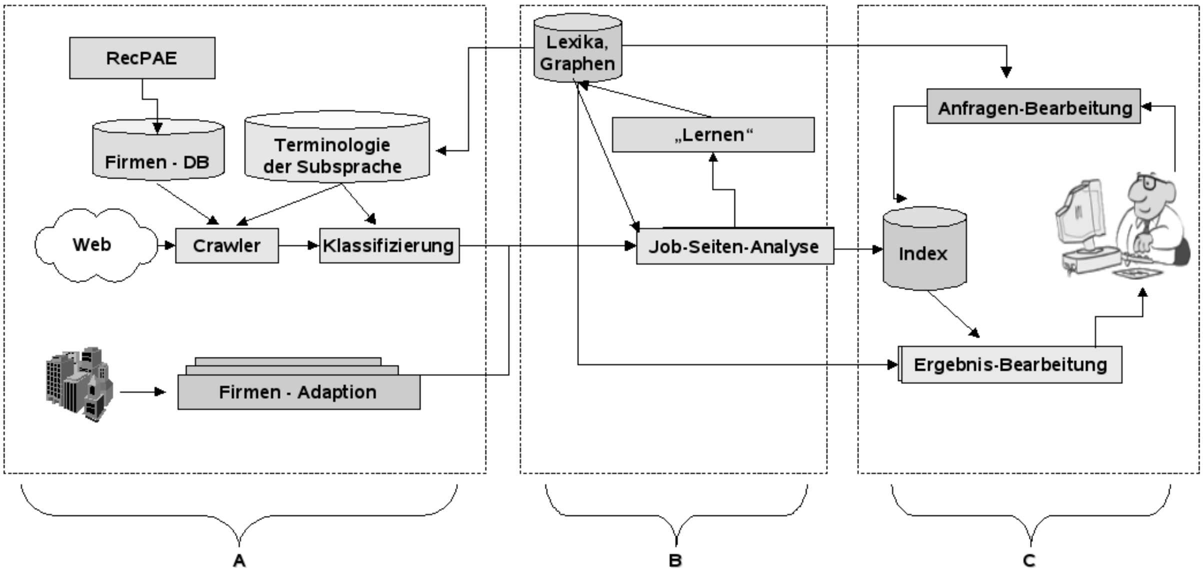


Abbildung 1: Übersicht zur Systemarchitektur

formation in Repräsentationsvektoren (Abschnitt 3), welche beim späteren Retrieval schnellere und qualitativ bessere Antworten auf Suchanfragen liefern sollen. So gibt Abschnitt 4 Einblick in die Ergebnisse der Systemevaluation und zeigt, wie vielversprechend die Werte für Precision (durchschnittlich 95.86 %) und Recall (durchschnittlich 92.98 %) bei der Erkennung anzeigenspezifischer Entitäten, wie z.B. Berufsbezeichnung, Arbeitsort etc. sind. Zuletzt werden noch mögliche Weiterentwicklungen und Anwendungen des hier beschriebenden Systems skizziert.

2 Lokalisieren von Stellenanzeigen im Web

Um die Idee einer zentralisierten Plattform für Stellenangebote zu verwirklichen, ist es notwendig, die im Internet verfügbaren Stellen automatisch zu finden. Zu diesem Zweck werden zwei unterschiedliche Strategien verfolgt: Einerseits werden bevorzugt Firmenwebseiten auf freie Stellen durchsucht, wofür ein System zur automatischen Erkennung dieser entwickelt wurde. Andererseits wird ein fokussierter Crawler eingesetzt, welcher mit Hilfe der speziellen Terminologie von Stellenanzeigen jede gecrawlte Website als solche identifiziert und dementsprechend klassifiziert.

2.1 Identifikation von Firmenwebsites

Die eben genannte Vorgehensweise – basierend auf einer Datenbank aus Internetadressen der jeweiligen Unternehmen – ermöglicht es, die auf den Firmenhomepages veröffentlichten Stellen zu finden. Dafür muss eine solche Datenbasis erst zur Verfügung gestellt und aufgrund der stetigen Bewegung auf dem Arbeitsmarkt immer aktuell gehalten werden. So ist es zwingend notwendig, jede URL, die als Webauftritt eines Unternehmens klassifiziert wurde, täglich aufzusuchen und auf ihre Erreichbarkeit zu testen. Des Weiteren müssen die dort veröffentlichten Stellenangebote im Vergleich mit ihrem Status des Vortages und anhand ihres Auschreibungszeitpunktes auf ihre Aktualität überprüft werden. Diese Update-Strategie erfordert ein großes Repertoire an Crawler-Seed-Links (URLs), welches aber mittels des neu entwickelten Systems zur automatischen Erkennung von Firmenwebsites[3] angelegt werden kann. Dabei handelt es sich um ein automatisches binäres Klassifikationssystem, welches für jede URL entscheidet, ob sie der Klasse „Organisation“ angehört oder nicht. Für die Klassifikation werden Merkmalsvektoren zur Entscheidungsfindung eingesetzt, welche sich bereits während der Lernphase bewährt haben, und nun eine Zuordnung zu insgesamt 10 vordefinierten Klassen ermöglichen.

Folgende Kriterien beeinflussen maßgeblich die Klassifikation der potentiellen Firmenwebsites:

- Orientierung an der HTML-Struktur des gesamten Dokumentes
- Auswertung der URLs, Meta-Informationen und des Titels
- Analyse und Einordnung der Ankertexte in vordefinierte semantische Klassen (siehe Tabelle 1)
- Identifikation des Firmennamens
- Berücksichtigung der Adresse, Telefonnummer, Handelsregisternummer, etc.
- Extraktion typischer Formulierungen und komplexer Terminologie

Unter Berücksichtigung von Flash-animierten Homepages konnten ca. 72 % der potiellen Firmenwebsites als solche korrekt identifiziert werden. Schließt man jedoch die Menge dieser Seiten aus, ist die Erkennungsrate deutlich höher (86 %) und stellt eine ausbaufähige, aber dennoch solide Basis für unser Vorhaben dar.

Firmennamenerkennung

Die automatische Erkennung von Organisationsnamen taucht relativ oft in der Literatur auf [Mallchok, 2004] und ist ein Teilbereich der automatischen Erkennung von Eigennamen (Named-Entity-Recognition). Es hat sich gezeigt, dass eine Liste von Organisationsnamen nicht ausreichend ist, um eine hohe Annotierungsquote zu erreichen,

[3] Die Firmenwebsites werden hier pro Webauftritt *(web site)* und nicht pro Webseite *(web page)* klassifiziert.

Carrière *(Einstellungsmöglichkeiten)*	Nous recrutons *(Wir stellen ein)* Nos offres d'emploi *(Unsere Stellenangebote)*
Produits/Services *(Produkte/Dienstleist.)*	Nos Produits *(Unsere Produkte)* Accès à la boutique *(Zum Shop)*
Contact *(Kontaktinformation)*	Nous contacter *(Kontaktieren Sie uns)* Pour venir nous voir *(Besuchen Sie uns)* Nos coordonnées *(Unsere Kontaktdaten)*
Société *(Firmeninformationen)*	Notre Société *(Unser Unternehmen)* Qui sommes nous? *(Wer wir sind)* Entreprise *(Unternehmen)*
Presse	Communiqués de Presse *(Pressemitteilungen)* La presse et nous *(Wir in der Presse)* Presse infos *(Presseinformationen)* media *(Medien)*
Clients/Partenaires *(Kunden/Partner)*	Nos Clients *(Unsere Kunden)* Espace clientèles *(Kundenbereich)* Nos partenaires *(Unsere Partner)* Relation client *(Kundenbeziehung)*

Tabelle 1: Ausgewählte Beispiele klassifizierender Ankertexte

da sehr viele Ambiguitäten bei Firmenbezeichnungen existieren. Allerdings sind alle bereits veröffentlichten Systeme gleichermaßen stark von den jeweiligen Trainingskorpora oder von der verwendeten Subsprache [Harris, 1968; 1988] abhängig. Um nun die Leistung dieses Systems der automatischen Klassifikation von Firmenwebseiten zu verbessern, wird zusätzlich zu den oben beschriebenen Eigenschaften der Name der Unternehmen extrahiert. Genau diese Information ist nötig, um als einer der Deskriptoren zu fungieren, um letztendlich die Klassifikationsentscheidung zu treffen.

Insgesamt wurden zwei verschiedene Methoden zur Extraktion der Firmennamen ausgearbeitet. Einerseits wird das linguistische Konzept der lokalen Grammatiken[4] angewandt, um die entsprechenden firmentypischen Kontexte zu beschreiben. Andererseits wurde ein Algorithmus entwickelt, der die Segmentierung des Domänennamens der zu analysierenden Website vornimmt.

Lokale Grammatiken [Gross, 1997] ermöglichen die Beschreibung eines lokalen Kontextes und schränken bestimmte lexikalische oder snytaktische Einheiten auf ein Fenster fester Größe ein. Dadurch ermöglichen sie es, Mehrdeutigkeit zu vermeiden bzw. einzugrenzen. In der Tat sollen die Auslöser – die einleitenden Kontexte – eines semantisch-syntaktischen Musters identifiziert, und mittels Bootstrapping [Gross, 1999] dessen Kontexte detailliert beschrieben werden.

In dieser Phase liegt der Schwerpunkt ganz auf der Sammlung externer Kontexte der Organisationsnamen, welche ebenfalls durch ein Bootstrapping-Verfahren mit Graphen [Senellart, 1998a; 1998b] ermittelt wurden und anschließend darin Verwendung fanden.

Floskeln und Firmenterminologie

Für die Erkennung und Extraktion von firmentypischen Kenngrößen, Standardredewendungen (Floskeln) oder unternehmensspezifischen Kontexten sind lokale Grammatiken ein adäquater Formalismus, um präzise und modular die rechten und linken Kontexte lexikalischer Einheiten zu beschreiben. Auf diese Weise kann die hohe syntaktische Variabilität bewältigt werden, was klare Vorteile gegenüber einer normalen Stringsuche bietet. Denn eine gewöhnliche Suche würde eine Liste aller gesammelten Floskeln mit dem Text der Website abgleichen, was dazu führen kann, dass bestimmte Ausdrücke mit minimalen morphosyntaktischen Variationen nicht mehr gefunden werden. Im Gegensatz dazu ist Bootstrapping mittels lokaler Grammatiken ein vielversprechender Ansatz, was folgendes Beispiel illustriert:

- Notre société , leader mondial sur le marché [...]
- Notre société est leader européen dans le secteur [...]
- Notre société leader dans son domaine [...]

Obwohl diese drei Phrasen sich auf lexikosyntaktischer Ebene stark unterscheiden, beschreiben sie einen ähnlichen Kontext des Wortes „leader". Doch mit Hilfe diverser statistischer Methoden, auf denen das System zur Extraktion von Konzepten basiert, können Ausdrücke dieser Gestalt aufgrund ihrer geringen Frequenz ignoriert werden, da trotz der syntaktischen Unterschiede bei der Paraphrasierung die Semantik erhalten bleibt. Jedoch ist es möglich diese Paraphrasen in einer lokalen Grammatik zusammenzufassen, wie sie beispielsweise in Abbildung 2 dargestellt wird.

Sobald eine Firmenhomepage als solche erkannt wurde, beginnt die Suche nach den Stellenangeboten, welche sich am HTML-Gerüst der Seite orientiert. Zu diesem Zweck wurde die Klasse der „Jobs" eingeführt, welche auf die Informationen der Ankertexte (siehe Tabelle 1) referiert, die

[4] Lokale Grammatiken kann man als „Landkarten der Sprache" bezeichnen [Mallchok, 2004], die einerseits Sequenzen von Wörtern, welche semantische Einheiten bilden, und andererseits syntaktische Strukturen beschreiben. Besonders auf dem Gebiet der lexikalischen Disambiguierung werden lokale Grammatiken verstärkt eingesetzt, welche üblicherweise durch einen endlichen Automaten bzw. einen Transduktor repräsentiert werden. In der Regel werden lokale Grammatiken in Form von Graphen [Paumier, 2004] visualisiert.

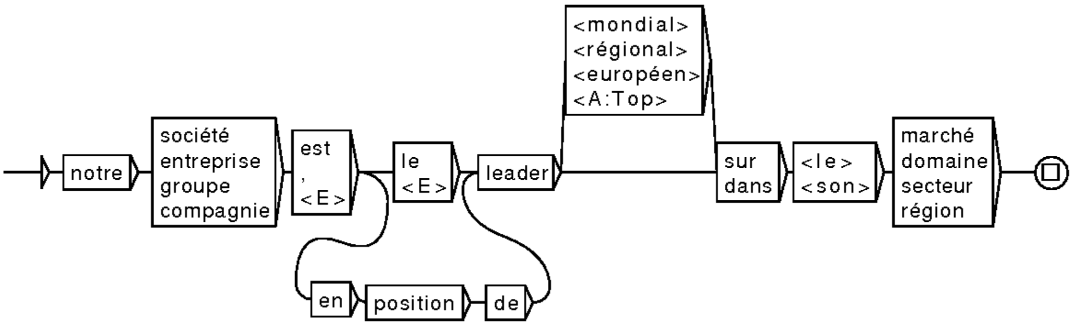


Abbildung 2: Lokale Grammatik, die den Kontext des Wortes *leader* modelliert

zu den offenen Stellenangeboten des jeweiligen Unternehmens führen, wie z.B.

- Wir stellen ein
- Unsere Stellenangebote

Bereits während der Lernphase konnten über 80 auf Stellenausschreibungen hinweisende Sequenzen dieser Kategorie zugeordnet werden.

2.2 Fokussierte Crawler

Eine weitere Vorgehensweise zum Auffinden von Stellenanzeigen konzentriert sich auf die typische Terminologie eines Stellenangebots, die einer Subsprache [Harris, 1968; 1988] gleichkommt. Dafür wurde noch während der Lernphase eine Reihe von Floskeln und komplexen Formulierungen gesammelt, die sich in zwei Typen einteilen lassen: Nominale Phrasen, welche die Stellenbeschreibungen semantisch strukturieren und sich in den Überschriften der einzelnen Abschnitte der Anzeige manifestieren, und Redewendungen, welche spezifische Verben oder Nomen der hier erwähnten Subsprachen beinhalten und nur im Kontext von Stellenangeboten Sinn ergeben.

Mit Hilfe der erwähnten Terminologie kann ein fokussierter Crawler nun entscheiden, ob eine Website in die Kategorie „Stellenanzeige“ einzuordnen ist, selbst wenn die HTML-Struktur des Dokuments keinerlei Aufschluss über eine mögliche semantische Gliederung der Anzeige gibt, oder statt allgemeiner Floskeln nur bestimmte Formulierungen darin auftreten.

Da die in Phase A (siehe Abbildung 1) gefundenen Stellenanzeigen uns nun als Volltext vorliegen, müssen sie anhand ihres Inhalts strukturiert werden, so dass ihr Informationsgehalt transparenter für den Jobsuchenden wird. Zu diesem Zweck werden einerseits die Daten des Stellenangebots über ihre semantischen Typen (z.B. Berufsbezeichnung, Firmenname) extrahiert, um später die Datenbank der Stellenanzeigen zu erstellen. Andererseits werden die Dokumente in Form von Vektoren dargestellt, welche die eben genannte Information enthalten, und auf denen letztendlich die Jobsuche effizienter gestaltet werden kann.

3 Informationsextraktion und Generieren von Dokumentrepräsentationsvektoren

Die zweite Phase des vorgestellten Systems (B) – der Hauptteil der vorliegenden Arbeit – behandelt die Informationsextraktion und die automatische Umwandlung der reinen Textform eines Stellenangebots in ein semantisch strukturiertes Dokument. Wir haben uns zu diesem Zweck auf die Erstellung einer bedeutenden Anzahl lokaler Grammatiken und elektronischer Lexika konzentriert, die es uns erlauben, die Datenbank der Stellenangebote automatisch zu füllen. Die Struktur der Datenbank kann als ein Formular betrachtet werden, über das die in jeder Stellenanzeige vorliegende Information strukturiert wird. Die gängigen Konzepte von Jobsuchmaschinen – auch neuerer Generation – erfordern hingegen ein manuelles Ausfüllen der entsprechenden Formulare.

Beispiel des auszufüllenden Formulars	
Datum der Veröffentlichung	22. Jan 2007
Bewerbungsfrist	fin février *(Ende Februar)*
Einstellungsdatum	mi-mars *(Mitte März)*
Stellenbezeichnung	ingénieur d'étude en électromécanique *(Elektromechanikprojektingenieur)*
Art des Vertrags	intérim à temps partiel : 1 à 2 jours/semaine *(Zeitarbeit : 1 bis 2 Tage/Woche)*
Anstellungszeitraum	8 mois renouvelables *(8 Monate verlängerbar)*
Arbeitsort	sud-est de Paris *(süd-westlich von Paris)*
Gehaltsvorschlag	selon profil *(nach Profil)*
Referenz der Stelle	MOR34544/ing-21
Arbeitserfahrung	expérience de 2 à 3 ans dans un poste similaire *(2 bis 3 Jahre Erfahrung in einem ähnlichen Beruf)*
gewünschte Ausbildung	de formation Bac+5 de type école d'ingénieur *(Abschluss als Diplom-Ingenieur)*
Firmenname	CGF Sarl
Firmensitz	**Adresse :** 34 bis rue Berthauds, 93110 Rosny **Tel :** 0 (+33) 1 12 34 45 67 **Fax :** 0 (+33) 1 12 34 45 68 **Email :** contact@cgf.fr **Homepage :** http:///www.cgf.fr
Kontakt	Directeur des RH, Mr. Brice *(Personaldirektor, Mr. Brice)*
Firmenbranche	Construction électromécanique *(Elektromechanische Konstruktion)*

Tabelle 2: Strukturiertes Stellenangebot in der Datenbank

Über ein im ersten Arbeitsschritt des Systems (siehe Abschnitt 2) gefundenes Dokument werden nun gewisse In-

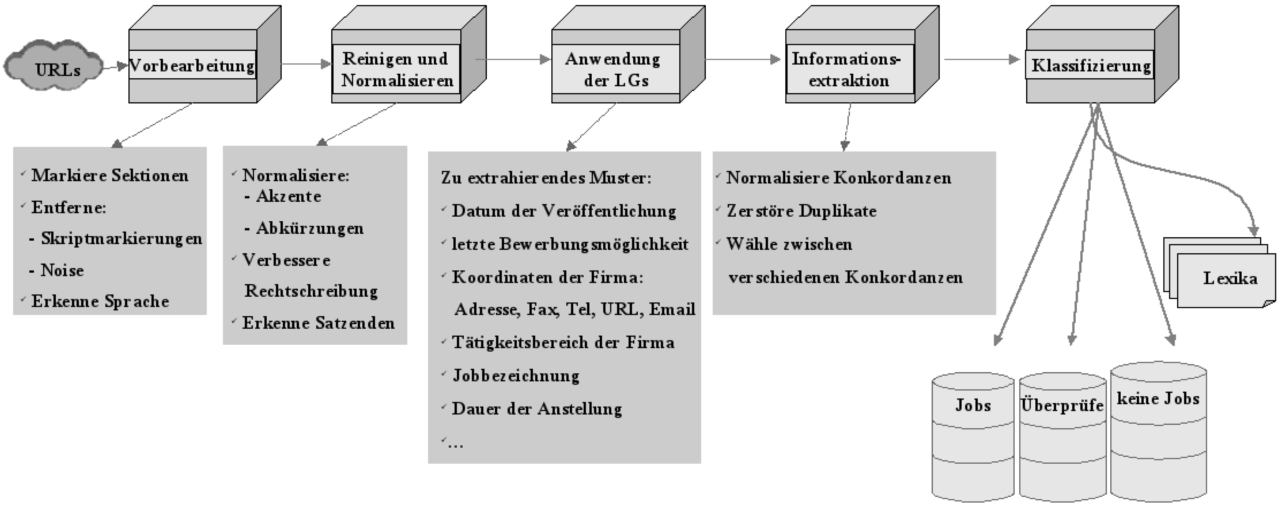


Abbildung 3: Vearbeitungsphasen eines potentiellen Stellenangebots

formationen automatisch aus dem Stellenangebot extrahiert, um damit ein Formular zu befüllen, wie es in Tabelle 2 abgebildet ist.

Für diese Transformation der ursprünglichen HTML-Dokumente in dieses Schema sind verschiedene Operationen nötig, deren chronologischer Ablauf in Abbildung 3 dargestellt wird.

In dieser Abbildung können fünf unterschiedliche Phasen unterschieden werden, auf die im folgenden näher eingegangen werden soll.

3.1 Vorverarbeitung

In diesem Schritt wird jedes Dokument mit semantisch-strukturellen Markierungen *(*[*TagMISSION*], [*TagPROFIL*], [*TagFORMATION*], *usw.)* gelabelt. Bereits während der Lernphase wurden insgesamt 13 Klassen ausgearbeitet, die Floskeln oder Phrasen enthalten, welche die Rolle von Untertiteln in einer Anzeige spielen, und durch eines der eben genannten Tags repräsentiert werden. So spiegelt [*TagMISSION*] beispielsweise die Klasse der *Tätigkeiten* wieder und enthält u.a. folgende Floskeln:

- Ihr Aufgabenbereich
- Ihre Aufgabengebiete sind
- Das erwarten wir von Ihnen:

Im Verlauf der Systementwicklung konnten wir feststellen, dass Stellenangebote grundsätzlich auf drei verschiedene Arten geschrieben werden. Entweder werden Floskeln benutzt, um das Dokument in einem hohen Maß zu strukturieren, oder es handelt sich um reine textuelle Beschreibung, ohne deutliche Struktur, oder es sind sehr kompakte Anzeigen mit einem Minimum an Information.

Nachdem die semantische Struktur des zu untersuchenden Dokuments analysiert und markiert wurde, werden alle HTML-Formatierungen sowie Programmskripte (z.B. JavaScript, ActionScript) gelöscht.

Anschließend wird die Sprache des verbliebenen Textes durch einen Abgleich mit verschiedensprachigen Wörterbüchern bestimmt, wobei dieses Modul aber nur Aufschluss darüber gibt, wie groß die Wahrscheinlichkeit dafür ist, dass das getestete Fragment in dieser Sprache geschrieben ist. So können alle nicht als französische Stellenbeschreibungen erkannte Dokumente herausgefiltert werden, da sie für unsere Zwecke ohne Bedeutung sind.

3.2 Bereinigung und Normalisierung

Die Bereinigung der Daten besteht darin, die orthographischen Fehler sowie die fehlenden Akzente im Text zu erkennen und zu korrigieren, wobei nur die nicht ambigen Einträge berücksichtigt werden. Dies geschieht mit Hilfe einer Liste von häufigen Rechtschreibfehlern, die schon während der Lernphase auf einem großen Korpus von Stellenangeboten gesammelt wurden.

In der Normalisierungsphase wird versucht Abkürzungen zu identifizieren und diese durch ihre jeweiligen Originalformen zu ersetzen, z.B. *(ing. ↦ ingénieur, comm. ↦ commercial, usw.)*. Zu diesem Zweck wurde auch eine Liste von Abkürzungen extrahiert und ihren entsprechenden ausgeschriebenen Wortlauten zugeordnet.

Gegen Ende dieses Arbeitsschrittes verfügt man über einen bereinigten und normalisierten Text, auf dem nun die syntaktischen und lexikalischen Analysen durchgeführt werden können.

3.3 Anwendung der lokalen Grammatiken

Wir haben mehrere spezialisierte Lexika für einfache Wörter und für Mehrwortlexeme erstellt, welche die Terminologie der erwähnten Subsprache erfassen und auch konform mit dem DELA-Format [Courtois *et al.*, 1990; 1997] sind. Diese von uns eingehaltene Konvention erlaubt die Anwendung lokaler Grammatiken innerhalb der LGPL[5]-Software Unitex[6]. Bei dieser Plattform handelt es sich um ein Korpusverarbeitungssystem, welches es ermöglicht, mit elektronischen Lexika umzugehen und lokale Grammatiken in Form eines Finite-State-Graphen (Directed Acyclic Graph: vgl. Abb. 2) zu entwickeln und auf ein Korpus anzuwenden.

Um diese Graphen verarbeiten zu können, wird zunächst der Text in Unitex durch folgende Prozess-Pipeline geschleust:

1. *Convert*: Konvertiert den Text in Unicode (UTF-16LE).
2. *Normalize*: Normalisiert die Sonderzeichen, die Leerzeichen und die Zeilenumbrüche.
3. Satzendeerkennung

[5] GNU **L**esser **G**eneral **P**ublic **L**icense

[6] http://www-igm.univ-mlv.fr/ unitex/

4. Auflösung von Kontraktionen (z.B. *d'une* ↦ *de une*).
5. *Tokenize*: Tokenisiert den Text aufgrund des Alphabets der jeweiligen Sprache.
6. *Dico*: Führt eine lexikalische Analyse durch, indem die Lexika auf die Tokenliste des Textes angewendet, und jedem Wort seine möglichen grammatikalischen Kategorien zugeordnet werden.

Nach der Ausführung der lexikalischen Analyse folgt die Phase der Informationsextraktion mit Hilfe der lokalen Grammatiken. Für jeden Typ der zu extrahierenden Informationen (derzeit ca. 20 Stück[7]) wurden mehrere Grammatiken entwickelt, die iterativ bzw. kaskadiert [Friburger *et al.*, 2001] sowie mit unterschiedlicher Priorität ausgeführt werden. Beispielsweise wurden zur Extraktion der Berufsbezeichnungen die entsprechenden lokalen Grammatiken auf acht Prioritätsebenen verteilt. Die Graphen der Ebene $n + 1$ werden nur ausgeführt, wenn die Grammatiken der Ebene n mit höherem Vorrecht keine Teffer liefern konnten.

3.4 Informationsextraktion

In der Phase, die wir als Informationsextraktion bezeichnen, geht es darum, die extrahierten Sequenzen zu normalisieren, sowie Duplikate und unvollständige Teilsequenzen zu entfernen. Nach einer Vielzahl von heuristischen Tests zeigte sich, dass es sinnvoll ist, jeweils den längsten Match zu bevorzugen. Obwohl man einräumen muss, dass in wenigen Fällen die falsche Entscheidung getroffen wird, blieben doch auf diese Weise die Verluste geringer.

Diese Entscheidungsphase ist notwendig, weil es oft vorkommt, dass Sequenzen mit verschiedenen Pfaden in den DAGs, die als Transduktoren fungieren, erkannt und dabei auf unterschiedliche Weise annotiert werden.

3.5 Klassifikation

Da es sehr selten vorkommt, dass ein Stellenangebot alle gesuchten Informationen gleichzeitig enthält, wurden Regeln ausgearbeitet, die in Abhängigkeit der jeweils erkannten Informationen im Dokument die Klassifikation in die Datenbank der Stellenangebote ermöglichen.

So wird die URL als Stellenanzeige klassifiziert und in der Datenbank durch die gefundenen Informationen indexiert, wenn einige der semantisch-strukturellen Floskeln, die Berufsbezeichnung und das Einstellungsdatum gefunden werden. Mit Hilfe einer benutzerfreundlichen Website kann jederzeit die annotierte Anzeige betrachtet werden. Dabei ist es auch möglich, die fehlenden Informationen manuell zu ergänzen. Denn die Benutzeroberfläche sollte eine schnelle Orientierung im Dokument ermöglichen und mit wenig Klicks können die fehlenden Felder des Formulars sowie die semantischen Wörterbücher um die farbig hervorgehobenen unbekannten Wörter erweitert werden.

Anwendungen

An einem konkreten Beispiel sollen kurz typische Ergebnisse und die Qualität des Extraktionsprozesses anhand der entwickelten lokalen Grammatiken dargelegt werden.

In Abbildung 4 wird ein annotiertes Dokument nach Durchführung aller hier beschriebenen Verarbeitungsschritte gezeigt.

Die erkannten, semantisch-strukturellen Floskeln, die in 13 Klassen[8] eingeteilt wurden, sind grün hervorgehoben. Im Beispiel konnten alle Einträge der fünf in der Stellenanzeige vorhandenen Klassen gefunden werden (*„TAGCPN = Firmeninformationen“, „TAGPOSTE = Stellenbeschreibung“, „TAGEXP = Qualifikation des Kandidaten“, „TAGSALAIRE = Gehalt“, „TAGCONTACT = Firmenkontakt“*). Einige dieser Floskeln haben eine starke Filterfunktion, da sie ausschließlich in Stellenanzeigen vorkommen.

14 von den 20 gesuchten Informationen (rosa hervorgehoben) wurden mit mehr als 100 verschiedenen lokalen Grammatiken eindeutig identifiziert und mit den korrekten semantischen Tags assoziert, wie z.B.

`<PosteName>` = *(Berufsbezeichnung)*

`<Location>` = *(Standort)*

`<Duree>` = *(Dauer des Vertrags)*

`<Salaire>` = *(Angebotenes Gehalt)*

`<CPN>` = *(Firmenname)*

`<DomainOrg>` = *(Tätigkeitsbereich des Unternehmens)*

`<TypeContrat>` = *(Art des Vertrags)*

`<Reference>` = *(Stellenreferenz)*

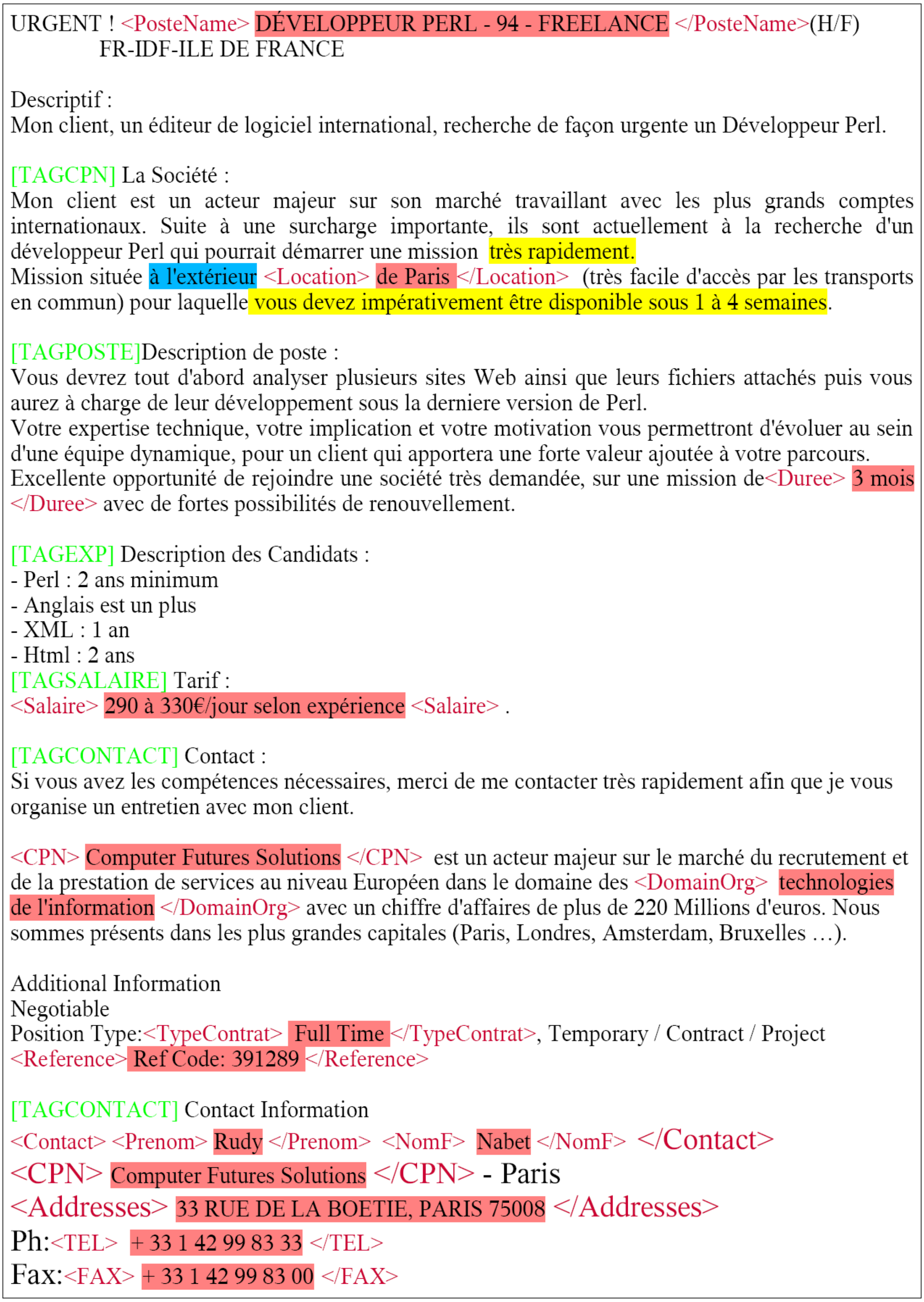
In diesem Beispiel gibt es eine Sequenz, die zur gefundenen Konkordanz gehören sollte, aber nur teilweise und daher fehlerhaft durch die Grammatik extrahiert wurde. Der gefundene Arbeitsort wäre hier gemäß automatischer Extraktion *„de Paris“* obwohl es *„à l'extérieur de Paris“ (Großraum Paris)* sein sollte. In dieser primären Konfiguration wurde die vorhandene Semantik stark verändert und die Performanz unseres System verschlechterte sich deutlich: Wenn ein Jobsuchender nach einer Stelle in der Stadt Paris selbst suchen würde, bekäme er aufgrund des hier generierten Fehlers diese Stelle angeboten. Aber wenn der Arbeitsuchende auch außerhalb von Paris eine Stelle suchen wollte, hätte er nach *IDF oder Île de France* gesucht, was dem „Großraum Paris“ entspricht.

Gesuchte Sequenzen, die durch die lokale Grammatiken nicht gefunden wurden, sind gelb hervorgehoben. Im Beispiel handelt es sich um das „Einstellungsdatum“, das durch die zwei Ausdrücke *„vous devez impératiement être disponible sous 1 à 4 semaines“ (Sie sollten in 1 bis 4 Wochen verfügbar sein)* und *„... pourrait démarrer une mission très rapidement“ (... könnte die Tätigkeit sehr bald beginnen)* umschrieben wurde. Diese Ausdrücke stellen keine feste Datumsangaben dar, aber beinhalten die gewünschte Information. Diese beiden fehlenden Informationen sind inzwischen den entsprechenden lokalen Grammatiken hinzugefügt worden. In diesem Sinne werden alle lokalen Grammatiken ständig erweitert, was dank der verfügbaren Unitex-Oberfläche [Paumier, 2004] sehr schnell umgesetzt werden kann.

Ein weiterer Vorteil der lokalen Grammatiken ist die gute Übersichtlichkeit der schon beschriebenen und der noch fehlenden syntaktischen Strukturen. Falls eine Information nicht extrahiert werden konnte, kann ein Pfad aufgrund der hohen Modularität sehr schnell in den Graphen eingefügt werden.

[7] Darunter fallen u.a. Informationen wie Berufsbezeichnung, Firmenname, Firmensitz, Arbeitsort und evtl. Gehaltsangaben.

[8] Mission, Tâches, Compétences, Qualités, Connaissances, Expérience et Formation, Durée, Date de début, Lieu, Description du poste, Salaire, Coordonnées du contact, Entreprise



URGENT ! <PosteName> DÉVELOPPEUR PERL - 94 - FREELANCE </PosteName>(H/F)
FR-IDF-ILE DE FRANCE

Descriptif :
Mon client, un éditeur de logiciel international, recherche de façon urgente un Développeur Perl.

[TAGCPN] La Société :
Mon client est un acteur majeur sur son marché travaillant avec les plus grands comptes internationaux. Suite à une surcharge importante, ils sont actuellement à la recherche d'un développeur Perl qui pourrait démarrer une mission très rapidement.
Mission située à l'extérieur <Location> de Paris </Location> (très facile d'accès par les transports en commun) pour laquelle vous devez impérativement être disponible sous 1 à 4 semaines.

[TAGPOSTE]Description de poste :
Vous devrez tout d'abord analyser plusieurs sites Web ainsi que leurs fichiers attachés puis vous aurez à charge de leur développement sous la derniere version de Perl.
Votre expertise technique, votre implication et votre motivation vous permettront d'évoluer au sein d'une équipe dynamique, pour un client qui apportera une forte valeur ajoutée à votre parcours.
Excellente opportunité de rejoindre une société très demandée, sur une mission de<Duree> 3 mois </Duree> avec de fortes possibilités de renouvellement.

[TAGEXP] Description des Candidats :
- Perl : 2 ans minimum
- Anglais est un plus
- XML : 1 an
- Html : 2 ans

[TAGSALAIRE] Tarif :
<Salaire> 290 à 330€/jour selon expérience <Salaire> .

[TAGCONTACT] Contact :
Si vous avez les compétences nécessaires, merci de me contacter très rapidement afin que je vous organise un entretien avec mon client.

<CPN> Computer Futures Solutions </CPN> est un acteur majeur sur le marché du recrutement et de la prestation de services au niveau Européen dans le domaine des <DomainOrg> technologies de l'information </DomainOrg> avec un chiffre d'affaires de plus de 220 Millions d'euros. Nous sommes présents dans les plus grandes capitales (Paris, Londres, Amsterdam, Bruxelles …).

Additional Information
Negotiable
Position Type:<TypeContrat> Full Time </TypeContrat>, Temporary / Contract / Project
<Reference> Ref Code: 391289 </Reference>

[TAGCONTACT] Contact Information
<Contact> <Prenom> Rudy </Prenom> <NomF> Nabet </NomF> </Contact>
<CPN> Computer Futures Solutions </CPN> - Paris
<Addresses> 33 RUE DE LA BOETIE, PARIS 75008 </Addresses>
Ph:<TEL> + 33 1 42 99 83 33 </TEL>
Fax:<FAX> + 33 1 42 99 83 00 </FAX>

Abbildung 4: Beispiel eines automatischen annotierten Stellenangebots

4 Evaluierung der Extraktionsergebnisse

Um die Qualität unseres Systems in der Erkennungsphase von stellentypischen Informationen zu demonstrieren, wurde ein kleines Testkorpus[9] bestehend aus ca. 1000 Stellenanzeigen manuell annotiert. Damit war es uns nun möglich, die Precision- und Recall-Werte für die automatisch gefundenen Resultate anzugeben und letztendlich auszuwerten, ob das System unsere Erwartungen erfüllt.

Extrahierter Informationstyp	Precision	Recall
Berufsbezeichnung	96.9 %	93.3 %
Firmenname	94.3 %	90.6 %
Firmensitz (Adresse)	93,0 %	92,3 %
Gehaltsangabe	97,1 %	91,8 %
Arbeitsort	98,0 %	96,9 %
Im Durchschnitt	95.86 %	92.98 %

Tabelle 3: Evaluationsergbnisse auf den Textkorpora

In Tabelle 3 wird deutlich, wie vielversprechend die Werte für Precision (durchschnittlich 95.86 %) und Recall (durchschnittlich 92.98 %) bei der Erkennung anzeigenspezifischer Entitäten sind. Für die hier vorgestellte Arbeit beschränkt sich die Auswertung auf fünf der wichtigsten von insgesamt 13 Informationsklassen. Daran wird ersichtlich, dass die eindeutige Identifikation des Firmennamens im Vergleich zu den anderen Kategorien noch die meisten Schwierigkeiten bereitet, aber dennoch um einiges qualitätiv besser ist als das, was marktführende Jobsuchmaschinen leisten.

5 Ausblick

Teile des hier konzipierten Systems einer optimierten Jobsuchmaschine können auch für andere Zwecke benutzt werden. Einerseits hält dieses System die Datenbank der Firmen und ihrer Websites immer auf dem aktuellen Stand und kann so für die verschiedensten Anwendungen von großem Nutzen sein. Andererseits konsultieren immer mehr Menschen das Internet, um beispielsweise einen Dienstleister oder Anbieter einer bestimmten Branche oder Region zu finden. Unser aktuelles Ziel ist es ein Klassifikationssystem zu entwickeln, das automatisch jedes Stellenangebot in die entsprechende Berufsbranche einordnen kann. Zu diesem Zweck soll eine Ontologie der Berufsbezeichnungen erstellt werden, welche es erlaubt, die Suche auf die semantischen Beziehungen zwischen den Anfragetermen zu erweitern.

Literatur

[Bsiri, 2007] Sandra Bsiri. *Extraction d'information: Génération automatique d'une base de données d'offres d'emploi*. Doktorarbeit, LMU München, 2007.

[Courtois *et al.*, 1990] Blandine Courtois, Max Silberztein. Dictionnaires électroniques du français. In *Langues française 87*, 11-22. Larousse, Paris, 1990.

[Courtois *et al.*, 1997] Blandine Courtois et al. *Dictionnaire électronique des noms composés DELAC : les composants NA et NN* Rapport Technique du LADL 55, Paris, Université Paris 7, 1997.

[Ferber, 2003] Reginald Ferber. Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web. dpunkt.verlag, 2003.

[Flury, 2005] Wolfgang Flury. *Information Extraction aus Online-Stellenanzeigen für eine Jobsuchmaschine*. Abschlussarbeit im Aufbaustudium „Computerlinguistik“, LMU München, 2005.

[Focus RH, 2006] Focus RH. *Le guide des 500 meilleurs sites emploi*. Jeunes Editions, Levallois-Perret, 2006.

[Fondeur *et al.*, 2005] Yannick Fondeur, Carole Tuchszirer. Internet et les intermédiaires du marché du travail. In *La lettre de l'IRES, n° 67*. IRES - Institut de Recherches Economiques et Sociales, Noisy-le-Grand, 2005.

[Friburger *et al.*, 2001] Nathalie Friburger, Denis Maurel. Elaboration d'une cascade de transducteurs pour l'extraction des noms personnes dans les textes. In *TALN 2001*, Tours, 2-5 Juli 2001.

[Geierhos, 2006] Michaela Geierhos. Lokale Grammatiken. In *Grammatik der Menschenbezeichner in biographischen Kontexten*, 16-23, Magisterarbeit, LMU München, 2006.

[Glöggler, 2003] Michael Glöggler. Suchmaschinen im Internet. Springer, Berlin, 2003.

[Gross, 1993] Maurice Gross. *Local grammars and their representation by finite automata*. M. Hoey (Hrsg.): Data, Description, Discourse, Papers on the English Language in honour of John McH Sinclair, 26–38. HarperCollins, London, 1993.

[Gross, 1997] Maurice Gross. *The Construction of Local Grammars*. E. Roche und Y. Schabès (Hrsg.): Finite-State Language Processing (Language, Speech, and Communication), 329–354. MIT Press, Cambridge, Massachusetts, 1997.

[Gross, 1999] Maurice Gross. A bootstrap method for constructing local grammars. In *Contemporary Mathematics: Proceedings of the Symposium*, University of Belgrad, 229–250. Belgrad, 1999.

[Harris, 1968] Zellig S. Harris. Mathematical Structures of Language. In *Interscience Tracts in Pure and Applied Mathematics 21*, 230–238. Interscience Publishers John Wiley & Sons, New York, 1968.

[Harris, 1988] Zellig S. Harris. Language and Information In *Bampton Lectures in America 28*, 120–128. Columbia University Press, New York, 1988.

[Kowalski, 1997] Gerald Kowalski. Information Retrieval Systems: Theory and Implementation. Kluwer Academic Publishers, Boston/Dordrecht/London, 1997.

[Mallchok, 2004] Friederike Mallchok. *Automatic Recognition of Organization Names in English Business News*. Doktorarbeit, LMU München, 2004.

[Paumier, 2004] Sébastien Paumier. *Manuel d'utilisation d'Unitex*, Université de Marne-la-Vallée, 2004.

[Senellart, 1998a] Jean Senellart. Locating noun phrases with finite state transducers. In *Proceedings of the 17th International Conference on Computational Linguistics*, 1212–1219. Montréal, Canada, 1998.

[Senellart, 1998b] Jean Senellart. Tools for locating noun phrases with finite state transducers. In *The computational treatment of nominals*. Proceedings of the Workshop, COLINGACL'98, 80–84. Montréal, Canada, 1998.

[9] Siehe annotiertes Testkorpus unter: http://www.cis.uni-muenchen.de/ sandrab/DA/IE-Korpus1.html

Erwartungsgesteuerte Informationsextraktion in Satzteilen

Peter Klügl[1], Martin Schuhmann[1], Frank Puppe[1], Hans-Peter Buscher[2]
[1]Universität Würzburg, Am Hubland, 97074 Würzburg, {pkluegl, schuhmann, puppe}@informatik.uni-wuerzburg.de
[2]Klinik für Innere Medizin II, DRK-Kliniken Berlin-Köpenick, h.buscher@drk-kliniken-koepenick.de

Abstract

Es wird ein Verfahren zur Informationsextraktion beschrieben und mit medizinischen Befundberichten evaluiert, das vorhandenes Wissen über zu erwartende Informationen in einem Text schon während des Prozesses der Informationsextraktion ausnutzt. Das Verfahren eignet sich insbesondere zum Einsatz für Kritiksysteme und fallbasierte Trainingssysteme, bei denen eine bekannte Problemlösung mit der Freitexteingabe eines Benutzers verglichen werden soll.

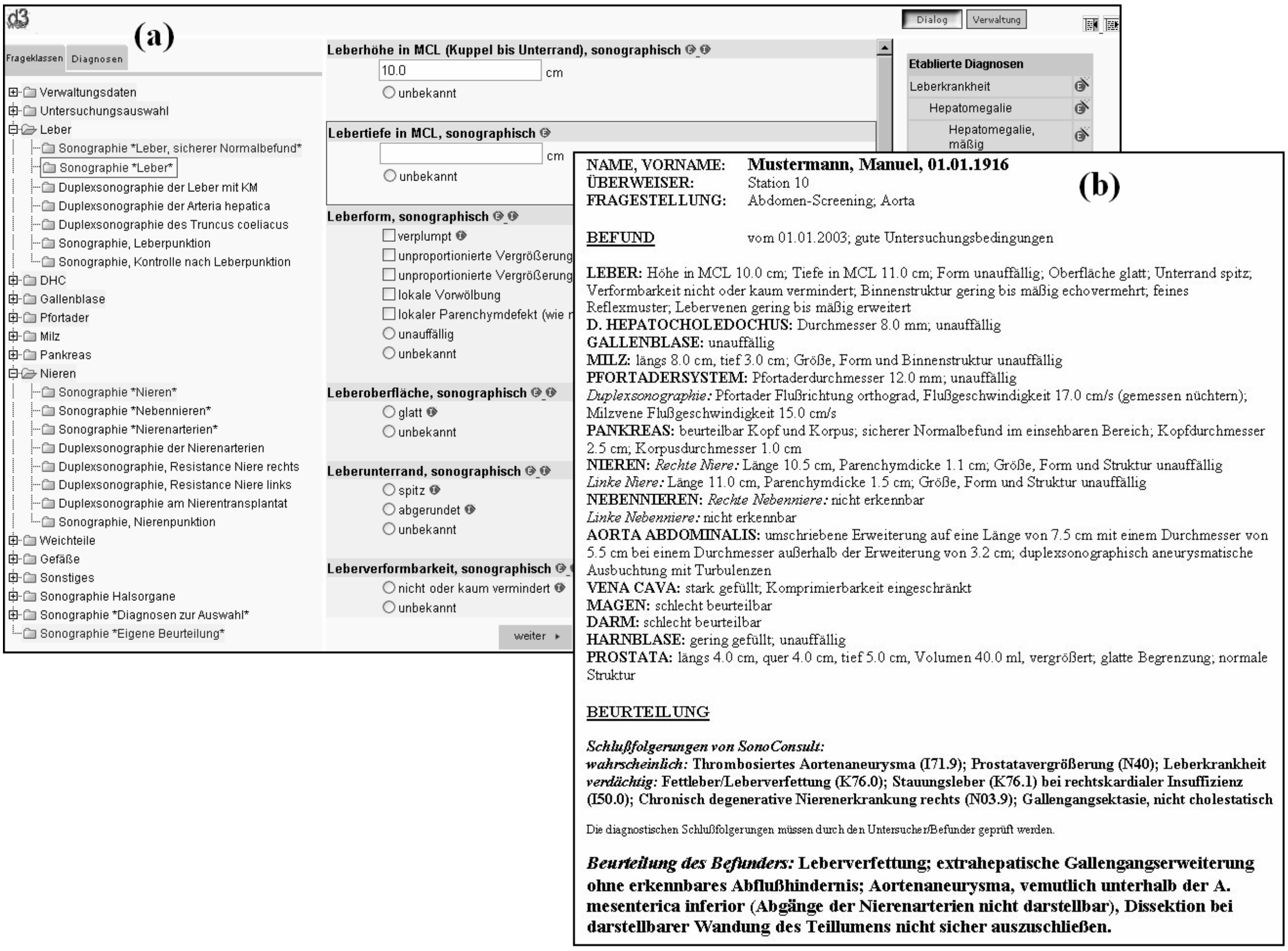
1 Einführung

In vielen Anwendungen liegen informelle und formalisierte Daten bzw. Wissenselemente zum gleichen Thema vor. Oft ist es wichtig, deren Konsistenz zu überprüfen. Die gezielte Suche nach formalisierten Informationen in informellen Dokumenten ist Gegenstand der erwartungsgesteuerten Informationsextraktion. Interessante Anwendungsgebiete sind z.B. Kritik- und fallbasierte Trainingssysteme, bei denen zu einem Problemfall eine bekannte, formal codierte Lösung mit einem Freitext des Benutzers verglichen werden soll. Falls Inkonsistenzen entdeckt werden, wird der Benutzer im Kritiksystem darauf hingewiesen bzw. im Trainingssystem bewertet. Als Anwendungsbeispiel dient in diesem Beitrag der Vergleich von automatisch hergeleiteten Diagnosen eines wissensbasierten Diagnose- und Dokumentationssystems (SonoConsult [Hüttig et al., 2004]) mit der freitextlichen Beurteilung durch einen Arzt. In Abb. 1 wird an einem Beispiel gezeigt, wie der Arzt seine sonographischen Befunde in dynamischen Formularen eingibt (linke Seite, (a)), daraus ein Befundbericht generiert wird (rechte Seite, (b)) und im Beurteilungsteil hergeleitete Diagnosen von SonoConsult mit der Beurteilung des Untersuchers verglichen werden kann ((b) unten). Da eine ungerechtfertigte, d.h. falsch negative Kritik/Bewertung vom Benutzer weniger akzeptiert wird als eine fehlende, d.h. falsch positive Kritik, sollte ein System nur dann eine Kritik äußern, wenn es sich ziemlich sicher ist. Zu diesem Zweck wurden Techniken der Informationsextraktion (IE) so erweitert, dass Erwartungen über vorgegebene Diagnosen bei der Analyse des Freitextes einfließen und auch bei Unsicherheiten eine Zuordnung der erwarteten Diagnosen zu entsprechenden Textpassagen angenommen wird. Diese weitgehende Heuristik basiert auf der Beobachtung, dass Untersucher sich vorsichtig ausdrücken und oft Diagnosen bewusst nur andeuten, wodurch ein exakter Match zu erwarteten Diagnosen unwahrscheinlicher wird. Der Ansatz ist einerseits deutlich einfacher als etwa das bekannte Morphosaurus-Projekt [Morphosaurus, 2007], da in diesem Ansatz außer einem Standard-Lexikon mit deutschen Wörtern nur kleinere sonographiespezifische Lexika verwenden werden (für Akronyme & Abkürzungen, Synonyme, Wortstämme, Präfixe & Suffixe und Kontexte) und daher nur ansatzweise eine lexikonbasierte Teilwortzerlegung in wesentlich geringerem Umfang wie in Morphosaurus durchgeführt wird. Andererseits werden zusätzliche Informationen genutzt (insbesondere die Segmentierung von Nominalphrasen in Befundberichten, die genaue Analyse von Negationen innerhalb der Segmente, die Kontextzuordnung der Segmente), die in Morphosaurus nicht benötigt werden.

Basierend auf Standardmethoden der Informationsextraktion (Kap. 2) werden wissensbasierte Techniken zur Kontexteinteilung von Segmenten und Diagnosen genutzt (Kap. 3), um eine grobe Zuordnung zu gewährleisten, die durch weitere Informationen verfeinert wird. In Kap. 4 werden zwei Fallstudien präsentiert, in denen zwei aufeinander aufbauende Systeme mit prospektiven Daten evaluiert werden. Eine Schwierigkeit besteht darin, dass Ärzte in der Sonographie häufig rotieren (z.B. alle 6 Monate) und daher ein System, das für die Beurteilungstexte eines Arztes gute Zuordnungsergebnisse liefert, nicht automatisch für Beurteilungstexte anderer Ärzte mit anderen Formulierungsgewohnheiten übertragbar ist. Ein weiteres Problem ist, dass Beurteilungstexte häufig in einem Telegrammstil, der hauptsächlich aus Nominalphrasen besteht, formuliert werden. Die vorläufigen Ergebnisse zeigen aber, dass die Kontexteinteilung von Segmenten relativ robust ist. Kap. 5 schließt mit einer Zusammenfassung und einem Ausblick.

2 Informationsextraktion

Das Ziel der IE ist „die Konstruktion von Systemen, die gezielt domänenspezifische Informationen aus freien Texten aufspüren und strukturieren können, bei gleichzeitigem Überlesen irrelevanter Informationen“ [Neumann, 2001, S.1]. Daher wird bei der IE im Gegensatz zu dem Ansatz des *Full Text Understanding* nicht versucht, den ganzen Text zu analysieren und zu verarbeiten, sondern nur relevante Informationen aus dem Text zu extrahieren. Hierfür muss zunächst festgelegt werden, welche Informationen als relevant angesehen werden. Dies geschieht durch die Definition sogenannter Templates, Attribut/Wert-Paaren, welche den Typ der semantischen Information bestimmen.


(a)
d3
Frageklassen
Diagnosen
Verwaltungsdaten
Untersuchungsauswahl
Leber
Sonographie *Leber, sicherer Normalbefund*
Sonographie *Leber*
Duplexsonographie der Leber mit KM
Duplexsonographie der Arteria hepatica
Duplexsonographie des Truncus coeliacus
Sonographie, Leberpunktion
Sonographie, Kontrolle nach Leberpunktion
DHC
Gallenblase
Pfortader
Milz
Pankreas
Nieren
Sonographie *Nieren*
Sonographie *Nebennieren*
Sonographie *Nierenarterien*
Duplexsonographie der Nierenarterien
Duplexsonographie, Resistance Niere rechts
Duplexsonographie, Resistance Niere links
Duplexsonographie am Nierentransplantat
Sonographie, Nierenpunktion
Weichteile
Gefäße
Sonstiges
Sonographie Halsorgane
Sonographie *Diagnosen zur Auswahl*
Sonographie *Eigene Beurteilung*
Dialog
Verwaltung
weiter ▸
(b)

Leberhöhe in MCL (Kuppel bis Unterrand), sonographisch
10.0 cm
○ unbekannt

Lebertiefe in MCL, sonographisch
cm
○ unbekannt

Leberform, sonographisch
□ verplumpt
□ unproportionierte Vergrößerung
□ unproportionierte Vergrößerung
□ lokale Vorwölbung
□ lokaler Parenchymdefekt (wie r
○ unauffällig
○ unbekannt

Leberoberfläche, sonographisch
○ glatt
○ unbekannt

Leberunterrand, sonographisch
○ spitz
○ abgerundet
○ unbekannt

Leberverformbarkeit, sonographisch
○ nicht oder kaum vermindert
○ unbekannt

Etablierte Diagnosen
Leberkrankheit
Hepatomegalie
Hepatomegalie, mäßig

NAME, VORNAME: **Mustermann, Manuel, 01.01.1916**
ÜBERWEISER: Station 10
FRAGESTELLUNG: Abdomen-Screening; Aorta

BEFUND vom 01.01.2003; gute Untersuchungsbedingungen

LEBER: Höhe in MCL 10.0 cm; Tiefe in MCL 11.0 cm; Form unauffällig; Oberfläche glatt; Unterrand spitz; Verformbarkeit nicht oder kaum vermindert; Binnenstruktur gering bis mäßig echovermehrt; feines Reflexmuster; Lebervenen gering bis mäßig erweitert
D. HEPATOCHOLEDOCHUS: Durchmesser 8.0 mm; unauffällig
GALLENBLASE: unauffällig
MILZ: längs 8.0 cm, tief 3.0 cm; Größe, Form und Binnenstruktur unauffällig
PFORTADERSYSTEM: Pfortaderdurchmesser 12.0 mm; unauffällig
Duplexsonographie: Pfortader Flußrichtung orthograd, Flußgeschwindigkeit 17.0 cm/s (gemessen nüchtern); Milzvene Flußgeschwindigkeit 15.0 cm/s
PANKREAS: beurteilbar Kopf und Korpus; sicherer Normalbefund im einsehbaren Bereich; Kopfdurchmesser 2.5 cm; Korpusdurchmesser 1.0 cm
NIEREN: *Rechte Niere:* Länge 10.5 cm, Parenchymdicke 1.1 cm; Größe, Form und Struktur unauffällig
Linke Niere: Länge 11.0 cm, Parenchymdicke 1.5 cm; Größe, Form und Struktur unauffällig
NEBENNIEREN: *Rechte Nebenniere:* nicht erkennbar
Linke Nebenniere: nicht erkennbar
AORTA ABDOMINALIS: umschriebene Erweiterung auf eine Länge von 7.5 cm mit einem Durchmesser von 5.5 cm bei einem Durchmesser außerhalb der Erweiterung von 3.2 cm; duplexsonographisch aneurysmatische Ausbuchtung mit Turbulenzen
VENA CAVA: stark gefüllt; Komprimierbarkeit eingeschränkt
MAGEN: schlecht beurteilbar
DARM: schlecht beurteilbar
HARNBLASE: gering gefüllt; unauffällig
PROSTATA: längs 4.0 cm, quer 4.0 cm, tief 5.0 cm, Volumen 40.0 ml, vergrößert; glatte Begrenzung; normale Struktur

BEURTEILUNG

Schlußfolgerungen von SonoConsult:
wahrscheinlich: **Thrombosiertes Aortenaneurysma (I71.9); Prostatavergrößerung (N40); Leberkrankheit**
verdächtig: **Fettleber/Leberverfettung (K76.0); Stauungsleber (K76.1) bei rechtskardialer Insuffizienz (I50.0); Chronisch degenerative Nierenerkrankung rechts (N03.9); Gallengangsektasie, nicht cholestatisch**

Die diagnostischen Schlußfolgerungen müssen durch den Untersucher/Befunder geprüft werden.

Beurteilung des Befunders: **Leberverfettung; extrahepatische Gallengangserweiterung ohne erkennbares Abflußhindernis; Aortenaneurysma, vemutlich unterhalb der A. mesenterica inferior (Abgänge der Nierenarterien nicht darstellbar), Dissektion bei darstellbarer Wandung des Teillumens nicht sicher auszuschließen.**

Abbildung 1: Befundeingabe mit SonoConsult: (a) Dialog mit dem SonoConsult-System (b) Automatisch erstellter Befundbericht mit den ermittelten Diagnosen und der manuell eingegeben Beurteilung des Untersuchers

Ein IE-System erhält als Eingabe eine Menge von Freitexten und Templates, welche nur die zu suchenden Attribute enthalten. Der Extraktionsprozess instanziiert die Templates, indem den Attributen die jeweiligen extrahierten Werte zugewiesen werden. Diese instanziierten Templates bilden die Ausgabe eines IE-Systems und können dann weiter verarbeitet werden.

Der IE-Prozess kann in vier Phasen eingeteilt werden (vgl. u.a. [Feldman und Sanger, 2007]:

1. Tokenisierung
2. Morphologische und lexikalische Analyse
3. Syntaktische Analyse
4. Domänenspezifische Analyse

Die Tokenisierung teilt zunächst die Eingabedokumente anhand ihrer Textstruktur auf (*structural information extraction*). Die daraus entstandenen Textfragmente werden mit Hilfe der Satzzeichen und Leerzeichen weiter in Sätze und einzelne Wörter (Tokens) zerlegt.

Diese einzelnen Tokens werden in der morphologischen und lexikalischen Analyse genauer betrachtet. Meist werden sie zunächst mit den Einträgen eines Lexikons oder mit einer Gazetteer abgeglichen. Anschließend wird deren Flexion, Wortart (Part of Speech, POS) und Wortstamm (Stemming) bestimmt. Weiterhin müssen noch Komposita und Hyphenkonstrukte (z.B. An- und Verkauf) zerlegt, sowie Abkürzungen ersetzt werden. Oft werden in dieser Phase Eigennamen erkannt.

Die syntaktische Analyse versucht daraufhin die Struktur und den Aufbau der einzelnen Sätze zu bestimmen. Hierbei kann zwischen einer vollständigen (full parsing) und einer flachen Analyse (shallow parsing) unterschieden werden. Da eine vollständige Analyse meist sehr aufwändig ist, beschränken sich viele IE-Systeme auf eine flache Analyse. In diesem Fall wird versucht, die Satzstruktur mit Hilfe der Ergebnisse der vorherigen Phasen zu ermitteln.

Abschließend werden in der letzten Phase, der domänenspezifischen Analyse, die Koreferenzen aufgelöst und die Templates mit Inhalt gefüllt.

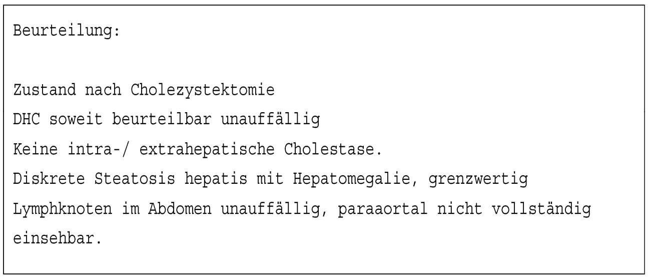
3 Erwartungsgesteuerte Informationsextraktion in Satzteilen

Die erwartungsgesteuerte Informationsextraktion in Satzteilen unterscheidet sich in zwei Bereichen von der normalen Vorgehensweise. Wie der Begriff bereits andeutet, fließen in diesem Verfahren die Erwartungen an das Ergebnis der Extraktion bereits in den Prozess mit ein. Weiterhin werden keine grammatikalisch korrekten Sätze als Eingabe erwartet. Diese Unterschiede sollen kurz erläutert werden.

3.1 Abgrenzung

Ausgehend von den gegebenen Templates werden bei der IE Informationen aus den Freitexten extrahiert. Hierbei wird jedoch nur der semantische Typ der Information betrachtet, welcher durch die Attribute der Templates bestimmt wird. In vielen Fällen wird der Extraktionsprozess mit gewissen Erwartungen über den Inhalt der Freitexte durchgeführt. Diese Erwartungen können allerdings erst nach der Erstellung der instanziierten Templates überprüft werden. In dieser Arbeit wird daher ein neuer Ansatz vorgestellt, wie die IE zu einer erwartungsgesteuerten IE erweitert werden kann, so dass die Erwartungen an den Inhalt der Texte in den Extraktionsprozess einfließen und in Folge dessen die Ergebnisse des Verfahrens deutlich verbessert werden.

Ein weiterer Unterschied besteht in der Qualität der Eingabe. Oft bestehen die vorgegebenen Freitexte nicht aus grammatikalisch korrekt formulierten Sätzen, sondern aus Satzteilen, insbesondere Nominalphrasen. Der syntaktische Aufbau des Textes kann unter diesen Voraussetzungen zwar erkannt werden, die daraus resultierende Satzstruktur eignet sich jedoch nur eingeschränkt zur Extraktion einer Information. In dieser Arbeit dienen vor allem medizinische Befundberichte als Eingabe für die erwartungsgesteuerte IE (vgl. Abb. 1). Die Texte unterscheiden sich abhängig von dem Untersucher sehr stark in ihrer grammatikalischen Struktur. Oft werden Nominalphrasen benutzt, um die Ergebnisse der Untersuchung zu beschreiben. Folglich ist die Bedeutung weniger in den Verben der Sätze, als vielmehr in einer Aufzählung von Konzepten zu finden. Zusätzlich zu den üblichen linguistischen Schwierigkeiten wie der Flexion kommen weitere Probleme hinzu. Hierzu gehören unter anderem Ambiguitätsprobleme, heterogene Wortstämme, unsystematische Wortmodifikationen bei Komposita und häufige Verwendung von teilweise ad hoc gebildeten Akronymen. Wegen der nicht immer einheitlichen Verwendung von Satzzeichen ist auch die Segmentierung ein Problem (Abb. 2 zeigt einen Rohtext, Abb. 3 die korrekte Segmentierung und Abb. 6 eine fehlerhafte Segmentierung gemäß dem Satzzeichen “.“).



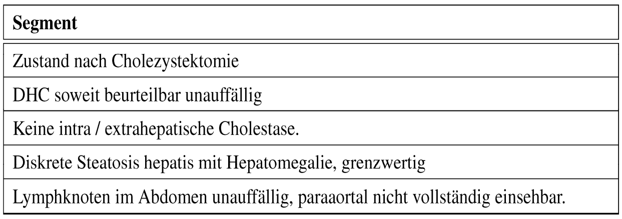
```
Beurteilung:

Zustand nach Cholezystektomie
DHC soweit beurteilbar unauffällig
Keine intra-/ extrahepatische Cholestase.
Diskrete Steatosis hepatis mit Hepatomegalie, grenzwertig
Lymphknoten im Abdomen unauffällig, paraaortal nicht vollständig
einsehbar.
```

Abbildung 2: Beurteilung eines Untersuchers (vgl. Abb. 1)

3.2 Methoden

Die in Kap. 2 vorgestellten Standardverfahren der vier Phasen werden für eine Vorverarbeitung der Texte benutzt. Die Tokenisierung in dieser Arbeit verwendet reguläre Ausdrücke, um den gegebenen Text in einzelne Token zu unterteilen. Die Abkürzungen und Synonyme werden mit manuell erstellten Listen aufgelöst. Das Stemming basiert auf dem Algorithmus von Jörg Caumanns [Caumanns, 1999] und für das Part-of-Speech Tagging wurde der TreeTagger der Universität Stuttgart (u.a. [Schmid, 1995]) verwendet. Weiterhin unterstützt ein manuell erstelltes, domänenspezifisches Lexikon die Zerlegung der Komposita. Neben diesen Standardverfahren werden zwei weitere Phasen eingeführt: die Segmentierung und die Kontexterkennung.

Zwar ist der Inhalt und der Typ der zu extrahierenden Information bekannt, aber nicht deren Aussehen, bzw. Auftreten. Die Segmentierung zerlegt den Freitext in einzelne inhaltlich zusammenhängende Textpassagen, damit die erwarteten Informationen den erstellten Segmenten zugeordnet werden können. Aufgrund der grammatikalisch unkorrekten Teilsätze und der häufig verwendeten Nominalphrasen kann dies jedoch nicht nur mit Hilfe einer Aufteilung bezüglich Satzzeichen geschehen. Für eine bestmögliche Segmentierung sollte die Satzstruktur der syntaktischen Analyse in diese Phase miteinbezogen werden. Im Rahmen dieser Arbeit wurde dies jedoch noch nicht realisiert, so dass die Segmentierung zum jetzigen Zeitpunkt auf Heuristiken für Satzzeichen und Zeilenumbrüche aufbaut und somit zwischen der Tokenisierung und der morphologischen Analyse einzuordnen ist. Ein Beispiel für eine korrekte Segmentierung des Beispiels aus Abb. 2 kann der Abb. 3 entnommen werden.

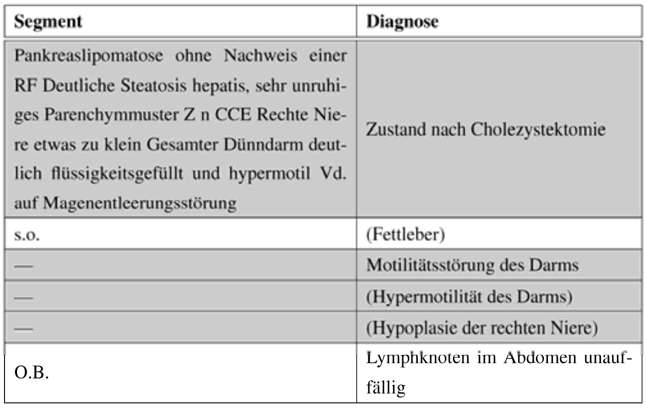


Segment
Zustand nach Cholezystektomie
DHC soweit beurteilbar unauffällig
Keine intra / extrahepatische Cholestase.
Diskrete Steatosis hepatis mit Hepatomegalie, grenzwertig
Lymphknoten im Abdomen unauffällig, paraaortal nicht vollständig einsehbar.

Abbildung 3: Segmentierung des Beispiels aus Abb. 2

Als eine weitere Phase wurde die Kontexterkennung hinzugefügt. Die relevanten Wörter eines Segments werden mit einem annotierten Lexikon für Kontexte abgeglichen. Die auf diese Weise ermittelten domänenspezifischen Kontexte werden daraufhin dem jeweiligen Segment zugewiesen. Die Erkennung relevanter Begriffe ist im Rahmen dieser Arbeit, der medizinischen Befundbeurteilung, äquivalent zur Erkennung der Eigennamen. Falls ein komplexes Kompositum nicht erkannt werden kann, d.h. im Lexikon nicht vertreten ist, wird dieses in seine Teilworte zerlegt, welche daraufhin mit dem annotierten Lexikon abgeglichen werden. Durch diese Aufteilung besteht eine erhöhte Wahrscheinlichkeit, einen passenden Kontext zu ermitteln. Im Rahmen dieser Arbeit beziehen sich die Kontexte meist auf das Organ der zu überprüfenden Diagnose. Eine Übersicht der Organe der Abdomen-Sonographie findet sich in Abb. 1 (b) (Organe, bzw. Kontexte in Großbuchstaben und fett gedruckt). Da versucht wird, jedem Segment einen möglichst genauen Kontext zuzuweisen, ist die korrekte Einteilung in Segmente eine wichtige Voraussetzung für die Kontexterkennung. Abbildungen 4 und 5 enthalten beispielhafte Zuordnungen des jeweiligen Segmentes mit der gegebenen Diagnose. In Abb. 4 schlug eine korrekte Segmentierung fehl und somit konnten keine passenden Kontexte zugeordnet werden. In Abb. 5 hingegen wurde durch eine korrekte Segmentierung die Grundlage für die Kontexterkennung geschaffen.

Dem fünften Segment konnte daher zum Beispiel der Kontext *Darm* zugeordnet werden.



Segment	Diagnose
Pankreaslipomatose ohne Nachweis einer RF Deutliche Steatosis hepatis, sehr unruhiges Parenchymmuster Z n CCE Rechte Niere etwas zu klein Gesamter Dünndarm deutlich flüssigkeitsgefüllt und hypermotil Vd. auf Magenentleerungsstörung	Zustand nach Cholezystektomie
s.o.	(Fettleber)
—	Motilitätsstörung des Darms
—	(Hypermotilität des Darms)
—	(Hypoplasie der rechten Niere)
O.B.	Lymphknoten im Abdomen unauffällig

Abbildung 4: Fehlerhafte Zuordnung auf Grund einer unpassenden Segmentierung (fehlerhafte Zuordnungen sind grau hinterlegt, „-“ repräsentiert eine fehlende Zuordnung, bzw. ein fehlendes Segment)

Segment	Diagnose	Kontext
Pankreaslipomatose ohne Nachweis einer RF	—	Pankreas
Deutliche Steatosis hepatis, sehr unruhiges Parenchymmuster	(Fettleber)	Leber
Z n CCE	Zustand nach Cholezystektomie	Gallenblase
Rechte Niere etwas zu klein	(Hypoplasie der rechten Niere)	Niere
Gesamter Dünndarm deutlich flüssigkeitsgefüllt und hypermotil	Motilitätsstörung des Darms	Darm
s.o.	(Hypermotilität des Darms)	Darm
Vd. auf Magenentleerungsstörung	—	Magen
O.B.	Lymphknoten im Abdomen unauffällig	Lymphknoten

Abbildung 5: Korrekte Zuordnung auf Grund passender Segmente und Kontexte

Für die eigentliche Extraktion werden folgende Methoden benutzt:

1. **Schablonen:** Die Schablonen (Templates) dienen vor allem zur Erkennung von Verneinungen und dem Bereich im Satz, auf welchen sich die Verneinung bezieht. Zum Beispiel betrifft die Verneinung in Abb. 1 (b) (*extrahepatische Gallengangserweiterung ohne erkennbares Ausflußhindernis*) mit dem Schlüsselwort *ohne* nur das *erkennbare Ausflußhindernis* und nicht die *extrahepatische Gallengangserweiterung*. Die Schablonen vermeiden somit vor allem zukünftige Fehler der nachfolgenden Methoden und werden mit regulären Ausdrücken umgesetzt.
2. **Token- und Silbenzuordnung:** Hierbei werden die einzelnen Tokens, bzw. Silben eines Segments mit der erwarteten Information verglichen. Ein Abgleich der Wortstämme, Wortarten und Flexion bestimmt die Sicherheit der Zuordnung.
3. **Ontologische Zuordnung:** An diesen Punkt werden nur noch Informationen und Segmente betrachtet, welche noch nicht zugeordnet werden konnten. Die ontologische Zuordnung erweitert die Tokenzuordnung mit Hilfe von ontologischem Domänenwissen, welches in Form von manuell erstellten Regeln vorliegt. Anhand dieses Domänenwissens kann zum Beispiel die erwartete Information *Leberveränderung* mit dem Segment *Leberverkleinerung* in Verbindung gebracht werden, da eine Verkleinerung eine Art von Veränderung darstellt.
4. **Kontext-Zuordnung:** Hierbei werden zunächst die Kontexte für jedes Segment und für jede Diagnose mit Hilfe eines manuell erstellten Lexikons bestimmt. Dieses domänenspezifische Lexikon enthält die Zuordnung üblicher Begriffe zu ihren jeweiligen Kontexten. Für jede Kontextkategorie (Organ) wird überprüft, wie viele Segmente, bzw. Diagnosen darin enthalten sind. Falls in einer Kontextkategorie nur ein Segment und eine Diagnose entdeckt werden, spricht dies stark für eine Zuordnung von Segment und Diagnose. Ansonsten müssen weitere Informationen zur Auflösung der Mehrdeutigkeit hinzugezogen werden. Die meisten Diagnosen haben nur einen Kontext, es gibt aber auch Ausnahmen. Die Diagnose *Hepatosplenomegalie* besitzt beispielsweise die Kontexte *Leber* (Hepa) und *Milz* (spleno).
5. **Fuzzy-Match:** Zuletzt werden Teilworte, welche einem Kontext zugeordnet werden können, aus der erwarteten Information entfernt und deren restliche Teilworte mit den noch nicht zugewiesenen Segmenten verglichen. Dabei wird die Ähnlichkeit der Kontexte und der restlichen Silben, bzw. Teilworte, zu den übrigen Segmenten bestimmt. Zum Beispiel werden der Kontext *Pankreas* und die Teilwörter *pseudo* und *zyste* der Information *Pankreaspseudozyste* mit den noch nicht zugewiesenen Segmenten verglichen.

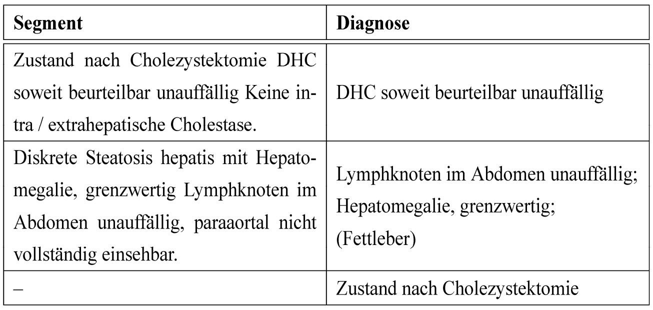
Die Erwartungen an die relevanten Informationen fließen in den verschiedenen Methoden auf unterschiedliche Weise ein: Die Tokenzuordnung, die Silbenzuordnung und der Fuzzy Match vergleichen die Buchstabenkombinationen des erwarteten Inhalts mit den vorliegenden Texten, bzw. Segmenten. Die Schablonen benutzen zusätzlich Informationen über die Satzstruktur. Daher unterstützen diese Methoden die IE vor allem abhängig vom Auftreten der erwarteten Information, bzw. deren linguistischen Vorkommen. Die hierarchische und ontologische Zuordnung hingegen benutzen beide Domänenwissen und kombinieren dieses mit dem erwarteten Inhalt, um innerhalb des gegebenen Kontexts eine korrekte Zuordnung vorzunehmen. Daher wirkt in diesem Fall die domänenspezifische Bedeutung der erwarteten Information unterstützend. Infolgedessen kann mit diesen Methoden nicht nur der erwartete Typ der zu extrahierenden Information, sondern auch dessen Inhalt den Extraktionsprozess verbessern.

3.3 Kategorisierung der Fehler

Für eine genauere Evaluation der erwartungsgesteuerten IE können die entstandenen Fehler neben der üblichen Einteilung in Fehler 1. Art und Fehler 2. Art weiter unterteilt werden. Im Rahmen dieser Arbeit wurden neun unterschiedliche Fehlertypen kategorisiert:

1. **Segmentierungsfehler:** Die heuristische Einteilung in unterschiedliche Segmente war fehlerhaft.
2. **Folgefehler der Segmentierung:** Ein Segmentierungsfehler verursachte weitere Fehler bei der Zuordnung.
3. **Verneinungsfehler:** Eine Verneinung wurde durch die Schablonen falsch interpretiert oder nicht erkannt.
4. **Heuristik - falsch positiv:** Es wurde eine fehlerhafte Zuordnung erzeugt.
5. **Heuristik - falsch negativ:** Eine erwartete Information wurde fälschlicherweise nicht zugeordnet.
6. **Fehlende Synonyme:** Eine erwartete Information konnte aufgrund fehlender Synonyme nicht gefunden werden (fehlendes ontologisches Wissen).
7. **Rechtschreibfehler:** Ein Rechtschreibfehler oder Tippfehler in der Beurteilung des Befundes verursachte eine fehlerhafte Zuordnung.
8. **Hierarchiefehler:** Die Kontext-Zuordnung führte zu einem Fehler.
9. **O.B./B.A. Fehler:** Ein Segment wurden fälschlicherweise nicht als *ohne Befund* (O.B.) oder als *Behandlungsanweisung* (B.A.) erkannt.

Abb. 6 zeigt zum Beispiel einen Segmentierungsfehler, welcher eine fehlerhafte Zuordnung zur Folge hat. Wenn das System eine Verneinung erkennt und diese auf das ganze Segment bezieht, wird der *Zustand nach Cholezystektomie* fälschlicherweise ebenfalls verneint und daher nicht zugeordnet.



Segment	Diagnose
Zustand nach Cholezystektomie DHC soweit beurteilbar unauffällig Keine intra / extrahepatische Cholestase.	DHC soweit beurteilbar unauffällig

Diskrete Steatosis hepatis mit Hepatomegalie, grenzwertig Lymphknoten im Abdomen unauffällig, paraaortal nicht vollständig einsehbar.	Lymphknoten im Abdomen unauffällig;
	Hepatomegalie, grenzwertig;
	(Fettleber)

–	Zustand nach Cholezystektomie

Abbildung 6: Beispiel einer fehlerhaften Segmentierung der Befundbeurteilung aus Abb. 2 und dem daraus resultierenden Folgefehler

4 Fallstudien

Für eine Evaluation des beschriebenen erwartungsgesteuerten Verfahrens wurden zwei Fallstudien durchgeführt. Die erste Fallstudie beruhte auf einem IE-System, welches im Rahmen einer Diplomarbeit [Dressler, 2005] erstellt wurde. Darin wurde die skizzierte Grundstruktur realisiert. Die Kontexterkennung wurde nur mit einem relativ kleinen annotierten Lexikon vorgenommen, weswegen die Kontext-Zuordnung kaum benutzt wurde. Die Segmentierung beruhte nur auf einer einfachen Zerlegung der Sätze abhängig von den vorkommenden Satzzeichen. Negationen wurden auf das ganze Segment bezogen. In der zweiten Fallstudie wurde eine Weiterentwicklung des ersten Systems verwendet, welches im Rahmen einer weiteren Diplomarbeit [Braun, 2006] erstellt wurde. Dieses System verwendete ein größeres Lexikon und konnte die Kontexte der zu extrahierenden Information besser bestimmen. Eine Negation wurde mit durch reguläre Ausdrücke dargestellten Templates auf Satzteile der Segmente bezogen. Weiterhin wurden die Segmentierung mit Hilfe von zusätzlichen Heuristiken ausgebaut und verschiedene kleinere Fehler des ersten Systems ausgebessert. Insgesamt wurden für die Evaluation der Systeme Befundberichte aus drei unterschiedlichen Zeiträumen verwendet, welche jeweils durch einen anderen Untersucher erstellt wurden (Datensatz A: 1.Quartal 2005, Datensatz B: 3.Quartal 2005 und Datensatz C: 1.Quartal 2006). Für die Berechnung der Präzision (P), der Vollständigkeit (V) und des F_1-Maßes wurden folgende Formeln benutzt, wobei stets β=*1* gesetzt wurde:

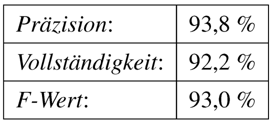
$$P = \frac{richtige\,Zuordnungen}{richtige\,Zuordnungen + Fehler\,1.Art} \tag{1}$$

$$V = \frac{richtige\,Zuordnungen}{richtige\,Zuordnungen + Fehler\,2.Art} \tag{2}$$

$$F_1 = \frac{(\beta^2 + 1) \cdot P \cdot V}{\beta^2 \cdot P + V} \tag{3}$$

4.1 Ergebnisse der ersten Fallstudie

Das erste System wurde mit Hilfe von 30 Befundberichten aus Datensatz A entwickelt und einer prospektiven Evaluation mit 220 neuen Befundberichten aus dem gleichen Datensatz evaluiert. In der Auswertung erreicht das erste System eine durchschnittliche Präzision und Vollständigkeit von über 90% (vgl. Abb. 7). Es stellte sich jedoch heraus, dass diese guten Evaluationsergebnisse darauf beruhten, dass der Datensatz A von einem einzelnen Untersucher erstellt wurde. Zu Beginn der Entwicklung des zweiten Systems wurden die Datensätze B und C verfügbar, so dass eine weitere Evaluation des ersten Systems mit neuen Befundberichten durchgeführt werden konnte. Die Ergebnisse dieser Evaluation zeigt Abb. 8 (es wurden jeweils 30 Befundberichte aus den zwei neuen Datensätzen verwendet). Der durchschnittliche F-Wert sank auf ca. 83%. Abb. 9 bietet eine Übersicht über die Art und Häufigkeit der dabei aufgetretenen Fehler und verdeutlicht die Schwächen des Systems bei fehlenden Synonymen, Diagnose-Hierarchie-Fehlern und O.B./B.A.-Fehlern.



Präzision:	93,8 %
Vollständigkeit:	92,2 %
F-Wert:	93,0 %

Abbildung 7: Prospektive Evaluation des ersten Systems mit 220 Befundberichten aus Datensatz A

	Datensatz B	*Datensatz C*	*Gesamt*
Richtige Zuordnungen:	151	152	303
Fehler 1. Art:	11	14	25
Fehler 2. Art:	53	42	95
Zuordnungen gesamt:	215	208	423
Komplett fehlerlose Fälle:	9 / 30	9 / 30	18 / 60
Präzision:	93,2 %	91,6 %	92,4 %
Vollständigkeit:	74,0 %	78,4 %	76,2 %
F-Wert:	82,5 %	84,4 %	83,45 %

Abbildung 8: Auswertung des ersten Systems mit neuen Datensätzen

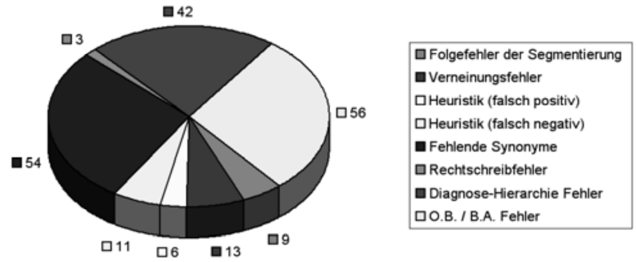


Abbildung 9: Einteilung der beobachteten Fehler des ersten Systems (auf umfassenderer Datenquelle als Abb. 8). Es wurden neun Folgefehler der Segmentierung beobachtet, übrige Fehler folgen im Uhrzeigersinn

4.2 Ergebnisse der zweiten Fallstudie

Das zweite System wurde mit jeweils 30 Befundberichten aus den drei Datensätzen entwickelt und retrospektiv evaluiert. Wie Abb. 10 verdeutlicht, erreichte dieses System einen F-Wert von 97%. Daraufhin wurde eine prospektive Evaluation durchgeführt, in welcher jeweils 50 neue Befundberichte aus den drei Datensätzen herangezogen wurden. Abb. 11 bietet einen Überblick über die Ergebnisse dieser Auswertung. Wie man erkennen kann, verschlechterten sich die einzelnen Werte nur geringfügig, so dass ein F-Wert von durchschnittlich über 96% erreicht werden konnte. In Abb. 12 kann man die Art und Häufigkeit der dabei aufgetretenen Fehler erkennen. Abb. 13 enthält schließlich eine Übersicht über die beobachteten Fehler beider Systeme.

	Datensatz A	*Datensatz B*	*Datensatz C*	*Gesamt*
Richtige Zuordnungen:	285	204	204	693
Fehler 1. Art:	6	6	1	13
Fehler 2. Art:	6	13	5	24
Zuordnungen gesamt:	297	223	210	730
Komplett fehlerlose Fälle:	23 / 30	18 / 30	26 / 30	67 / 90
Präzision:	97,9 %	97,1 %	99,5 %	98,2 %
Vollständigkeit:	97,9 %	94,0 %	97,6 %	96,7 %
F-Wert:	97,9 %	95,6 %	98,6 %	97,4 %

Abbildung 10: Retrospektive Evaluation des zweiten Systems

	Datensatz A	*Datensatz B*	*Datensatz C*	*Gesamt*
Richtige Zuordnungen:	505	404	419	1328
Fehler 1. Art:	18	5	8	31
Fehler 2. Art:	18	31	17	66
Zuordnungen gesamt:	541	440	444	1425
Komplett fehlerlose Fälle:	26 / 50	29 / 50	33 / 50	88 / 150
Präzision:	96,6 %	98,8 %	98,1 %	97,7 %
Vollständigkeit:	96,6 %	92,9 %	96,1 %	95,3 %
F-Wert:	96,6 %	95,7 %	97,1 %	96,5 %

Abbildung 11: Prospektive Evaluation des zweiten Systems

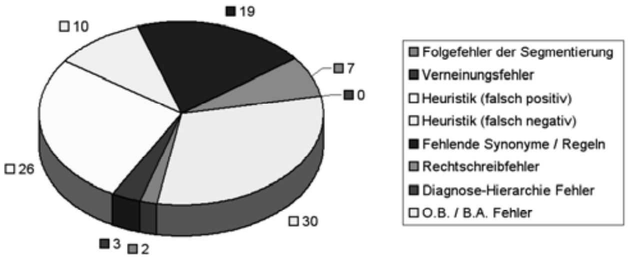


Abbildung 12: Einteilung der beobachteten Fehler des zweiten Systems. Es wurden zwei Folgefehler der Segmentierung beobachtet, übrige Fehler folgen im Uhrzeigersinn

4.3 Diskussion

Wie Abb. 13 aufzeigt, konnte die Treffergenauigkeit deutlich erhöht werden. Hervorzuheben sind in diesem Zusammenhang vor allem die Segmentierungsfehler, Verneinungsfehler, die Diagnose-Hierarchie-Fehler, d.h. Kontextfehler, und O.B./B.A. Fehler, welche durch die Erweiterungen des zweiten System deutlich reduziert werden konnten. Da die Evaluierungsergebnisse stark von dem behandelnden Untersucher abhängen, kann eine Verschlechterung der Trefferquote bei neuen Untersuchern nicht ausgeschlossen werden. Vor einem praktischen Einsatz des Systems müssen daher noch weitere Auswertungen mit anderen Untersuchern vorgenommen werden. In diesem Fall sollte eine Korpora-Analyse mit vorhandenen Befundberichten dieses Untersuchers verwendet werden, um eventuelle Probleme mit der Terminologie des neuen Untersuchers frühzeitig zu erkennen.

5 Zusammenfassung und Ausblick

Diese Arbeit stellte eine erwartungsgesteuerte Informationsextraktion in Satzteilen vor. Zunächst wurde kurz auf die allgemeine Informationsextraktion eingegangen und die verbreitete Vorgehensweise vorgestellt. Daraufhin wurde ein neuer Ansatz skizziert, wie die Erwartungen an die extrahierten Informationen direkt in den Extraktionsprozess mit einfließen können. In den Fallstudien wurden zwei Systeme miteinander verglichen, wobei die Auswertungen zeigten, dass die Verbesserungen bezüglich der erwartungsgesteuerten Informationsextraktion eine über 10% höhere Treffergenauigkeit bewirken.

Die Ergebnisse des erwartungsgesteuerten Systems können durchaus als gut bezeichnet werden, obwohl die Verallgemeinerungsfähigkeit bei verschiedenen Untersu-

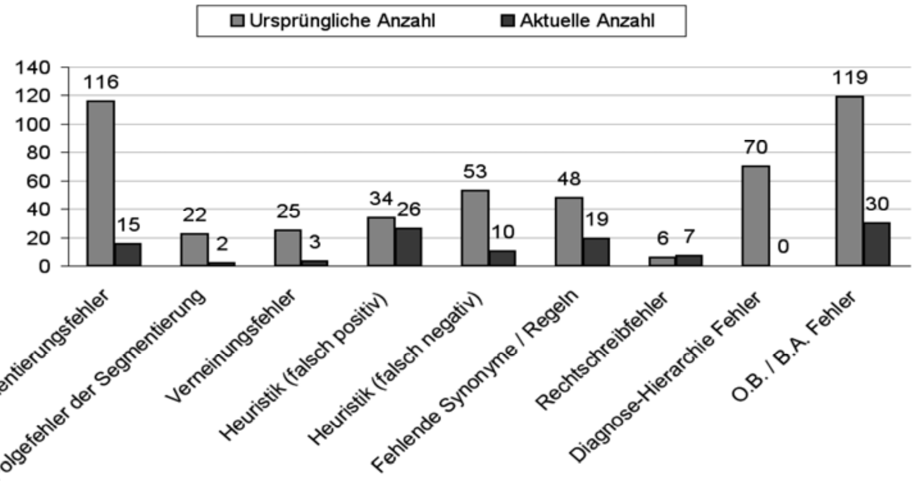


Abbildung 13: Vergleich der aufgetretenen Fehler in 1. und 2. System

chern für konkrete Anwendungen noch erhöht werden muss. Insbesondere müssen Evaluationen mit weiteren Untersuchern durchgeführt werden, da ihr Sprachgebrauch erheblich variieren kann. Der Einsatz großer medizinspezifischer Lexika mit standardisierten Subwortlisten wie in Morphosaurus erscheint vielversprechend, da er auch eine verbesserte Kontexterkennung ermöglicht. Ein Einsatz des Systems ist in verschiedenen Anwendungen angedacht. Zunächst wäre hierzu ein Kritiksystem für SonoConsult zu nennen (vgl. [Puppe, 1999]). Mit Hilfe der erwartungsgesteuerten Informationsextraktion und des Expertensystems SonoConsult können die manuell eingegebenen Befundberichte überprüft werden. Falls Abweichungen zwischen den ermittelten und eingegebenen Diagnosen vorliegen, kann der Untersucher auf fehlende oder fehlerhafte Diagnosen hingewiesen werden. Das System kann jedoch auch in anderen Bereichen benutzt werden. Beispielhaft sei hier eLearning genannt: Dem Tutanden werden beim eLearning oft Multiple-Choice- oder Long-Menu-Fragen gestellt, um dessen Lernerfolg einzuschätzen und zu überprüfen. In manchen Lernbereichen wie z.B. juristischen Fallbearbeitungen eignet sich eine textuelle Bearbeitung der Aufgabenstellung aber weitaus besser. Die eingegebenen Lösungen der Tutanden könnten mit Hilfe dieses erwartungsgesteuerten Systems überprüft und vorkorrigiert werden, was eine manuelle Nachkorrektur zwar nicht ersetzen, aber wesentlich erleichtern kann. Als eine weitere Anwendung kann auch eine Terminologieextraktion, bzw. deren Überprüfung angedacht werden.

Literaturverzeichnis

[Braun, 2006] Christian Braun. Informationsextraktion zur erwartungsgesteuerten Diagnosecodierung. Diplomarbeit, Universität Würzburg, 2006.

[Caumanns, 1999] Jörg Caumanns. A Fast and Simple Stemming Algorithm for German Words. Technical report, Center für Digitale Systeme, Freie Universität Berlin.

[Dressler, 2005] Alexander Dressler. Diagnosecodierung medizinischer Freitexte am Beispiel sonographischer Befundberichte. Diplomarbeit, Universität Würzburg, 2005.

[Feldman und Sanger, 2007] Ronen Feldman und James Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambrige University Press, 2007.

[Hüttig et al., 2004] Matthias Hüttig, Georg Buscher, Thomas Menzel, Wolfgang Scheppach, Frank Puppe, Hans-Peter Buscher. A Diagnostic Expert System for Structured Reports, Quality Assessment, and Training of Residents in Sonography. *Medizinische Klinik*, 3:17-22, 2004.

[Morphosaurus, 2007] http://www.morphosaurus.de (29.8.07) Kategorie „project information“; hier finden sich auch zahlreiche Publikationen zum Projekt; die nicht einzeln zitiert werden.

[Neumann, 2001] Günter Neumann. Informationsextraktion. In Carstensen et al, Editor, *Computerlinguistik und Sprachtechnologie - Eine Einführung*, Spektrum Akademischer Verlag, Heidelberg.

[Puppe, 1999] Frank Puppe. Meta Knowledge for Extending Diagnostic Consultation to Critique Systems. In *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW)*, 367-372. Berlin, 1999. Springer.

[Schmid, 1995] Helmut Schmid. Improvements in part-of-speech tagging with an application to German. In Feldweg und Hinrichs, Editor, *Lexikon und Text*, 47-50. Niemeyer, Tübingen. 1995.

Integration possibilities of a context-based search engine into a project planning portal in the mechanical engineering domain

Raiko Eckstein, Andreas Henrich
University of Bamberg
Chair of Media Informatics
D-96052 Bamberg, Germany
{raiko.eckstein, andreas.henrich}@wiai.uni-bamberg.de

Abstract

Product development processes in the mechanical engineering field are getting more and more complex while the requirements of cost and time reduction increase. Thus, there is an increasing necessity to support engineering designers by doing their everyday work. The paper has the aim to show a way to integrate a framework for context-sensitive information retrieval into a project planning portal. This kind of software targets the successful execution of processes in enterprises and facilitates cooperation between the involved project teams. The included search functionality often is insufficient for the companies goals: promotion of reuse of existing parts and information like best practice and lessons learned documents. The proposed extended search functionality includes context information about both the engineering designer himself and the documents to anticipate the user's information need and thus improve the quality of the retrieval results. Amongst others, especially process and workflow information is gathered to supply adequate information for the current working task of the engineering designer.

1 Introduction

In the field of engineering design projects are nowadays mostly done in a distributed way spread over several teams and places. Additionally, several domain experts are included in these projects: mechanical engineers, electrical engineers, mechatronic engineers and computer scientists. All these experts fulfill certain roles in the projects and are responsible for parts of the final result. To manage the interaction between the different parties usually some kind of electronical project planning takes place. Ranging from simple excel sheets up to full blown project planning portals these solutions have in common that they aim at supporting and ensuring the success of the project.

To leverage the knowledge in the company it is necessary to increase the reuse of results of previous projects. In contrast to a simple query-based search, a context-sensitive search engine can return more relevant search results according to the user's current information need. There exists a large variety of potential influencing factors which affect the anticipated search results starting from personal domain knowledge, current state of the project, available product models to company guidelines concerning construction, e.g. for specific design objectives such as low-cost production.

Our intended search engine exceeds the usual extent of document coverage of a common search engine. In the domain jargon the interesting documents are subsumed as *product models* which describe all emerging representations of the final product. That includes paper-based, digital and physical product models [Lauer *et al.*, 2007]. Arguably, it is difficult to return a tangible model in a digital tool, but often the existence and creation of physical models is noted somehow and therefore a pointer to the real model can be returned.

It is intended to cover all evolving product models during the whole development process starting from initial product specifications from the customer to the final product documentation. In between lies a vast amount of different document types which are needed to represent a product during construction. This starts from text-based requirement specifications, cost calculations, feasibility studies and time schedules. In later phases (design and elaboration phase) the description of the product models changes to CAE (Computer Aided Engineering) documents, such as 3D models, technical drawings, or simulations and their results. As a first (incomplete) overview the document types can be subsumed as structured and unstructured text documents, 2D and 3D models as well as simulation data.

The diversity of artifacts requires a differentiated view of the documents. They usually contain several features which have to be considered during similarity search. A CAD (Computer Aided Design) drawing not only contains geometry information but also includes material data and topology information about assemblies, parts and subparts. Due to those different features the necessity of different index structures and combining algorithms arises.

Furthermore, our context-sensitive search framework incorporates metadata gathered from documents and context information to describe a document and its function in a process more precisely. These different views at a document enable a multi-criteria search and ranking which is illustrated in figure 1. It visualizes the change in the information need of the user during the proceeding of the process as well as the different weightings of the features depending on the process progress. The different aspects of each information need and each artifact are represented in separate ways. Thereafter, a weighted combination of the similarities/relevancies of the different aspects is created applying the metrics $m_{i,r,f}$ and weights $w_{i,r,f}$ selected for the information need i, the representation r and the feature f.

For the document-specific features extractors were implemented for common document types and exchange formats typical in the domain under research, e.g. STEP (Standard for the Exchange of Product Model Data) and DXF (Drawing Interchange Format). To gather more metadata

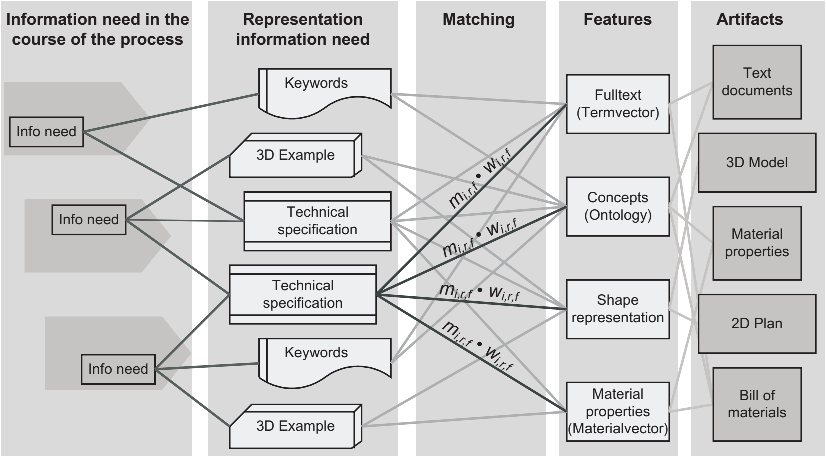


Figure 1: Example of a multi-criteria search

which goes beyond the creator and the creation date, we employ a process management portal which already delivers a huge amount of necessary metadata derived from a process metamodel. That allows the classification of documents into a process, the assignment of documents to different roles etc. which is described further in section 3. For user queries it is now possible to include additional context information which as well can be derived from the project planning portal. The paper therefore outlines the steps to integrate a context-sensitive search engine into a project planning portal and points out which contextual factors can be extracted and which further information would provide additional benefit.

The paper is organized as follows. In section 2 related work for the field of context-sensitive information retrieval is discussed. Section 3 deals with the information content of a project planning software and outlines the contextual factors an integrated search engine can use. Section 4 discusses which additional information would be helpful in searching but is not yet provided. Section 5 shows which steps were taken for the integration of the context-sensitive search engine into the project planning portal. Section 6 introduces the possibilities and problems of user interface design concerning contextual search and search result presentation in terms of being non-intrusive. Finally the paper outlines encountered problems and solutions and provides a conclusion and a view into the future work of the project.

2 Related Work

The aim of our IR (Information Retrieval) system is a goal-oriented provision of information in the product development process by providing reusable artifacts and considering context information. Thereby an improvement of the development process and a lowering of construction time and cost can be achieved.

The *principle of polyrepresentation* [Ingwersen, 1994] from the field of IS&R (Information Seeking and Retrieval) suggests that the information need of a person should be represented by a vast variety of influencing "features" which have an impact on the user query. The same applies to the documents in the index, i.e. their representation is not feature-complete when just the actual contents are taken into account. Therefore, an advanced search engine needs to combine different aspects. For example, a 3D similarity comparison should include information about geometry, topology, material properties, and metadata of an artifact to improve the retrieval results.

Moreover, the inclusion of contextual knowledge will have an important impact on search results [Allan, 2003]. Ingwersen's *label effect* [1982] stated that a user is not expressing the whole information he has about his information need, but only that amount he thinks is *enough* for a human recipient [Ingwersen and Järvelin, 2005]. Therefore, an augmentation of contextual knowledge to the expressed query is necessary.

At this point, the concept of context has to be divided into two aspects: the context of the indexed documents and the context of the person starting a search. According to Dey and Abowd [2000], we understand the latter one as the accumulation of all factors which have an impact on the information need of an engineering designer during his work. The document's context is e.g. characterized by its classification into the process and the role it plays in the process's lifetime.

Furthermore, a search engine can support two opposed ways of query issuing. The classic way consists of querying the search functionality on demand. That means the user asks for search results when needing them. A context-sensitive search engine then may augment the query with collected context information of the user to narrow the search space or state the query more precisely.

The contrary way produces search results proactively when the search engine notices that the user has an information need. The query is assembled through the different sensed contextual inputs which form the information need.

In an enterprise it is a common use case for a purchaser to search for off-the-shelf parts. If it would be possible to find similar parts of an assembly which are already produced or purchased in the company it would be beneficial to use these parts instead of redesigning them from scratch. Such a search will occur quite frequently and will often be queried actively by the purchaser.

In contrast, many tasks of an engineering designer are highly creative and therefore the worker does not want to get interrupted by unnecessary search activities. But during these designing tasks the search engine can proactively analyze the user's context as well as the viewed and edited files. Through inference the search engine can deduce the user's information need. That enables the retrieval of parts in the company's part repository which fulfill the same function and fit in the current project assembly. If the retrieval is successful it not only saves money for the company since the part does not have to be designed and maintained for the whole product lifecycle, but also the different testing tasks for the part can be skipped which is a huge timesaver.

In conclusion, our IR-system should deliver different kinds of artifacts relevant for a designer's information need in a specific process phase.

One of the first references of context-aware computing in literature came from Schilit [1994] where a ubiquitous computing system was introduced which was able to react to different signals of a wireless, palm-sized computer. The field of context-sensitive or context-based information retrieval was then broadened to a wide variety of fields of application.

With the development of the World Wide Web several recommendation systems for web pages were proposed. Examples are Lieberman's Letizia [1997] and Joachim's Web Watcher [1997]. Horvitz et al. [1998] developed a prototype for a context-sensitive help system for Microsoft Office which determined the user's context from several influencing factors which were input for a bayesian network.

The COBAIR project aimed at supporting computer scientists in the software engineering domain [Morgenroth, 2006; Henrich and Morgenroth, 2003]. According to the interactions of a user with the integrated development environment (IDE) the search engine inferred the user's information need and returned artifacts the user might need, e.g. software classes the user could reuse.

The DFKI[1] project KnowMore tried to support users working in a knowledge-intensive workflow. Proactively, i.e. without user action, the system provides information the user needs for his current task. Abecker et al. [2000] modeled the information needs in several ontologies which were subsumed as an *organizational memory information system*. An ontology-based heuristic retrieval method is used to identify concepts of the modeled ontologies and to deliver the needed information.

Our context-sensitive information retrieval framework intends to support the search for all emerging product models during the development process. A system which has the same intentions is the *Design Navigator System* introduced in [Karnik *et al.*, 2005]. It targets the management of design information. The authors distinguish between information related to the design and information related to the design's evolution and history. That leads to six different types of information: requirements and specifications, functional decomposition, assembly structure, part geometry, annotations, and interconnections. The main goal is to consider all information during the design process which includes design rationales that can be reused in other projects by means of best practices. The system allows various access methods. The search functionality supports browsing, design rationale search and geometry-based search amongst others. However, context information is not incorporated.

3 Information content provided by a project planning portal

This section deals with the information contained in a project planning portal and therefore the potential input data for a context-sensitive search engine. Due to a tight integration into this kind of software it is ensured that much information concerning the user as well as the administrated documents can be gathered for querying. The challenge of the context-sensitive retrieval model is to filter and weight those following criteria which are influencing a document's as well as the user's context. Figure 2 provides an overview of the functions and submodules of a project planning software and its interlinkage with the search framework.

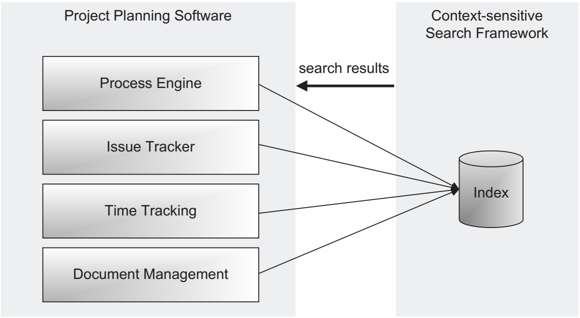


Figure 2: The tasks of a project planning software

3.1 Process Model

Each project usually follows a certain process model which describes the sequence of phases and activities, the documents that can/must be created therein, the responsible roles and the applicable methods. Project planning software usually allows the modeling of process models to fit the company's infrastructure. This modeling includes a partial description of the user's information need as documents are assigned to phases or tasks and accompanying roles. Thus documents might be entry or exit criteria for a phase, they might be revised in a phase or just needed for reference like company guidelines. This knowledge provides a much deeper understanding of a document and its purpose in a process for a search engine.

The classification of documents into a process model is of high significance for a context-sensitive search engine as the user's information need has a strong relation to the process phase the user is currently working in (cf. figure 1). Whereas during early phases more general documents are needed, later phases require more specific documents. This is motivated by the creative process in the beginning where potential solutions are evaluated. When the chosen solution is going to be constructed the designing engineer needs specific documents and partial solutions which meet the defined requirements.

Considering the filetypes it shows that each phase requires specific documents which should be taken into account when ranking documents. In the planning phase the dominating product models are text-based while in the design phase specific CAE models (2D/3D models, simulations etc.) are generated.

By knowing the affiliation of a document to a task or process phase a search engine can then deliver search results which help the user in his current task. This connection can be inferred through the assignment of roles to users which denote the tasks the user is responsible for. This permits the search engine to better infer the user's information need. Additionally, the interactions of the user with the project planning tool are analyzed and taken as input. If the task demands the existence of a finished document the search

[1] Deutsches Forschungszentrum für Künstliche Intelligenz (German Research Center for Artificial Intelligence)

engine can look for similar entry documents and then deliver the inferred result documents from past projects.

3.2 Time Tracking

When a project is executed the process model is instantiated and time, monetary, and human resources are allocated and assigned to the different parts of the project. To allow a successful completion of the project in a timely manner time restrictions for the phases and work tasks have to be set. A project planning software can assure that deadlines are followed in enabling notification mechanisms if a due date is approaching. Additionally, project managers can get a quick overview of the project's progress. Time tracking and the (financial) acquisition of amounts of work are additional factors a context-sensitive search engine can use in ranking a document.

If it is known how much time a work package will approximately take a search engine can upvalue documents from other projects with similar prerequisites to allow skipping this phase or at least to diminish the expenses. For example simulations can take a long time to complete, especially acoustic simulations, which makes it impossible to carry out too many change and test cycles. If simulation results are presented to the engineering designer he may postpone the actual simulation which is still needed but does not have to be executed unnecessarily.

Monetary values of documents can have an influence on the ranking as well. If the user is currently creating a document of high value – because of the required manual effort or the costly simulations, prototypes etc. – it seems to be beneficial to present similar or related documents as early as possible in order to boost reuse and the exploitation of analogies. To this end, the "value" of a document as well as its fit for the current situation of the user should be considered by the search system.

A context-sensitive search algorithm can take advantage of these factors when ranking documents because the search query comprises a more precise description of the user's information need. That leads to more precise answers and helps the user to deal with the information overflow.

3.3 Issue Tracking

In the software engineering domain issue tracking systems like JIRA[2] and Bugzilla[3] are widely used to manage the release process and the resolution of bugs. In product development small changes in parts often have impacts on other parts or modules which have to be resolved as well. A project planning software can support the assignment of those changes to a certain user. By inferring the next and time-pending tasks the context-sensitive search engine can recommend solutions for the raised issue querying similar projects. Additionally, it may provide and include *best practice* and *lessons learned* documents to prevent the user from following the wrong solutions and from repeating done mistakes in previous projects. The availability of further information enables the search engine to augment the proactive query with the task context which is derived from the issue tracking system.

3.4 Document Management

The comparison of several companies in the product development domain showed a variety of ways how documents produced during process execution are stored. This starts from storing files on a network share which is a semantically – at least for an indexing module – weak representation of the files. Storing documents in a PDM (Product Data Model) system denotes the other extreme as the user is forced to annotate files with metadata. Furthermore, those systems include workflow components which allow the classification into the process. The administration and storing of versions and variants supports configuration management.

A project planning software can pursue three different approaches. The simplest way is to just save a link to a file which was created during the process with all the arising problems, e.g. stale links due to the moving of files. If the system offers an integrated DMS (Document Management System) all the needed data (document and associated metadata) can be retained together and later used for an enhanced similarity search. The drawback of this approach is the potential redundancy as the files usually are as well stored in PDM systems. They allow a tighter integration with the used application systems in the company. Especially CAE systems often offer interfaces which enable a simple file exchange with a PDM system.

Our approach assumes that a direct connection between the project planning software and the DMS yield the best results. This allows a reliable way to augment documents with metadata. The emerging documents are *checked into* the project planning portal which relays them with added contextual information to the DMS. If an existing document is accessed the portal retrieves the document from the DMS, handles locking and makes the document available to the user. This final approach assumes the existence of a pluggable interface which enables the integration. Hereby no superfluous redundancy will occur and the document's context can be accessed from current data.

3.5 User Profile

Often it is necessary to store additional data for and about the user, e.g. his expert-level. This information often is used as a user interface filter which enables or disables several options according to the user's experience. Such a user profile will contain several types of information about the user. The user can provide information about his fields of interest, about expert knowledge, his former projects, his current roles, etc. A conceivable enhancement is the integration of skill databases which provide information about experts in an enterprise who can help or deliver information for specific tasks. A context-sensitive search engine will take this context information into account when handling a query to improve the description of the user's information need.

Search engine results frequently include information a user already knows and therefore clutter the result pages which makes the target-oriented information acquisition difficult. The consideration of contextual information can act as a filter which fades out the already known and therefore irrelevant result items.

3.6 Search functionality

Project planning tools support the retrieval of information stored in the program. But the supported extent of the search engine differs for the different tools on the market.

Usually, the integrated search supports the reactive approach which enables the user to query the search engine when he has an information need. The query is entered

[2] http://www.atlassian.com/software/jira/

[3] http://www.bugzilla.org/

into a textfield and the search results are presented in a one-dimensional list. Often the user has some filtering options like restricting the search to the current project or to include/exclude certain parts of the application, e.g. excluding search results from the issue tracker. The advanced search for special metadata is often implemented only rudimentarily which hinders the efficient retrieval of documents that would satisfy the information need of the user.

Often the search only includes the built-in descriptive pages which depict the documents and process parts (phases, roles and methods) but do not include the actual document content.

4 Desirable improvements for project planning software

As pointed out above a project planning portal enables document traceability which subsumes the localization of artifacts in a process. That information depicts a temporal relationship. If a causal relationship between parts of documents could be preserved in terms of requirements traceability, the search engine could support tasks where requirements change in later phases – which happens frequently – and present documents which were created because of the original requirement and now need a revision.

A similar improvement would be the storage and traceability of design rationales which denote the rational background why an artifact is designed the way it is [Lee and Lai, 1991]. This information is valuable in several situations. If design changes are necessary in later phases mutual interactions between part functions are better visible and explained by design rationales. That helps avoiding modifications which are impractical or impossible in regard to other product functionality. Furthermore, this information encapsulates design knowledge which should be reused in other projects to benefit from past experiences. Studies concerning the collection of design rationales can be found in [Regli *et al.*, 2000] and [Nomaguchi *et al.*, 2004].

More precise contextual information about specific artifacts like parts or assemblies could be derived if additional systems would be connected with the project planning portal or with the context-sensitive search framework. An ERP (Enterprise Resource Planning) system can deliver information about order and lot sizes as well as manufacturing costs which can play a role in document ranking. That pays off if parts which are already purchased in high quantities can be incorporated in new assemblies which are under construction.

In the domain of mechanical engineering several measures exist which describe the progress of a process. The *degree of maturity* can be derived in various ways which include the ratio of finished tasks to the number of all tasks, the ratio of finished requirements to all requirements amongst others [Pfeifer-Silberbach, 2004]. That information abstracts from the measurement of the process progress in terms of finished tasks or documents and focuses on a more product-centric degree of progress.

5 Steps taken for Integration

This section deals with the steps that were taken to integrate a context-sensitive search engine framework into a full-blown project planning portal software. Hereby we use the web-based process management portal *project kit* which was provided by our industrial project partner *method park Software AG*[4].

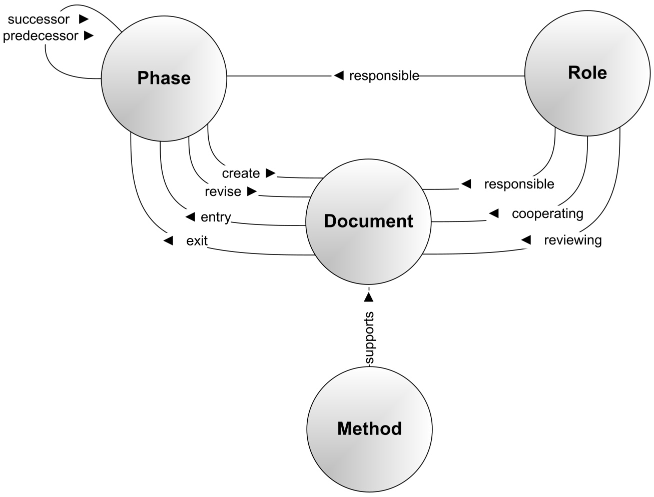


Figure 3: Document-oriented model process

5.1 Integration of a model process

As this research takes place in the domain of mechanical engineering a model process is needed to reveal domain-specific problems. A process was modeled according to the VDI[5] Guidelines 2221, 2222 (part 1) and 2223 [VDI, 2006]. This process consists of four main phases: planning, conception, design and elaboration. Our search engine supports the designing engineer during construction. Therefore, we omit the additional covered phases before start of production which are needed in process planning but do not yield an extra added value for our target group.

Due to the document-oriented process approach mainly found in the addressed domain documents which represent the product models are the driving parts in the process. Documents are created in certain phases. The user utilizes methods to produce the required product models. Special states can be assigned to documents, e.g. a document might represent a *milestone* which denotes the final outcome of a phase and therefore has to be created.

To approximate a more real-world model taking into account norms and practical experiences the phases were further divided which resulted in 15 phases with attached entry and exit criteria. This step was done with the help of two automotive suppliers and other research groups from the academic mechanical engineering environment. Before finally deploying the project planning software the process has to be tailored to meet the company's requirements.

As the project planning software includes a definable process metamodel the modeling of all kinds of processes is supported. The used process metamodel is displayed in figure 3. It defines four major process elements. Due to the document-oriented nature the *document* is the center of those elements. Documents are created or revised in a *phase* and can be entry or exit criteria for a phase, i.e. a phase cannot be finished before all necessary documents are delivered. The same applies to the preconditions. A phase cannot be started if a needed document has not been finished yet. *Roles* which are assigned to users are responsible for documents. Additionally a role can be cooperating

[4] http://www.methodpark.de

[5] Verein Deutscher Ingenieure (Association of German Engineers) – http://www.vdi.de

in creating a document or do the review process. This review process depends on the modeled document lifecycle which describes the different states an artifact can/must undergo. A change in a lifecycle state can as well be restricted to a certain role for coordination processes concerning the documents. For simplicity the document lifecycle was restricted to four stages: not existent, draft, review and released. Real-world company document lifecycles usually are more detailed.

Finally *methods* are assigned to documents which describe possible ways to create a document. For example after the function structure of a product is defined the possible solution principles have to be determined. One method to support creativity is the morphological box which collects possible technical solutions for sub-functions.

5.2 Identification of hooks for retrieving context

Our context-sensitive search engine can handle multiple contextual factors. To gather this information hooks into the project planning software have to be provided. The context information has to be distinguished into the user and the document context which both have to be determined.

The context-sensitive indexing module must be notified if documents are changed in the project planning software, i.e. when new documents are checked in, when old documents are changed or deleted. If new versions are created this relationship has to be preserved, so that it is possible to revert a document later and the index does not contain stale data.

The indexing step is the crucial part to derive additional data about the documents in the project planning software. Due to the integration of a process model (cf. section 3.1) much more information can be collected which describes a document and its function in the process further than a simple filescan of a network share. In addition the user can be forced to enter additional meta information, e.g. design rationales. With the interlinkage between the issue tracking system (cf. section 3.3) the tracking of design or bug changes appears possible.

The same applies to the influencing factors of a user's context. Although the user's context not just stems from his interaction with the project planning software, a quite reliable assessment of the state in the process he is in can be derived from the information in which projects he is active and to which tasks he is assigned. The history of projects and roles the user worked on leads to a promising assessment of the user's knowledge, special capabilities and technical strengths.

5.3 Integration Architecture

Figure 4 shows the architecture of the integration of our context-sensitive search framework into the web-based project planning portal solution is based on. To minimize the changes in *project kit* the interface between the two systems is established through aspect oriented programming [AOSD, 2007] – more precisely AspectJ [AspectJ, 2007]. All document changes are monitored using aspects which notify the indexing component to update the index. The same applies to the contextual information which is gathered using different aspects.

As can be seen in the figure 4 *context extractors* extract the document's context and transform it into features in the *feature generator*. The query module is split up into two parts. Reactive user queries are received and automatically augmented by contextual data of the user in the *context augmenter* module. If the search engine works proactively the *proactive query generator* constructs the query. Both types of queries are then passed to the *multi criteria matcher* which executes the multi-criteria search using several *metrics* and *weights* for each feature which are depending on the user's and document's context. The search results are finally passed back to the project planning portal which displays them to the user.

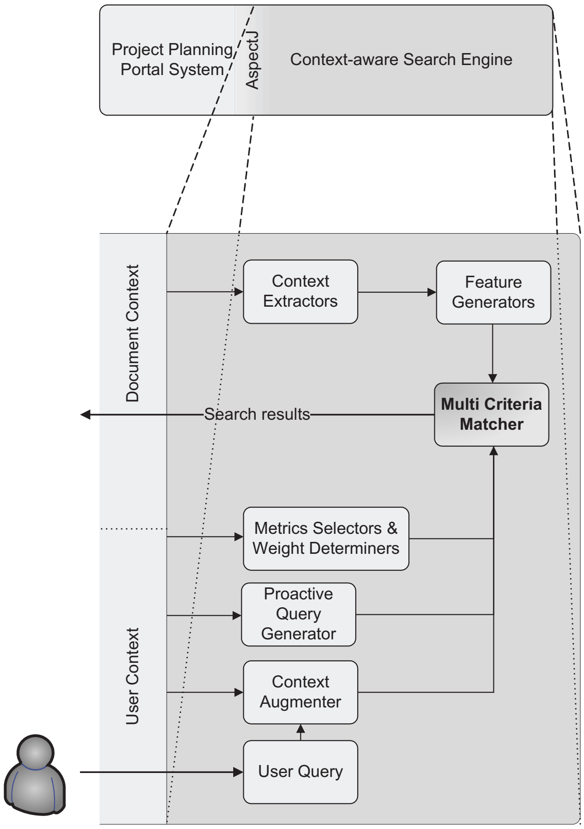


Figure 4: Simplified integration architecture for a context-sensitive search engine into a project planning portal

6 Thoughts on User Interface Design for a context-sensitive Search Engine

The integration of an advanced search engine into an existing application will affect its user interface. Our context-sensitive search framework will support two different ways of offering results. The "traditional" way – which usually is already included in the out-of-the-box search – is to enter the query in a search field. Even if the search engine augments this query with contextual information, the UI design does not have to change as the action appears underneath. The alternative way of issuing a query is the proactive search. Here the search engine tracks the user interaction, tries to infer on what tasks the user is currently working. In addition a synchronization is done with the user profile to see if the user needs help with the current task. If an information need is identified the search engine queries the index itself with a self-constructed query with contextual information. This query can be envisaged as the difference of the short term and the long term context of the user.

Now the presentation of this data is the crucial part, as any form of pop-up window might disturb the user and interrupt him from doing his work – which is disrupting especially in a highly creative task like construction in the mechanical engineering domain. Therefore, the user interface must allow a non-intrusive way of presenting the search results. Furthermore, the user must have the possibility to turn off this kind of search because he might feel patronized by the search engine.

If a web-based portal solution is used, a HTML frame is imaginable which is updated after a period of time or is notified if relevant search results are available. Through the adoption of AJAX [Garrett, 2005] this is easily implemented. A bit more involving but less intrusive way could be the use of some icons, e.g. an unlit bulb and a lit bulb as a notification that search results were found. With a mouse-click the user can display the results on demand.

A completely different approach for the representation of search results is the usage of a faceted search [Yee *et al.*, 2003]. Hereby the user browses search results and has the ability to filter documents by adding facets to the provided search results. This approach seems feasible for the outlined problem from section 1 as the documents in the engineering domain can be classified by several categories. The process-oriented approach (cf. section 3.1) already provides different process meta information. Faceted search can start on top of all documents contained in the index but also supports an initial search based on a user query. The user then may filter the results in adding a facet which describes documents from a certain phase, the document type, the project and so on. Therefore, the user can narrow down the search results to browse only the documents relevant to him.

For using a faceted search the indexing module has to determine the classifications for the facets. Although that introduces problems if only a plain document is available, the existence of the project planning tool which provides additional document information (context) as described in section 3 enables several initial facets. In addition the document extractors during the indexing step can try to infer more facets like part numbers and material codes.

Incorporating context information in a search engine has another pitfall which has to be considered carefully in user interface design. Common full-text search engines return small snippets of the search results which exemplify why they were delivered. The user is able to see the matches of his textual query and the results. If the context-sensitive search engine proactively queries the index or augments a user query that approach is not feasible. Hence, an explanation must be provided so that the user can understand why and how the search results were retrieved. This functionality is necessary to ensure a high user acceptance – if the user does not comprehend why the search results are returned he might reject the search engine.

7 Encountered Problems and Challenges

The encountered problems of the integration of an context-sensitive search engine into a project planning portal can be divided into two parts: adoption and technical problems.

7.1 Adoption Problems

A specialized search engine should strive for full coverage of documents that might be useful for the context it is working in. In the domain of mechanical engineering it should support the retrieval of all documents which are created during construction and are necessary in the process. This can be a drawback for a project planning tool if engineering designers do not document everything they produce but only final results and milestones. For example "analog" sketches might disappear in ring binders and some steps which could be supported electronically are done in the designer's mind and are not archived for later access. Experienced engineers often deduce the functional structure of an assembly and think of possible solution principles and pick the right one. An inexperienced worker could use the method of the morphological box which is a matrix representation of sub-functions and their possible technical solutions. After the creation the best fitting solution for each function is taken and is used as a preliminary design. The retrieval of this kind of document can be useful later if requirement changes occur and another solution for a sub-function has to be chosen.

Other industry partners in the FORFLOW project stated that similar systems were not fully accepted and adopted by the users. Only those documents were archived in a document management system (DMS) which are dictated by the system. This problem evolves if the user does not see an immediate revenue for his efforts in doing this "extra work". Hence, a combination of a system addressing the documents maintained in the DMS and a system indexing the project fileshare seems to be a promising approach.

Rights management and authorization are also problem areas which cannot be solved easily and are partially dependent on the company culture. In some companies there exists a high degree of competition between different departments, i.e. results are not simply shared and therefore are not accessible for other departments.

A search engine that uses context information could present documents from other or older projects which include the solution to a current problem but would not be allowed to present them because of secrecy issues. Since there exists no solution to this problem, it would be imaginable that the search engine does not return the actual document but a pointer to a responsible person or project team which has the rights on those documents so that the engineer can try to obtain access through the official channels. If even this "expert search result" is not allowed at all in the company the main goal of our search engine framework – the support of higher reuse of components and assemblies – cannot be achieved. Nonetheless, prototypes like design studies that are strictly confidential exist in companies which should be excluded from searches.

7.2 Technical Problems

Some technical problems occurred as well. The integration of a search engine into a finished product can introduce unwanted redundancy into the system. A project planning tool needs some sort of persistence layer which uses a certain persistence framework. The context search framework as well needs to store information. Both the context of the documents and the user have to be persisted. This creates duplicate information which not only takes unnecessary space but as well might introduce stale data if some information is not updated correctly. The external storage is required to allow a fast retrieval of documents by its contextual information.

The retrievable documents themselves are stored in an index which saves the document representations. This index does not use a relational database but some kind of index structure like an inverted file which allows faster retrieval of full-text contents. The built-in search of a project planning tool usually uses some indexes which after the integration are superfluous and should be removed because of the higher hard disk usage.

Process planning tools not only try to support the steering of the process but also try to recommend the next steps an engineer might take. Therefore, much process documentation is included which explains methods, documents and process phases. A search engine should deliver this information as well and needs a way to index that data properly, but in many cases the documentation is in more or less proprietary formats.

As these tools offer the customization of the process, hooks are needed which notify the indexing module of occurred changes so that an index update can be triggered.

8 Conclusion and future work

This paper showed the process of integrating a context-sensitive search engine framework into a project planning software which until now only had a simple full-text search. Therefor we proposed an integration architecture which keeps changes in the target platform – the project planning portal – to a minimum. We outlined the advantages of the integration of an advanced search engine which takes process information and user profiles into account to deliver situation specific search results. This leverages the retrieval of existing knowledge in the company and facilitates the reuse of components.

The next steps consist in further exploiting the information supplied from the project planning portal and the inclusion of external contextual information to describe the user and document context more precisely. This will include e-mails, network fileshares and other applications like MS Office and especially for this domain CAE software. Finally specific parameters from the engineering domain will be researched, e.g. the design situation the engineer is in, the stage of maturity of a product, its complexity and its purpose of use.

Acknowledgements

The work described in this paper is part of the Bavarian Research Cooperation FORFLOW consisting of six Bavarian universities working in the fields of engineering design and computer science. It collaborates with 21 companies and is promoted by the Bavarian Research Foundation (Bayerische Forschungsstiftung BFS).

References

[Abecker *et al.*, 2000] Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann, Otto Kühn, and Michael Sintek. Context-aware, proactive delivery of task-specific information: The knowmore project. *Information Systems Frontiers*, 2(3/4):253–276, 2000.

[Allan, 2003] James Allan. Challenges in information retrieval and language modeling. *SIGIR Forum*, 37(1):31–47, 2003.

[AOSD, 2007] AOSD, 2007. `http://aosd.net/` - 2007-07-02.

[AspectJ, 2007] AspectJ, 2007. `http://www.eclipse.org/aspectj/` - 2007-07-02.

[Dey and Abowd, 2000] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *Proc. of Conf. on Human Factors in Computing Systems (CHI 2000)*, Den Haag, Niederlande, 2000.

[Garrett, 2005] Jesse James Garrett, 2005. `http://www.adaptivepath.com/publications/essays/archives/000385.php` - 2007-07-02.

[Henrich and Morgenroth, 2003] Andreas Henrich and Karlheinz Morgenroth. Supporting collaborative software development by context-aware information retrieval facilities. In *Proc. of the 14th Int. Workshop on Database and Expert Systems Applications (DEXA)*, Prague, Czech Republic, September 2003.

[Horvitz *et al.*, 1998] Eric J. Horvitz, John S. Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, Wisconsin, USA, July 1998.

[Ingwersen and Järvelin, 2005] Peter Ingwersen and Kalervo Järvelin. *The Turn - Integration of Information Seeking and Retrieval in Context*. 2005.

[Ingwersen, 1982] Peter Ingwersen. Search procedures in the library analyzed from the cognitive point of view. *Journal of Documentation*, 38:165–191, 1982.

[Ingwersen, 1994] Peter Ingwersen. Polyrepresentation of information needs and semantic entities: elements of a cognitive theory for information retrieval interaction. In *SIGIR '94: Proc. of the 17th Int. ACM SIGIR conf.*, pages 101–110, New York, NY, USA, 1994.

[Joachims *et al.*, 1997] Thorsten Joachims, Dayne Freitag, and Tom M. Mitchell. Web watcher: A tour guide for the world wide web. In *Proc. of the 15th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 770–777, 1997.

[Karnik *et al.*, 2005] M.V. Karnik, S.K. Gupta, D.K. Anand, F.J. Valenta, and I. A. Wexler. Design navigator system. In *Proc. of IDET/CIE 2005, Long Beach, California, USA*, September 24-28 2005.

[Lauer *et al.*, 2007] Wolfgang Lauer, Josef Ponn, and Udo Lindemann. Purposeful integration of product models into the product development process. In *Proc. of the Int. Conf. on Engineering Design, ICED'07*, 2007.

[Lee and Lai, 1991] Jintae Lee and Kum-Yew Lai. What's in design rationale? *Human-Computer Interaction*, pages 251–280, 1991.

[Lieberman, 1997] Henry Lieberman. Autonomous interface agents. In *Proc. of the ACM Conf. on Computers and Human Interface (CHI-97)*, Atlanta, Georgia, USA, 1997.

[Morgenroth, 2006] Karlheinz Morgenroth. *Kontextbasiertes Information Retrieval*. Logos Verlag Berlin, 2006.

[Nomaguchi *et al.*, 2004] Y. Nomaguchi, A. Ohnuma, and K. Fujita. Design rationale acquisition in conceptual design by hierarchical integration of action, model and argumentation. In *Proc. of IDET/CIE 2004*, Salt Lake City, Utah, USA, September 2004.

[Pfeifer-Silberbach, 2004] Ullrich Pfeifer-Silberbach. *Ein Beitrag zum Monitoring des Reifegrads eines Produktes in der Entwicklung*. Phd thesis, Fachbereich Maschinenbau an der technischen Universität Darmstadt, 2004.

[Regli *et al.*, 2000] W. C. Regli, X. Hu, M. Atwood, and W. Sun. A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*, 16:209 – 235, 2000.

[Schilit *et al.*, 1994] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US, 1994.

[VDI, 2006] VDI. VDI 2221 - 2223. VDI-Handbuch Produktentwicklung und Konstruktion, 2006.

[Yee *et al.*, 2003] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *CHI '03: Proc. of the SIGCHI Conf. on Human factors in computing systems*, pages 401–408, New York, NY, USA, 2003.

Information Retrieval on the Semantic Web - Does it exist?

Peter Scheir[1,2], **Viktoria Pammer**[1,2], **Stefanie N. Lindstaedt**[2]
[1]Graz University of Technology, Austria
peter.scheir@tugraz.at
[2]Know-Center Graz, Austria
vpammer,slind@know-center.at

Abstract

Plenty of contemporary attempts to search exist that are associated with the area of Semantic Web. But which of them qualify as information retrieval for the Semantic Web? Do such approaches exist?

To answer these questions we take a look at the nature of the Semantic Web and Semantic Desktop and at definitions for information and data retrieval. We survey current approaches referred to by their authors as information retrieval for the Semantic Web or that use Semantic Web technology for search.

1 Introduction

Although Semantic Web research is still a young discipline a fair amount of research exists and progress into the direction of the vision of [Berners-Lee *et al.*, 2001] is made. A major category in this area of research is search and retrieval of information in this new type of web. Within this paper we are going to take a closer look at current approaches to search in the Semantic Web and on the Semantic Desktop[1].

The paper is structured as follows: in section 2 we aim at identifying characteristics of an information retrieval system for the Semantic Web. In section 3 we survey present approaches to search in the Semantic Web, we focus on systems and models for searching documents and ontological concepts (section 3.1) and systems for searching ontologies (section 3.2). In section 4 we try to classify the surveyed systems according to common properties. We conclude with section 5.

2 Information Retrieval for the Semantic Web

Various approaches to search associated with the area of Semantic Web exist. Diverse techniques are employed addressing a variety of problems. However, the notion of information retrieval in the context of Semantic Web seems to be rather diffuse.

We propose the following characteristics of an information retrieval system for the Semantic Web:

- **Criterion 1:** The system operates on the Semantic Web
- **Criterion 2:** The system is based on technology for the Semantic Web
- **Criterion 3:** The system performs information retrieval and not data retrieval

2.1 Ad criterion 1: Semantic Web vs. Semantic Desktop

The Semantic Web [Berners-Lee *et al.*, 2001] is not indented to be a new web but an extension to the current one. In the current form of the web information gathering is a task performed mainly by humans using a web browser. The Semantic Web shall provide an infrastructure so that this task can be performed by computer programs. To allow for processing the information on the web by machines, information is annotated with machine-interpretable data. The Semantic Desktop [Sauerman *et al.*, 2005] paradigm aims at applying technologies developed for the Semantic Web to desktop computing to finally provide for a closer integration between (semantic) web and (semantic) desktop.

In order for a system to qualify as operating on the Semantic Web we require it to search resources that are publicly available on the web and to potentially search the whole web.

At present most search approaches based on semantic technologies are for the Semantic Desktop. The incubation of search approaches in a desktop environment is a known phenomenon in information retrieval and current web search engines are based on techniques originally development for non-web environments.

2.2 Ad criterion 2: Ontology-driven information retrieval vs. information retrieval for the Semantic Web

To our understanding, a central element of information retrieval approaches for the Semantic Web is that they use technologies developed for the Semantic Web. Examples of such technologies are the standards RDF , RDF Schema and OWL.

Ontology-driven information retrieval approaches utilize ontologies for retrieval purposes, e.g. to increase retrieval performance by using the information modeled in ontologies. Many information retrieval systems for the Semantic Web use ontologies for retrieval purposes as well. However ontology-driven information retrieval does not necessarily target the Semantic Web.

2.3 Ad criterion 3: Data Retrieval vs. Information Retrieval

For pointing out the difference between Data Retrieval and Information Retrieval we refer to [Baeza-Yates and Ribeiro-Neto, 1999]:

[1] Methods used by the systems presented here also appear in other field of research, such as databases, XML retrieval, geographic retrieval, etc. Within our work we focus on finding a working definition for information retrieval on the Semantic Web and only take a closer look at systems that meet this definition.

A data retrieval language aims at retrieving all objects which satisfy clearly defined conditions ... a single erroneous object among a thousand retrieved objects means total failure. For an information retrieval system, however, the retrieved objects might be inaccurate and small errors are likely to go unnoticed.

Some of the current approaches to search in the Semantic Web are data retrieval approaches in the sense of the definition above (c.f. [Castells *et al.*, 2007] for a discussion). The most prominent ones are the SQL like query languages for the Semantic Web as SPARQL Query Language for RDF (SPARQL)[2].

3 Survey

This section gives an overview of existing approaches[3] to search in the context of the Semantic Web or the Semantic Desktop. We here differentiate between two search approaches that retrieve information in the Semantic Web or on the Semantic Desktop:

1. Systems or models that *search for information in the form of documents or ontological elements* (see section 3.1)
2. Approaches that *search for ontologies* (being a special type of information on the Semantic Web) (see section 3.2)

For listing approaches in this section one of the following criteria had to be fulfilled:

1. Technologies for the Semantic Web are used for retrieval purposes (e.g. RDF or OWL)
2. The approach is referred to as *for the Semantic Web* by its authors

3.1 Search for documents or ontological elements

In this section we survey systems and models for searching resources on the Semantic Web and on the Semantic Desktop.

SHOE: *SHOE Knowledge Annotator* allows for embedding semantic markup into HTML pages. This markup can be searched using *SHOE Search* [Heflin and Hendler, 2000] which provides a graphical user interface for building complex queries based on an ontology. *Expos* a web-crawler searches for web-pages with SHOE markup, extracts the markup and stores it in a local knowledge base.

SEAL: SEAL [Stojanovic *et al.*, 2001] is a framework developing semantic portals. Part of this framework is ranking of search results. While all results to a search in a semantic portal are equally relevant, SEAL ranks those results with least inference steps needed from the original knowledge base highest.

OWLIR: OWLIR [Shah *et al.*, 2002] indexes RDF triples together with document content. OWLIR treats distinct RDF triples as indexing terms. RDF triples are generated by natural language processing techniques based on textual content. Search can be performed based on words and RDF triples with wildcards. Shah et al. [2002] report an increase of Average Precision using an approach taking semantic information into account opposed to a text-only approach.

SCORE: The Semantic Content Organization and Retrieval Engine (SCORE) [Sheth *et al.*, 2002] uses classification and information-extraction techniques to extract metadata from textual sources. This metadata is later used for semantic search. A user can issue a query by specifying the category of document and one or more attribute values of the metadata to the document.

QuizRDF: QuizRDF [Davies *et al.*, 2002] combines keyword-based search with search and navigation through RDF(S) based annotations. Indexing in QuizRDF is based on *content descriptors*, which can be terms from the documents and literals from RDF statements. A query is formulated in using terms, in addition the type of the resource that should be returned can be specified. Search results can be filtered by property values of resources.

PISTA: Aleman-Meza et al. [2003] present an approach to *Semantic Association Ranking* in the Semantic Web. Starting from two entities in a RDF graph they aim at finding semantic associations between them and rank these semantic associations based on their importance. Semantic associations are based on relations between the two entities in the RDF graph and on the types of the relations.

TAP: TAP [Guha *et al.*, 2003] aims at enhancing search results from the WWW with data from the Semantic Web. It performs a graph based search on a RDF graph from the web. It starts at one or more *anchor nodes* in the RDF graph, which have to be mapped to query terms. A breath first search is performed in the RDF graph, collecting a predefined amount of triples. Optionally only links of a certain type are followed in traversing the RDF graph.

[Stojanovic *et al.*, 2003]: Stojanovic et al. [2003] - as in [2001] - aim at ranking search results of a semantic portal. In [Stojanovic *et al.*, 2003] they use the *specifity* of the instance of a relation, which is higher the less often the instances of the concepts in the relation are present in other instances of relations. In addition the inference process of the statements is taken into account for ranking results. This time the deduction of rules is use for ranking search results.

BioPatentMiner: Bamba and Mukherjea [2004] present an approach to *ranking Semantic Web query results*. The set of triples returned by a RDQL query is ranked based on various factors. A central aspect for weighting nodes in the results graphs is an adapted version of Kleinberg's HITS algorithm. Instead of hub and authority scores *subjectivity* and *objectivity* scores are calculated and the type of links between resources is taken into account as well. In addition the position in the class hierarchy or the super class of a resource is taken into account for weighing nodes. Edges are weighted by inverse property frequency of a property in the results graph. For every result graph all node and edge weights are combined to produce a relevance value.

KIM: KIM [Kiryakov *et al.*, 2004] relies on information extraction and relates words in documents with concepts from an ontology. Before indexing, documents are enriched with identifiers for the ontological concepts the words in the document represent. These identifers are directly inserted into the indexed text. For homonyms the same identifier is used. Queries are formed using concepts and relations from the ontology.

InWiss: [Priebe *et al.*, 2004] present a *Search Engine for RDF Metadata* implemented in the InWiss knowledge portal. Before search is performed the RDF graph is extended by following transitive properties and directly relating the resources found this way with the resources the traversal orig-

[2] http://www.w3.org/TR/rdf-sparql-query/

[3] We, the authors of this paper are thankful for every information on existing approaches to search on the Semantic Web, that we have overlooked and thus are not listed in this section.

inated from. The same action is performed for the query. In an additional step a set theoretic approach to ranking search results is performed, where the number of matching properties of a query with a resource is divided by the total number of properties of the query.

[Rocha *et al.*, 2004]: Rocha et al. [2004] present a *Hybrid Approach for Searching in the Semantic Web* that combines full-text search with spreading activation search in an ontology. Search starts with a keyword based query. Results to the fulltext search are instances from the ontology. Those instances are used to initiate a spreading activation search in the ontology to find additional instances.

[Bangyong *et al.*, 2005]: Bangyong et al. [2005] present a preliminary approach to association search in the Semantic Web. They propose to generate a Bayesian network from an ontology and use it to find instances not found by the initial query. Initial search is done by traditional web search technology which returns a set of instances. Based on the results of this search, associated instances are searched using the Bayesian network. The transformation of the ontology into a Bayesian network is not explained.

[Song *et al.*, 2005]: Song et al. [2005] present a *Ontology-Based Information Retrieval Model for the Semantic Web.* They suggest to use *semantic index terms* for indexing documents, which means to use the same index terms for synonyms and different indexing terms for homonyms. Also, the query should be represented using semantic index terms. How the indexing process should take place and how a query should be formulated is not addressed.

[Zhang *et al.*, 2005]: Zhang et al. [2005] present an *enhanced model for searching in semantic portals*. They integrate text based search with a fuzzy version of the description logic $\mathcal{ALC}$. A result set to a query is represented as a class in the knowledge base. The retrieval status values (RSV) of the documents retrieved by text-based search for a query are used as the fuzziness degrees for the instances of this class. Zhang et al. [2005] provide a Semantic Web service for search which is queried programmatically.

CORESE: CORESE [Corby *et al.*, 2006] is an ontology-based search engine which operates on conceptual graphs internally. COROSE is queried using one or a combination of triples. The query language is similar to SPARQL, SeRQL or RDQL but allows for approximate search. Approximate search is based on semantic distance of two classes in a common hierarchy and the `rdfs:seeAlso` property. The relevance of a result is measured by the similarity to the query.

OntoSearch: Jiang and Tan [2006] call OntoSearch a *Full-Text Search Engine for the Semantic Web.* Search starts with a term-based query which yields to a set of documents, from these documents semantic metadata is extracted and used for a spreading activation search in an ontology. The extended set of concepts is used to rank the search results of the term based search. Ranking is done using the cosine measure with concepts from the ontology being introduced as additional dimensions in the vector space.

Beagle++: Beagle++ [Chirita *et al.*, 2006] extends the opensource search engine Beagle[4] by indexing functionality for RDF triples. For every document the predicate and object of the RDF triple the document is the subject in are indexed. Additionally so called *predicate paths* are indexed. These are paths of those predicates in the RDF graph that are traversed starting from the document node. Beagle++ is queried using terms.

[Choudhury and Phon-Amnuaisuk, 2006]: Choudhury and Phon-Amnuaisuk [2006] present a search system similar to the one presented by Rocha et al. [2004]. They implement a subset of the weighting functionality of the system presented in [Rocha *et al.*, 2004] and test their system on a smaller data set.

MESH: Castells et al. [2007] combine SPARQL based search with full text search. For ranking results of a SPARQL query they weight annotations of documents with concepts from an ontology using an tf*idf-like measure. Then they combine the results of a full-text-search with the ranked list obtained via the SPARQL query using the CombSUM strategy. For performing the full-text search they extract certain parts of the SPARQL query and use them as query terms.

3.2 Search for ontologies

Ontology search engines are systems that retrieve information in the form of ontologies. The search results are either entire ontologies, ontology modules or ontology elements like statements or single concepts.

Various applications of searching for (parts of) ontologies are possible, such as agent support, ontology reuse or pure information retrieval.

OntoKhoj: OntoKhoj [Patel *et al.*, 2003] returns a ranked list of ontologies for a given WordNet-sense. OntoKhoj searches according to WordNet synonyms and hypernyms of the input. Appropriate ontologies are ranked according to their interconnectivity, i.e. ontologies that are referred to more often are ranked higher. Relationships considered by OntoKhoj are e.g. an *rdfs:subClass* relationship to an element of another ontology. OntoKhoj differently weights different relations and also considers chained relationships.

Swoogle: Swoogle[5] [Ding *et al.*, 2004] is again a search engine for ontologies (Semantic Web documents in Swoogle terminology). Ding et al. implemented various crawlers, e.g. one using Google, another monitoring given websites and a third analyzing retrieved ontologies and spreading out using outgoing hyperlinks. Retrieved ontologies are indexed by keywords and metadata like ontology language and ranked. An ontology's rank is defined by where the ontology lies on a continuous line between "database" and "schema".

[Biddulph, 2004]: In [Biddulph, 2004], the author implemented a Semantic Web crawler which was designed to serve as an aggregation service for external software (agents). The main difference to the engines presented above is that Biddulph merges all retrieved statements into one model stored on and queried from a Joseki[6] server. Biddulph assumes all collected models as partial aspects of the same world and makes inferences on the merged model.

OntoSelect: OntoSelect[7] [Buitelaar *et al.*, 2004] is an ontology library that monitors the web for changes in ontologies and crawls for new ontologies. Available ontologies can be browsed by ontology, keywords (labels), classes and properties. The library can also be searched for ontologies containing certain keywords. The search is either performed directly on the given keywords or with whole sites, i.e. given a Wikipedia topic or any other website, OntoSelect will select one ore more ontologies that match the given page. For this, OntoSelect first extracts relevant keywords from the page and then searches for ontologies matching these keywords.

[4] `http://www.beagle-project.org/`

[5] `http://swoogle.umbc.edu/`

[6] `http://www.joseki.org/`

[7] `http://olp.dfki.de/ontoselect`

OntoSearch2: OntoSearch2[8, 9] [Pan *et al.*, 2006] provides the functionality to query ontologies in a repository. OntoSearch2 relies on an external agent (human or machine) to add ontologies to the repository. It then preprocesses these ontologies for querying. OntoSearch2 supports SPARQL queries and queries all ontologies in the repository. Return results are exact matches of the SPARQL queries and can be either whole ontologies or parts of them. OntoSearch2 supports DL-Lite, which is a sublanguage of OWL DL. Incoming ontologies are preprocessed by translation into DL-Lite.

Watson: Watson[10] [dAquin *et al.*, 2007] is the latest of the here presented ontology search engines. Watson crawls the web for semantic documents (OWL, RDF(S), DAML+OIL). The ontologies are analysed in a first step to detect new locations of ontologies. Then a number of metadata are calculated, e.g. expressivity and level of axiomatization. Watson also checks for "semantic" duplicates, e.g. the same ontology replicated at different locations, the same ontology in another language etc. Watson retrieves single ontology elements like classes or instances and corresponding ontologies.

4 Classification of existing approaches

For our work, we adopt the two top level categories introduced by [Esmaili and Abolhassani, 2006], *Semantic Search Engines* and *Ontology Search Engines*, and present two distinct classifications, one for *approaches to search for documents or ontological elements* (section 4.1) and one for *approaches to search for ontologies* (section 4.2).

4.1 Classification of approaches to search for documents or ontological elements

[Tomassen, 2006] classifies current approaches to ontology-based information retrieval into *Knowledge Base* and *vector space model* driven approaches. More generally speaking one can classify current approaches to Semantic Web information retrieval into (1) approaches that operate on top of knowledge bases and (2) approaches that operate on top of information retrieval systems. Approaches on top of knowledge bases model documents as elements in the knowledge base, for example as instances of a special class in the knowledge base representing documents. Queries are formulated using special query languages and reasoning mechanisms are employed in the knowledge based to retrieve relevant documents. Approaches on top of information retrieval systems index document metadata originating from the ontology together with document content.

The following abbreviations are used for column heads of Table 1:

W ... (Semantic) Web Search is performed in a web environment. Related to criterion 1 from section 2.

D ... (Semantic) Desktop Search is performed in a desktop. Related to criterion 1 from section 2. environment

CBQ ... concept based query The query is formed of concepts or entities stemming from a knowledge representation (e.g. as in SPARQL).

TBQ ... term based query The query is formed of words (terms) as in common search engines.

KBS ... knowledge-based system A knowledge based system is used, a knowledge representation is searched. Based on [Tomassen, 2006].

IRS ... information retrieval system An information retrieval system is used, a document representation is searched. Based on [Tomassen, 2006].

DR ... data retrieval All results are relevant and equally relevant. Related to criterion 3 from section 2.

IR ... information retrieval A ranked list of results is returned. Related to criterion 3 from section 2.

KR ... knowledge representation used Type (format) of the knowledge representation used.

4.2 Classification of approaches to search for ontologies

We think that ontology search engines can effectively be classified by the kind of input they accept. Further distinction is possible by which kind of metadata about the ontologies is generated and available for search, and whether search results are entire ontologies, modules or ontology elements like statements or concepts.

We summarise *Ontology Search Engines* in Table 2. The following characteristics are used:

Input Type of input: free text (F), keyword (K), formal element e.g. WordNet sense, ontology concept/property (FE), formal structure e.g. SPARQL (FS).

Crawler included: Yes/No

Storage of ontologies: Yes/No

Index Yes / No

KR Supported knowledge representations: OWL, DAML+OIL, RDF, RDFS

API Does an API (or any access-point for software like e.g. a webservice) to the search engine exist?

Online available: Yes / No

5 Conclusions

Under critical examination none of the surveyed approaches to *search for documents and ontological concepts* was able to fulfill all of the three characterizations introduced in section 2. Most of the presented systems operate in a desktop environment and some of the presented approaches are data retrieval systems that do not return a relevance value for a search result. Not all of the surveyed systems actually use technology for the Semantic Web. As their authors referred to these systems as being for the Semantic Web, they probably had a different conception about information retrieval in the Semantic Web. Another explanation could be that they have used ontology-based information retrieval and information retrieval for the Semantic Web synonymously.

We remark that the diversity of *ontology search engines* is not overly impressive, but on the other hand ontology search engines mostly adhere to all three criteria for information retrieval on the Semantic Web as given in section 2.

Finally, all of the presented system operate on the lower levels of the Semantic Web, making use of knowledge representation standards as RDF and OWL. Work in progress topics in Semantic Web research as *proof* and *trust* are not addressed in the presented systems.

Acknowledgments

We thank the anonymous reviewers of our submission for their constructive feedback. This work has been partially funded under grant 027023 in the IST work programme of

[8] `http://www.ontosearch.org/`

[9] OntoSearch2 and its predecessor OntoSearch mentioned in this section are not related to OntoSearch in section 3.1.

[10] `http://watson.kmi.open.ac.uk/Overview.html`

the European Community. The Know-Center is funded by the Austrian Competence Center program Kplus under the auspices of the Austrian Ministry of Transport, Innovation and Technology (www.ffg.at/index.php?cid=95) and by the State of Styria.

References

[Aleman-Meza *et al.*, 2003] B. Aleman-Meza, Ch. Halaschek-Wiener, I. B. Arpinar, and A. P. Sheth. Context-aware semantic association ranking. In *Proc. of SWDB'03, 1st Int. Workshop on Semantic Web and Databases*, 2003.

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

[Bamba and Mukherjea, 2004] B. Bamba and S. Mukherjea. Utilizing resource importance for ranking semantic web query results. In *Semantic Web and Databases, Second Int. Workshop, SWDB 2004*, 2004.

[Bangyong *et al.*, 2005] L. Bangyong, T. Jie, and . Juanzi. Association search in semantic web: search + inference. In *WWW '05: Special interest tracks and posters of the 14th int. conf. on World Wide Web*, 2005.

[Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.

[Biddulph, 2004] M. Biddulph. Semantic web crawling. In *XML Europe 2004*, 2004.

[Buitelaar *et al.*, 2004] P. Buitelaar, Th. Eigner, and Th. Declerck. Ontoselect: A dynamic ontology library with support for ontology selection. In *In: Proc. of the Demo Session at the Int. Semantic Web Conf.*, 2004.

[Castells *et al.*, 2007] P. Castells, M. Fernndez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Trans. Knowl. Data Eng.*, 19:261–272, 2007.

[Chirita *et al.*, 2006] P. A. Chirita, S. Costache, W. Nejdl, and R. Paiu. Beagle++: Semantically enhanced searching and ranking on the desktop. In *The Semantic Web: Research and Applications*, 2006.

[Choudhury and Phon-Amnuaisuk, 2006] S. Choudhury and S. Phon-Amnuaisuk. Semantic retrieval with spreading activation. In *Proc. of the MMU Int. Symposium on Information and Communications Technologies M2USIC 2006*, 2006.

[Corby *et al.*, 2006] O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, and F. Gandon. Searching the semantic web: Approximate query processing based on ontologies. *IEEE Intelligent Systems*, 21(1):20–27, 2006.

[dAquin *et al.*, 2007] M. dAquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, and E. Motta. Watson: A gateway for the semantic web. In *European Semantic Web Conf., ESWC 2007 (poster)*, 2007.

[Davies *et al.*, 2002] J. Davies, U. Krohn, and R. Weeks. Quizrdf: search technology for the semantic web. In *WWW2002 workshop on RDF & Semantic Web Applications, 11th Int. WWW Conf.*, 2002.

[Ding *et al.*, 2004] L. Ding, T. Finin, A. Joshi, R. Pan, R. Scott Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A Search and Metadata Engine for the Semantic Web. In *Proc. of the 13thACM Conf. on Information and Knowledge Management*, 2004.

[Esmaili and Abolhassani, 2006] K. S. Esmaili and H. Abolhassani. A categorization scheme for semantic web search engines. In *4th ACS/IEEE Int. Conf. on Computer Systems and Applications (AICCSA-06)*, 2006.

[Guha *et al.*, 2003] R. Guha, R. McCool, and E. Miller. Semantic search. In *WWW '03: Proc. of the 12th int. conf. on World Wide Web*, 2003.

[Heflin and Hendler, 2000] J. Heflin and J. Hendler. Searching the web with SHOE. In *Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01.*, 2000.

[Jiang and Tan, 2006] X. Jiang and A.-H. Tan. Ontosearch: A full-text search engine for the semantic web. In *Proc. of the 21st National Conf. on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conf.*, 2006.

[Kiryakov *et al.*, 2004] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics: Science, Services and Agents on the World Wide We*, 2:49–79, 2004.

[Pan *et al.*, 2006] J. Z. Pan, E. Thomas, and D. Sleeman. Ontosearch2: Searching and querying web ontologies. In *Proc. of the IADIS Int. Conf. WWW/Internet 2006*, 2006.

[Patel *et al.*, 2003] C. Patel, K. Supekar, Y. Lee, and E. K. Park. Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In *WIDM '03: Proc. of the 5th ACM int. workshop on Web information and data management*, 2003.

[Priebe *et al.*, 2004] T. Priebe, Ch. Schlager, and G. Pernul. A search engine for rdf metadata. In *DEXA '04: Proc. of the Database and Expert Systems Applications, 15th Int. Workshop on (DEXA'04)*, 2004.

[Rocha *et al.*, 2004] C. Rocha, D. Schwabe, and M. Poggi de Aragão. A hybrid approach for searching in the semantic web. In *Proc. of the 13th int. conf. on World Wide Web (WWW)*, 2004.

[Sauerman *et al.*, 2005] L. Sauerman, A. Bernardi, and A. Dengel. Overview and outlook on the semantic desktop. In *Proc. of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conf.*, 2005.

[Shah *et al.*, 2002] U. Shah, T. Finin, and A. Joshi. Information retrieval on the semantic web. In *CIKM '02: Proc. of the 11th int. conf. on Information and knowledge management*, 2002.

[Sheth *et al.*, 2002] A. Sheth, C. Bertram, D. Avant, B. Hammond, K. Kochut, and Y. Warke. Managing semantic content for the web. *IEEE Internet Computing*, 6:80–87, 2002.

[Song *et al.*, 2005] J. Song, W. Zhang, W. Xiao, G. Li, and Z. Xu. Ontology-based information retrieval model for the semantic web. In *2005 IEEE Int. Conf. on e-Technology, e-Commerce, and e-Services (EEE)*, 2005.

[Stojanovic *et al.*, 2001] N. Stojanovic, A. Maedche, S. Staab, R. Studer, and Y. Sure. Seal: a framework for developing semantic portals. In *Proc. of the 1st Int. Conf. on Knowledge Capture (K-CAP)*, 2001.

[Stojanovic *et al.*, 2003] N. Stojanovic, R. Studer, and L. Stojanovic. An approach for the ranking of query results in the semantic web. In *Int. Semantic Web Conf.*, 2003.

[Tomassen, 2006] S. L. Tomassen. Research on ontology-driven information retrieval. In *OTM Workshops (2)*, 2006.

[Zhang *et al.*, 2005] L. Zhang, Y. Yu, J. Zhou, Ch. Lin, and Y. Yang. An enhanced model for searching in semantic portals. In *WWW '05: Proc. of the 14th int. conf. on World Wide Web*, 2005.

	W	D	CBQ	TBQ	KBS	IRS	DR	IR	KR
[Heflin and Hendler, 2000]	X		X		X		X		SHOE
[Stojanovic *et al.*, 2001]	(X)[a]		X		X			X	F-Logic
[Davies *et al.*, 2002]		X	(X)[b]	X	X	X		X	RDF & RDFS
[Shah *et al.*, 2002]		X	X[c]	X	[d]	X		X	DAML+OIL
[Sheth *et al.*, 2002]		X	X	[e]	X		X		unknown
[Aleman-Meza *et al.*, 2003]	[f]		X		X			X	RDF
[Guha *et al.*, 2003]	X		(X)[g]		X		X		RDF
[Stojanovic *et al.*, 2003]	(X)[h]		X		X			X	F-Logic
[Bamba and Mukherjea, 2004]		X	X		X			X	RDF & RDFS
[Kiryakov *et al.*, 2004]		X	X	X	X	X	X[i]	X	RDFS
[Priebe *et al.*, 2004]	(X)[j]		X		X			X	RDF
[Rocha *et al.*, 2004]	(X)[k]			X	X	X		X	unknown[l]
[Bangyong *et al.*, 2005]	[m]			X	X	X		X	unknown
[Song *et al.*, 2005]	[n]		X			X		X	none
[Zhang *et al.*, 2005]	(X)[o]		X	(X)[p]	X	X		X	fuzzy DL $\mathcal{ALC}$
[Corby *et al.*, 2006]		X	X		X			X	RDF, RDFS, OWL Lite
[Choudhury and Phon-Amnuaisuk, 2006]		X		X	X	X		X	unknown
[Jiang and Tan, 2006]		X		X	X	X		X	taxonomy[q]
[Chirita *et al.*, 2006]		X		X		X		X	RDF
[Castells *et al.*, 2007]		X	X		X	X		X	OWL

Table 1: Characteristics of approaches to search for documents or ontological elements

[a] search on semantic portal

[b] filtering by type and search in property values

[c] RDF-triples with wildcards

[d] inference in knowledge base to expand set of triples indexed with text

[e] not designed for full-text search, but could be done

[f] search in RDF graphs

[g] nodes in RDF graph

[h] search on semantic portal

[i] entities can be retrieved from knowledge base

[j] search on knowledge portal

[k] search on website

[l] „can be easily mapped to RDF"

[m] model only, no system

[n] model only, no system

[o] search on semantic portal

[p] term based queries are modeled as instances to concepts in knowledge base

[q] ACM Computing Classification System

Search Engine	Input	Crawler	Storage	Index	KR	API	Online
OntoKhoj	FE	Yes	-	Yes	RDF(S), DAML+OIL, OWL	-	No
[Biddulph, 2004]	FE	Yes	Yes	No	RDF(S), OWL	Yes	No
Swoogle	K	Yes	No	Yes	RDF(S), DAML+OIL, OWL	Yes	Yes
OntoSelect	F / K	Yes	-	-	RDF(S), DAML, OWL	No	Yes
OntoSearch2	K[a] / FS	No	Yes	No	OWL	Yes	Yes
Watson	K	Yes	Yes	Yes	RDF(S), DAML+OIL, OWL	Yes	Yes

Table 2: Overview of ontology search engines

[a] It seems that results are only available for queries containing a single keyword

Ontologiebasierter Forschungsführer für die Bildungsforschung

Carola Carstens, Marc Rittberger
Deutsches Institut für Internationale Pädagogische Forschung
Informationszentrum Bildung
D-60486 Frankfurt am Main
carstens@dipf.de; rittberger@dipf.de

Abstract

Dieser Beitrag beschreibt das Vorhaben, Semantic Web-Technologien für den Aufbau eines ontologiebasierten Forschungsführers einzusetzen. Heterogene, verteilte Datenquellen des Informationszentrums Bildung sollen auf diese Weise semantisch integriert, angereichert und über entsprechende Recherchemöglichkeiten zugänglich gemacht werden. Die einzelnen Schritte zur Umsetzung des geplanten Projekts werden vorgestellt, wobei insbesondere auf zu erwartende Mehrwerte gegenüber der aktuellen Datenquellennutzung eingegangen wird.

1 Einleitung

Semantic Web-Technologien werden vorzugsweise entwickelt, um die Maschinenlesbarkeit und die Interoperabilität von im Internet verteilten Informationen zu gewährleisten und darauf basierend Anbieter übergreifende semantische Suchfunktionalitäten zu entwickeln. Darüber hinaus können sie allerdings auch eingesetzt werden, um die Daten eines einzelnen Informationsanbieters semantisch zu integrieren, anzureichern und über ontologiebasierte Retrievalmethoden recherchierbar zu machen. Dies wird anhand eines konkreten Anwendungsvorhabens im vorliegenden Beitrag skizziert.

Die Konzeption eines ontologiebasierten Forschungsführeres soll verdeutlichen, dass durch den Einsatz von Semantic Web-Technologien auf der Datenbasis eines einzelnen Informationsanbieters bereits Mehrwerte für den Prozess der Informationssuche generiert werden können.

2 Projektbeschreibung

Das Projektziel besteht darin, einen ontologiebasierten Forschungsführer für den Bereich Bildungsforschung zu konzipieren, der Informationen über Forscher und Forschungsinstitutionen dieses Fachgebiets sowie ihre jeweiligen Forschungsbereiche auf der Basis semantischer Technologien in einem Portal präsentiert und recherchierbar macht.

Die für den Aufbau des Forschungsführers zur Verfügung stehenden Ausgangsdaten des Informationszentrums Bildung (IZB) liegen derzeit in verteilten heterogenen Datenquellen vor. Sie sollen mithilfe von Semantic Web-Technologien semantisch integriert und angereichert werden, sodass ein Mehrwert gegenüber der aktuellen Nutzung der Datenquellen entsteht.

Die Anreicherung des Wissens erfolgt durch Inferenzregeln, die durch logische Schlussfolgerungen zu neuartigen Verknüpfungen des vorhandenen Wissens führen. Die auf diese Weise erweiterte Wissensbasis soll den Nutzern über eine Rechercheschnittstelle zugänglich gemacht werden. Die ontologiebasierte Datenintegration ermöglicht es, über Anfragen an die Ontologie Datenquellen übergreifende konzeptbasierte Suchen durchzuführen.

Den Einsatz derartiger semantischer Verfahren für die Prozesse der Datenintegration und des Retrievals gilt es abschließend zu evaluieren.

3 State of the Art

Der Vision des Semantic Web zufolge sollen Informationen im Internet in Zukunft derart semantisch ausgezeichnet sein, dass Agenten diese Informationen Anbieter übergreifend aggregieren und interpretieren können [Berners-Lee *et al.* 2001]. Eine wichtige Voraussetzung für die Realisierung dieser Vision stellen Ontologien dar, welche zur semantischen Annotation genutzt werden und den Anwendungen die Interpretation der semantischen Zusammenhänge zwischen den einzelnen Auszeichnungselementen erst ermöglichen.

Da es nicht realistisch ist, eine allumfassende Ontologie des gesamten Weltwissens zu modellieren, widmen sich verschiedene Semantic Web-Initiativen der Aufgabe, Standardontologien für abgegrenzte Anwendungsgebiete zu definieren. Im Zuge dieser Bestrebungen sind unter anderem die folgenden Ontologien entwickelt worden, welche für die Annotation der Inhalte des geplanten Forschungsführers relevant erscheinen: FOAF[1] („Friend-of-a-friend“) für die Darstellung von Personenprofilen und -netzwerken, SKOS[2] („Simple Knowledge Organisation System“) für die Abbildung von Thesaurusrelationen und SWRC („Semantic Web for Research Communities“) für die Modellierung von Forschungsgemeinschaften [Sure *et al.* 2005].

Semantic Web-Anwendungen sehen sich derzeit jedoch noch häufig mit dem Problem der mangelnden Verfügbarkeit von derart ausgezeichneten Daten im Internet konfrontiert. Sie müssen sich daher entweder auf die verfügbaren semantisch annotierten Daten beschränken, oder es wird der Ansatz verfolgt, herkömmliche Daten aus Webseiten zu extrahieren und in

[1] http://www.foaf-project.org/

[2] http://www.w3.org/2004/02/skos/

einen Semantic Web-Standard zu überführen, um sie für diese Anwendungen nutzbar zu machen.

Im Projekt DBpedia[3] wird beispielsweise das Ziel verfolgt, Informationen aus der Online-Enzyklopädie Wikipedia für das Semantic Web aufzubereiten. Zu diesem Zweck werden strukturierte Informationen, z.B. aus Wikipedia-Infoboxen, extrahiert und in RDF überführt. Auf der Grundlage dieser Datenbasis können über Anwendungen wie den DBpedia Relationship Finder[4] die semantischen Verknüpfungen zwischen Objekten der Datenbasis abgefragt werden [Auer und Lehmann 2007].

Im „Personal Publication Reader“-Projekt werden Informationen über Forscher und Publikationen mithilfe von Wrappern aus Webseiten extrahiert und in den RDF-Standard überführt. Über eine Ontologie werden diese Daten miteinander in Beziehung gesetzt, um regelbasiert personalisierte Sichten auf die ursprünglich verteilten Datenbestände generieren zu können [Baumgartner *et al*. 2005].

Das Projekt FLINK hingegen bezieht die Daten für die Darstellung von Forschernetzwerken in einem semantisch basierten Portal unter anderem aus verfügbaren FOAF-Profilen und mithilfe von Web Mining-Methoden [Mika 2005].

Das Projekt Ontoframe verwendet im Gegensatz dazu projektinterne Ressourcen für den Aufbau einer semantischen Informations- und Kollaborationsplattform. Zur Instanziierung der dem Portal zu Grunde liegenden Ontologie werden unter anderem die Datensätze einer Publikationsdatenbank in RDF-Tripel transformiert [Jung et al. 2007].

Dass sich die in den beschriebenen Projekten angewandten Methoden der auf Semantic Web-Standards basierenden Informationsintegration, -anreicherung und -präsentation auch Mehrwert bringend für die Zusammenführung interner Datenquellen einsetzen lassen, soll im geplanten Projekt gezeigt werden.

Die Informationsintegration verfolgt hierbei das Ziel strukturelle und semantische Heterogenitäten zwischen den Ursprungsdatenquellen aufzulösen. Entitäten, die in den einzelnen Datenquellenschemata unterschiedlich modelliert sind, sollen über die Ontologie zusammengeführt und semantisch kontextualisiert werden. Dass die Heterogenitätsbehandlung bei der Integration von Informationssystemen eine große Herausforderung darstellt, unterstreichen auch die Arbeiten von Baerisch und Stempfhuber, die Methoden zum Umgang mit semantischen und strukturellen Heterogenitäten bei der Implementierung von Datenquellen übergreifenden Rechercheschnittstellen beschreiben [Baerisch 2007; Stempfhuber 2003].

Im geplanten Projekt wird die Auflösung derartiger Heterogenitäten durch die ontologiebasierte Datenzusammenführung angestrebt.

4 Vorgehensweise

In diesem Abschnitt skizzieren wir die geplanten Schritte für die Realisierung des Forschungsführers. Hierzu zählen die Auswahl der Datenbasis, die Entwicklung eines Ontologieschemas für die semantische Datenintegration, die Entscheidung für eine Methode der Datenquelleneinbindung, die Definition von Inferenzregeln für das Reasoning über die Ontologie, sowie die Konzeption einer Benutzerschnittstelle.

4.1 Auswahl der Datenquellen

Für den geplanten Forschungsführer sind alle verfügbaren Informationen über Forscher, Institutionen sowie Forschungsinhalte aus dem Bereich der Bildungsforschung von Interesse. An dieser Stelle wird daher ein kurzer Überblick darüber gegeben, welche Datenquellen des Informationszentrums Bildung relevante Daten enthalten, und wie sie derzeit genutzt werden.

FIS Bildung Literaturdatenbank

Die FIS Bildung Literaturdatenbank[5] enthält momentan über 600.000 Literaturnachweise zu pädagogischen und erziehungswissenschaftlichen Publikationen und ist über eine Rechercheschnittstelle des Fachportals Pädagogik[6] zugänglich [Bambey und Jornitz 2006].

Alle Literaturnachweise enthalten neben bibliografischen Angaben auch Schlagworte, die intellektuell nach vorgegebenen Standards vergeben werden. Die auf diese Weise entstehende Beziehung zwischen Autoren, ihren Publikationen sowie den vergebenen Schlagworten stellt eine wichtige Ressource für den Forschungsführer dar.

FIS Bildung Wörterbuch

Das Informationszentrum Bildung hat für die eigene Indexierungsarbeit ein Wörterbuch mit thesaurusähnlichen Strukturen wie Hyperonym-, Hyponym- und Synonymrelationen für den Bereich Bildung entwickelt. Darüber hinaus ist jedem Term eine Nummer zugewiesen, die ihn einer Kategorie der vom IZB genutzten Pädagogik-Systematik zuordnet. Das Wörterbuch ist nur intern zugänglich, wird in einer relationalen Datenbank verwaltet und für die intellektuelle Indexierung von Dokumenten der FIS Bildung Literaturdatenbank eingesetzt.

Institutionen- und Personendatenbank

Über das Fachportal Pädagogik ist zudem der Zugriff auf eine Institutionen- und Personendatenbank möglich[7]. Sie gibt Auskunft über die Forschungsgebiete, Adressen und Mitarbeiter von Institutionen und Personen aus dem Bereich der Bildungsforschung. Momentan ist der Zugriff auf diese Datenbank über ein hierarchisches Browsing sowie über eine Volltextsuche möglich.

Portalnutzer haben die Möglichkeit, ihr Profil formularbasiert in der Personendatenbank zu hinterlegen. Auf die gleiche Weise können sich auch Institutionen präsentieren.

4.2 Erstellung eines Ontologieschemas

Zur semantischen und strukturellen Integration dieser Daten muss ein Ontologieschema erstellt werden, das die Konzeptwelt des Bereichs Bildungsforschung abbilden kann. Es soll die oben aufgeführten Daten in der Form von Konzepten abbilden und miteinander über semantische Relationen in Beziehung setzen.

[3] http://dbpedia.org/docs/

[4] http://wikipedia.aksw.org/relfinder/

[5] http://www.fachportal-paedagogik.de/fis_bildung/index.html

[6] http://www.fachportal-paedagogik.de/

[7] http://www.fachportal-paedagogik.de/branchenverzeichnis/index.html

Da im Projekt der Anspruch verfolgt wird, semantisch ausdrucksstarke Relationen zu definieren, kann dieser Modellierungsprozess nur intellektuell erfolgen. Außerdem muss berücksichtigt werden, dass die Güte der späteren Anwendung maßgeblich von der Gestaltung des Ontologieschemas abhängt.

Bei der Definition dieses Schemas wird auf bereits bestehende Ontologien zurückgegriffen, um im Sinne der Semantic Web-Vision die spätere Interoperabilität des Systems mit externen Anwendungen zu unterstützen. Wie aus der folgenden Grafik (Abb.1) hervorgeht, wird das Vokabular der SWRC-Ontologie übernommen und um eigene Ausdrücke mit dem Präfix *vocab* erweitert. Darüber hinaus sollen auch die in Abschnitt 3 angesprochenen Standards FOAF und SKOS Anwendung finden.

Diese Grafik bildet einen Ausschnitt des auf diese Weise erstellten Ontologieschemas ab:

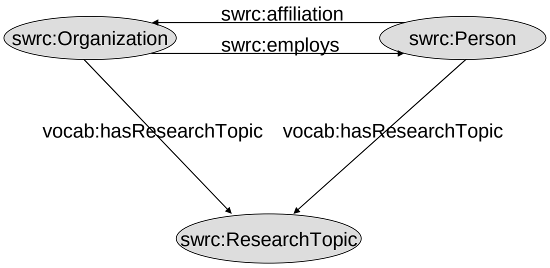


Abb.1: Ausschnitt aus dem Ontologieschema

In der Abbildung sind die Konzepte *swrc:Organization*, *swrc:Person* und *swrc:ResearchTopic* dargestellt, die über semantische Relationen miteinander verbunden sind.

Dass die Ontologie komplexe Modellierungsmöglichkeiten für die semantische Kontextualisierung der Daten bietet, veranschaulicht die folgende Grafik (Abb.2) beispielhaft anhand der Modellierung der Relation swrc:*email*. So lassen sich nicht nur Klassen hierarchisch organisieren, sondern auch Relationen.

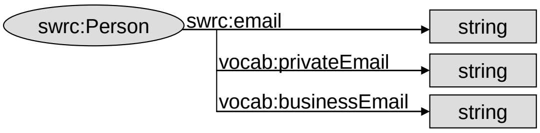


Abb.2: Modellierung der Relation swrc:email

Aus der Abbildung wird ersichtlich, dass die aus der SWRC-Ontologie übernommene Relation *swrc:email* für die eigenen Zwecke differenzierter modelliert werden musste. Dies führte zur Erweiterung der Relation *swrc:email* um die Unterrelationen *vocab:privateEmail* und *vocab:businessEmail*.

4.3 Einbindung der Datenquellen

Die Herstellung der Beziehungen zwischen dem Ontologieschema und den in den Datenquellen enthaltenen Daten erfolgt über einen Datenimport.

Alternativ wäre es auch möglich gewesen, die Daten in den Ausgangsdatenquellen zu belassen und über das ontologische Schema lediglich miteinander in Beziehung zu setzen. Mithilfe der Sprache D2RQ können beispielsweise relationale Datenschemata auf ein Ontologieschema gemappt werden [Bizer und Seaborne 2004]. Dadurch wird es möglich, RDF-basierte SPARQL-Abfragen an die Datenbanken abzusenden. Auf der Basis der in D2RQ definierten Mappings können diese Abfragen in SQL-Abfragen an die einzelnen Datenquellen transformiert werden, wobei das Ergebnis wiederum als RDF-Tripel zurückgeliefert wird.

Ein wichtiges Argument für den direkten Datenimport war jedoch die Tatsache, dass die prototypische Forschungsführer-Anwendung nicht in den laufenden Betrieb der Ursprungsdatenquellen eingreifen soll. Darüber hinaus bietet der skriptbasierte Datenimport hohe Flexibilität bei der Modellierung der ontologischen Wissensbasis. Als nachteilig stellen sich hingegen die redundante Datenhaltung und der Aufwand für eine regelmäßige Aktualisierung der Daten dar.

Die für den Datenimport erstellten Importskripte senden Abfragen an die Ursprungsdatenquellen und transformieren die zurückgelieferten Daten in Klassen, Instanzen und Relationen der Ontologie. Auf diese Weise wurden beispielsweise Daten aus der Personendatenbank ausgelesen und in der Ontologie als einzelne Personeninstanzen und ihnen zugeordnete Relationen wie *vocab:hasResearchTopic* und *swrc:email* abgebildet.

Die Kategorien der Pädagogik-Systematik wurden als Unterklassen der Klasse *swrc:ResearchTopic* angelegt, denen die einzelnen Wörterbuchterme als Instanzen zugeordnet werden. Beim Einlesen der in der Personendatenbank verzeichneten Forschungsgebiete der einzelnen Personen soll ein stringbasierter Abgleich mit diesen Forschungsgebieten stattfinden. Die daraus resultierende Verknüpfung zwischen Instanzen der Klassen *swrc:Person* und *swrc:ResearchTopic* integriert die Personendaten in einen größeren semantischen Kontext.

Die ontologiebasierte Integration von Daten aus zwei verschiedenen Quellen wird anhand von Abbildung 3 beispielhaft veranschaulicht.

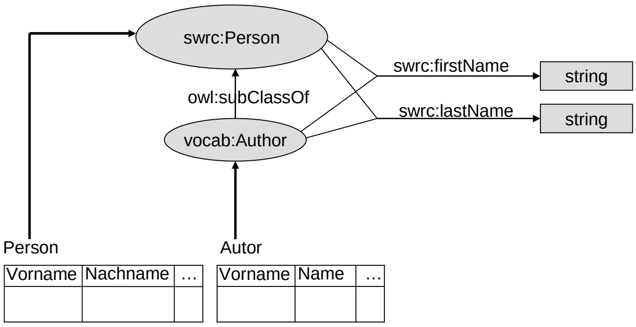


Abb. 3: Beispiel für die Überführung von Daten aus den Ausgangsdatenquellen in die Ontologie

Aus unterschiedlichen Quellen stammende Autoren- und Personendaten sollen über die Ontologie zusammengeführt werden. Dass Autoren eine Untergruppe der Klasse *swrc:Person* darstellen, wird in der Ontologie abgebildet, indem die Klasse *vocab:Author* als Unterklasse von *swrc:Person* modelliert wird. Letztere erbt daher die Relationen *swrc:firstName* und *swrc:lastName* der Oberklasse. Autorendaten und Personendaten aus verschiedenen Ursprungsdatenquellen werden auf diese Weise in einen semantischen Zusammenhang gebracht.

Auf der Basis der Ontologie sollen jedoch nicht nur strukturelle Heterogenitäten zwischen den einzelnen Datenquellen aufgelöst und eine semantische Integration der Daten erzielt werden, sondern auch eine stärkere Verknüpfung der Daten wird angestrebt. Dies lässt sich durch den Einsatz von Reasonern realisieren.

4.4 Reasoning

Um die vorliegende Datenbasis durch die Generierung neuen Wissens anzureichern, können so genannte Reasoner eingesetzt werden. Sie greifen bei der Wissensgenerierung auf Inferenzregeln zurück, die in einer Regelsprache definiert werden müssen.

In unserem Anwendungskontext könnte eine solche Regel beispielsweise festlegen, dass eine Person, die ein Dokument zu einem bestimmten Thema publiziert hat, mit diesem Thema über die Relation *hasResearchTopic* verknüpft wird. Diese Regel würde beispielsweise in der „Semantic Web Rule Language“ (SWRL) folgenderweise formalisiert werden [O’Connor *et al.* 2005]:

hasPublished(?x, ?y)
∧ *hasTopic(?y, ?z)*
→ *hasResearchTopic(?x, ?z)*

In der Ontologie würden nach der Aktivierung des Reasoners folglich sämtliche Verknüpfungen zwischen Autoren und Forschungsinhalten hergestellt werden, die aus der Analyse resultieren, mit welchen Themen sich die Publikationen der einzelnen Autoren beschäftigen.

Das Reasoning kann darüber hinaus auch für Zwecke der Datenbereinigung eingesetzt werden. Legt man beispielsweise fest, dass eine Emailadresse lediglich einer Person zugeordnet sein darf, so können zwei Objekte mit derselben Adresse als identisch identifiziert werden. Der Einsatz derartiger Regeln kann sich insbesondere bei der Zusammenführung von Daten aus unterschiedlichen Quellen als sehr hilfreich erweisen.

4.5 Entwicklung einer Benutzerschnittstelle

Auf dieser ontologiebasierten integrierten Datenbasis soll der Forschungsführer als Informationsportal konzipiert werden, das sowohl eine ontologiebasierte Browsing-Möglichkeit als auch Retrievalkomponenten für bestimmte Suchszenarien bereitstellt. Diese Komponenten der Benutzerschnittstelle werden im Folgenden kurz beschrieben.

(1) Ontologiebasiertes Browsing

Informationen über Personen, Forschungsinhalte und Institutionen sollen auf Profilseiten dargestellt werden, die untereinander über ihre ontologischen Beziehungen verlinkt werden. Auf diese Weise wird das ontologiebasierte Browsing realisiert.

(2) Logikbasierte Suche

Die Retrievalkomponente soll eine logikbasierte Suche umfassen, welche es ermöglicht, anhand eines Pulldown-Menüs wie dem folgenden Anfragen an die Ontologie zu definieren:

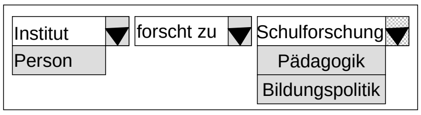


Abb. 4: Logikbasierte Suche

(3) Darstellung ontologischer Verknüpfungen

Bei der Freitextsuche nach den Namen zweier Objekte der Ontologie soll es darüber hinaus möglich sein, die ontologischen Verknüpfungen dieser Objekte darzustellen. So würde die Eingabe zweier Personennamen A und B zu einer Ergebnispräsentation in der Form von Verknüpfungsketten wie der folgenden führen:

A forscht zum Thema P
P ist Unterthema von Q
B forscht zum Thema Q

Auf diese Weise sollen Suchfunktionalitäten bereitgestellt werden, welche die ontologische Organisation der Daten nutzen, um dem Informationssuchenden einen Mehrwert gegenüber der Recherche in heterogenen Datenbankbeständen zu ermöglichen.

5 Erwarteter Mehrwert

Durch die ontologiebasierte Datenintegration wird dem Informationssuchenden ein integrierter Zugriff auf die verteilten Datenbestände ermöglicht. Dies wird erreicht, indem Daten aus verschiedenen Datenbanken konzeptbasiert zusammengeführt werden. Durch die in Abschnitt 4.3 geschilderte Modellierung der Relation *swrc:email* wird es beispielsweise möglich, nach der Email einer Person zu suchen, ohne die Existenz aller verfügbaren Emailversionen abfragen zu müssen. Stattdessen kann direkt auf die subsumierende Relation *swrc:email* zugegriffen werden.

Das im vorigen Abschnitt angeführte Beispiel zur Darstellung ontologischer Verknüpfungen veranschaulicht, wie die Ontologie dazu eingesetzt werden kann, dem Nutzer Zusammenhänge zwischen einzelnen Informationsobjekten zu präsentieren. Die Abfrage derartiger Zusammenhänge aus einer oder mehreren Datenbanken würde hingegen voraussetzen, dass der Nutzer bereits eine Vorstellung über die Art der Verknüpfung zwischen den Objekten besitzt. Durch die ontologiebasierte Informationsintegration wird es möglich, diese Zusammenhänge auch Datenquellen übergreifend darzustellen.

Der Einsatz eines Reasoners führt zudem dazu, dass Informationen abgefragt werden können, die in den Ursprungsdaten nicht explizit enthalten sind. Mithilfe der logikbasierten Suche könnte beispielsweise nach einer Person gesucht werden, die zu einem bestimmten Thema forscht. Während die Ursprungsdatenbank eventuell keinen Personendatensatz enthalten würde, welche den Suchterm im Feld „Forschungsgebiet“ aufführt, können über die neu ermittelten Verknüpfungen in der Ontologie auch diejenigen Personen ermittelt werden, die zu dem angegebenen Forschungsthema publiziert haben und für den Nutzer ebenfalls von Relevanz sind.

Das ontologiebasierte Browsing soll es dem Nutzer ermöglichen, in dem Forschungsführer frei zu navigieren. Der Nutzen dessen ontologischer Struktur lässt sich an einem Szenario wie dem folgenden illustrieren: der Nutzer betrachtet das Profil eines Forschers und klickt dessen Forschungsgebiet an, um sich über weitere Forscher aus diesem Bereich zu informieren. Gleichzeitig werden ihm auf der Basis der Ontologie auch verwandte Forschungsbereiche wie Teilgebiete und Obergebiete angezeigt. Auf diese Weise wird der Nutzer sowohl bei der Verbreiterung als auch bei der Spezifizierung seiner Informationssuche unterstützt.

Derartige ontologische Verknüpfungen in der Form von Hyponymen, Hyperonymen und Synonymen sind auch für die Generierung von anfragesensitiven Suchvorschlägen einsetzbar. Integriert in eine Freitextsuchkomponente wird dem Nutzer damit die Möglichkeit geboten, seine Suchanfrage zu reformulieren oder zu skalieren.

Auch eine automatische Query Expansion-Funktionalität lässt sich durch die Nutzung der in der Ontologie hinterlegten Synonymrelationen in die Freitextsuche integrieren.

Durch die Nutzung von Standardvokabularen wird darüber hinaus die spätere Interoperabilität mit externen Semantic Web-Anwendungen angestrebt.

6 Evaluierung

Die abschließende Evaluierung verfolgt das Ziel, die Funktionalitäten des Forschungsführers anhand von Nutzungsszenarien zu bewerten. Einer internen Studie zufolge schreibt etwa ein Drittel der Fachportal-Nutzer den dort verfügbaren erziehungswissenschaftlichen Informationen über Experten und Institutionen ein hohes Gewicht zu[8].

Im Rahmen von Nutzertests sollen daher praxisnahe Recherchеszenarien in Form von Use Cases eingesetzt werden, um das Retrieval auf den verteilten Ausgangsdatenquellen mit der Recherche im Forschungsführer-Portal zu vergleichen. Hierbei sind insbesondere die Aspekte der Informationsaufbereitung und –anreicherung sowie der Retrievalmöglichkeiten zu bewerten. Ziel ist es zu eruieren, ob der Einsatz semantischer Technologien im geschilderten Anwendungskontext einen Mehrwert für den Nutzer darstellt.

7 Fazit

Der vorliegende Beitrag hat den geplanten Aufbau eines ontologiebasierten Forschungsführers für den Fachbereich Bildungsforschung geschildert. Es wurde dargestellt, wie Semantic Web-Technologien eingesetzt werden sollen, um mehrere heterogene Datenquellen des Informationszentrums Bildung semantisch zu integrieren und den Informationssuchenden darauf basierend Mehrwert bringende Recherchefunktionalitäten anbieten zu können.

Literatur

[Auer und Lehmann 2007] S. Auer, J. Lehmann. What Have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. 4th European Semantic Web Conference (ESWC 2007), Innsbruck, Österreich.

[Baerisch 2007] S. Baerisch. Heterogenität in wissenschaftlichen Fachdatenportalen. In: A. Oßwald, M. Stempfhuber, C. Wolff: Open Innovation. Neue Perspektiven im Kontext von Information und Wissen. Konstanz: UVK Verlagsgesellschaft, 509-518.

[Bambey und Jornitz 2006] D. Bambey, S. Jornitz. Fachportal Pädagogik – Recherche und mehr. Buch und Bibliothek, 57(4): 336-337, 2006.

[Baumgartner *et al*. 2005] R. Baumgartner, N. Henze, M. Herzog. The Personal Publication Reader: Illustrating Web Data Extraction, Personalization and Reasoning for the Semantic Web. 2nd European Semantic Web Conference (ESWC 2005), Heraklion, Griechenland.

[Berners-Lee *et al.* 2001] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web. Scientific American, 284(5): 34-43, Mai 2001.

[Bizer und Seaborne 2004] C. Bizer, A. Seaborne. D2RQ - Treating Non-RDF Relational Databases as Virtual RDF Graphs. 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan.

[Jung *et al*. 2007] H. Jung, M. Lee, W. Sung, D. Park. Semantic Web-Based Services for Supporting Voluntary Collaboration among Researchers Using an Information Dissemination Platform. Data Science Journal, 6: 241-249, 2007.

[Mika 2005] P. Mika. Flink: Semantic Web technology for the extraction and analysis of social networks. Journal of Web Semantics, 3(2-3): 211-223.

[O'Connor *et al.* 2005] M. O'Connor, H. Knublauch, S. Tu, B. Grosof, M. Dean, W. Grosso, M. Musen. Supporting Rule System Interoperability on the Semantic Web with SWRL. 4th International Semantic Web Conference (ISWC 2005), Galway, Irland.

[Stempfhuber 2003] M. Stempfhuber. Objektorientierte Dynamische Benutzungsoberfläche - ODIN. Bonn: GESIS Informationszentrum Sozialwissenschaften.

[Sure *et al.* 2005] Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, D. Oberle. The SWRC Ontology - Semantic Web for Research Communities. 12th Portuguese Conference on Artificial Intelligence - Progress in Artificial Intelligence (EPIA 2005), Covilha, Portugal, Dezember 2005.

[8] D. Bambey: Das Evaluationskonzept des Fachportal Pädagogik: Einsatz von Evaluationsmethoden und Erstellung eines Wiki-Leitfadens. Vortrag auf der ISI 2007 & IuK 2007 am 01.06.2007. http://www.iuk2007.de/fileadmin/user_upload/iuk_abstracts.html#bambey [zuletzt besucht am 22.08.2007].

Eine empirische Studie zur Suchmaschinenoptimierung am Beispiel eines Firmenauftritts

Julia Maria Schulz, Christa Womser-Hacker, Thomas Mandl
Informationswissenschaft, Universität Hildesheim
Marienburger Platz 22
D-31141 Hildesheim, Deutschland
mandl@uni-hildesheim.de

Abstract

Suchmaschinenoptimierung zielt ab auf die erhöhte Sichtbarkeit von Internet-Seiten in den Trefferlisten von Suchmaschinen. Der Suchmaschinenoptimierung wird im Information Retrieval zunehmend mehr Beachtung geschenkt. Eine beispielhafte Evaluierung für 250 Seiten eines Firmenauftritts zeigte, dass mehrere von den Betreibern tolerierte Maßnahmen bei zwei unterschiedlichen Suchmaschinen zu erheblichen Verbesserungen der Trefferpositionen führen. Die gleichzeitige Durchführung mehrerer Maßnahmen führt zu besseren Erfolgen als die Summe von Einzelmaßnahmen. Auch die Modifikation der Linkstruktur einer Seite kann sich positiv auf die Positionen in Trefferlisten auswirken. Zwar kann der Erfolg für eine einzelne Seite nicht sicher gestellt werden, jedoch erhöht sich der durchschnittliche Trefferplatz deutlich.

1 Einleitung

Suchmaschinenoptimierung (*search engine optimization*, SEO) stellt ein Teilgebiet des Information Retrieval dar, bei dem die Perspektive der Informationsanbieter im Zentrum steht. Anbieter im Internet wollen, dass ihre Informationen gut gefunden werden. Da der Zugriff der Benutzer häufig über Suchmaschinen erfolgt, versuchen die Anbieter ihre Seiten so zu gestalten, dass diese in den Trefferlisten möglichst auf guten Positionen auftauchen. Das Ziel der Suchmaschinenoptimierung besteht also darin, die Wahrscheinlichkeit zu erhöhen, dass bestimmte Seiten in den Trefferlisten von Suchmaschinen auftreten.

Die Anbieter von SEO müssen dies ohne genaue Kenntnis der in den Suchmaschinen angewandten Algorithmen erreichen. Diese Algorithmen stellen einen großen Teil des Know-How der Suchmaschinenbetreiber dar und zum anderen haben die Betreiber ein zwiespältiges Verhältnis zur Suchmaschinenoptimierung. Zwar geben sie in gewissem Umfang Hilfestellung[1], andererseits wünschen sie keine zu erfolgreiche Suchmaschinenoptimierung und halten die genauen Algorithmen intransparent. Sie argumentieren, dass bei genauer Kenntnis Täuschungsversuche (Spam) sehr viel leichter möglich wären und Benutzer dann zu Seiten geleitet würden, welche die Suchmaschine über ihre eigentlichen Inhalte täuscht. Gerade Link-Spamming hat in den letzten Jahren zu zahlreichen Anstrengungen geführt [Fetterly et al. 2004]. Darüber hinaus vermarkten Suchmaschinen bezahlte Anzeigen und wollen diesen Markt nicht durch zu einfache Suchmaschinenoptimierung obsolet machen [Grappone and Couzin 2006].

Suchmaschinenoptimierung selbst hat sich in den letzten Jahren nicht zuletzt zu einem Geschäftsmodell entwickelt. Seriöse empirische Untersuchungen und Evaluierungsansätze sind noch kaum zu finden.

2 Suchmaschinenoptimierung

Maßnahmen zur Suchmaschinenoptimierung lassen sich in on-page und off-page Techniken unterteilen. Zu den off-page Maßnehmen zählen die Auswahl des Domain-Namens, die Anpassung von Verzeichnis- und Dateinamen und vor allem die Link-Popularität. Vor allem Qualitätsmetriken auf der Basis von eingehenden Links wie beispielsweise PageRank gelten als wichtiges Kriterium für das Ranking in Suchmaschinen [Mandl 2006]. Dieser Beitrag fokussiert vorrangig auf on-page Maßnahmen. Dazu zählen Techniken, die sich aus dem traditionellen Lehrbuchwissen im Information Retrieval ableiten lassen [Womser-Hacker and Mandl 2007]. Diese betreffen beispielsweise die Häufigkeit der Suchbegriffe im Dokument und beeinflussen die Gewichtung des Begriffs im Index. Eine hohe inverse Dokumenthäufigkeit führt zu einer höheren Ranking-Position.

Darüber hinaus definieren Suchmaschinenbetreiber einige Techniken als unerwünscht und damit als sogenannten Spam. Diese Methoden täuschen einen anderen Inhalt vor als letztlich auf den Seiten zu finden ist. Für einige dieser Methoden haben die Suchmaschinen mittlerweile sehr effiziente Algorithmen zur Aufdeckung entwickelt, so hat bspw. jede Suchmaschine intern einen Höchstwert für die Keyword Density festgelegt, bei dessen Überschreitung die Seite als Spam identifiziert wird. Der Grad der Überschreitung spielt bei diesen Algorithmen ebenfalls eine Rolle, eine geringe Überschreitung wird meist nicht geahndet. Stellt eine Suchmaschine mehrere Spam-Methoden auf einer Website fest, so führt dies meist zum Ausschluss aus dem Index [Moritz 2005].

3 Experimentelles Vorgehen

Für die empirische Analyse der Wirksamkeit der Suchmaschinenoptimierung wurde eine Erlebnisgeschenke-Firma[2] gewählt. Die Produktseiten dieser Firma eignen sich besonders gut für eine derartige Analyse, da das breite Spektrum an Erlebnissen, die aus diversen Bereichen stammen, sehr unterschiedliche Keywords für die Suchmaschinen-

[1] http://www.google.com/support/webmasters/

[2] http://www.yamando.net

optimierung fordern. Suchmaschinen stellen immer nur eine begrenzte Anzahl von Seiten einer Domain in den Ergebnislisten zu einer auf eine Suchanfrage dar, weil sie dem Benutzer eine möglichst große Auswahl an Ergebnissen zur Verfügung stellen möchten. Um eine empirische Studie durchzuführen, werden daher Seiten benötigt, die auf unterschiedliche Keywords bzw. Keyword-Phrasen hin optimiert werden können, da bei der Optimierung von Seiten mit ähnlichen Produkten, die durch dasselbe Keyword beschrieben werden, nur wenige in den Ergebnislisten angezeigt werden und eine Verfolgung der Rankingverbesserung daher nicht ermöglichen.

3.1 Hilfsmittel der SEO

Als Werkzeug wurde WebCEO[3] gewählt. Das kostenpflichtige WebCEO liefert unter anderem Informationen zur täglichen Suchhäufigkeit eines Terms und ähnlicher Suchanfragen. WebCEO bietet wesentlich detaillierte Informationen als die meisten der kostenlos im Internet verfügbaren Werkzeuge. Des Weiteren können die Informationen sowohl für Google als auch MSN und Yahoo abgerufen werden. Ebenso stellt WebCEO die Anzahl der Dokumente, die dasselbe Keyword enthalten bereit. Damit liefert es ein Maß für die Bewertung des Wettbewerbs um einen Suchbegriff (competition). Zusätzlich berechnet WebCEO einen Keyword Effectiveness Index (KEI), der die *competition* und die Suchhäufigkeit eines Begriffes ins Verhältnis setzt. KEI gibt somit Auskunft darüber, welche Begriffe sich am besten für eine effektive Website-Optimierung eignen. Je höher der KEI ist, desto größer ist die Suchhäufigkeit und desto geringer ist die Anzahl der Mitbewerber.

3.2 Ziele und Phasen der SEO

Die Untersuchung evaluierte einige on-page Optimierungsmaßnahmen empirisch auf ihre Wirksamkeit. Unsere Studie beschränkte sich auf Maßnahmen, welche die Suchmaschinenbetreiber explizit tolerieren. Auf einzelnen Produktseiten wurden jeweils genau die Merkmale verändert, die eine bestimmte Maßnahme charakterisieren. Die Wirkung auf die Rankingposition wurde gemessen, indem jeweils vor und nach der Neuindexierung durch die Suchmaschinen die Rankingposition mit Hilfe des Programms WebCEO ermittelt wurde.

Am Beginn der Studie stand die Entscheidung, für welche Suchmaschinen optimiert werden sollte. Da die Gewichtung einzelner Kriterien und die daraus resultierenden Rankingpositionen in den Ergebnislisten bei jeder Suchmaschine unterschiedlich ausgeprägt ist resp. die optimalen Werte für einzelne Kriterien wie die Keyword Density von einander abweichen, ist diese Festlegung von zentraler Bedeutung. Diese Studie beschränkt sich auf die drei Suchmaschinen mit dem größten deutschen Marktanteil: Google, MSN und Yahoo.

Für jede Produktseite wurde ein Keyword bzw. eine Keyword-Phrase ermittelt, die den Inhalt der Seite treffend beschreibt und von den Suchmaschinennutzern genutzt werden könnte. Zu diesem Zweck erfolgte zunächst Brainstorming, um eine Auswahl an möglichen Keywords zu erhalten. Zusätzlich wurde die Synonymfunktion von Microsoft Word genutzt, um alternative Begriffe zu ermitteln. Des Weiteren wurde das „Research Keywords“ Tool von WebCEO verwendet. Es wurden vorwiegend Zwei- und Drei-Wort-Phrasen als Keywords ausgewählt, da diese zum einen statistisch gesehen die höchste Suchwahrscheinlichkeit aufweisen [Fischer 2006], zum anderen in der Regel keine so starke Konkurrenz wie einzelne Keywords aufweisen, die nur aus einem Wort bestehen. Ein weiterer Vorteil ist, dass die Seite zusätzlich für die einzelnen Terme der Keyword-Phrase optimiert wird. Für Seiten, die ähnliche Produkte beschreiben und daher die Verwendung der gleichen Keywords denkbar ist, wurden nach Möglichkeit unterschiedliche Keywords verwendet, da Suchmaschinen nur eine begrenzte Anzahl von Seiten einer Domain in der Ergebnisliste zu einem Keyword anzeigen. Das Keyword mit dem höheren KEI bleibt dabei dem Produkt vorbehalten, das häufiger verkauft wird.

Für die Produktseiten stellte diese Studie die erste SEO Manahmen dar. Die Betreiber pflegten Keywords gezielt auf den Seiten ein. In der ersten Optimierungsphase kamen Maßnahmen zum Einsatz, die das Vorkommen der Keywords auf den Produktseiten erfordern. Die Keyword Density wurde im Body-Tag, in den Metadaten erhöht und Keywords wurden im Body-Tag hervorgehoben. Dabei wurde eine Vorkommenshäufigkeit (Keyword Density) zwischen 2% und 3% angestrebt.

Des Weiteren wurde überprüft wie viel Einfluss das Verwenden des Title-Tags und der Metatags Description haben. Auf 50 Produktseiten wurden sie sowohl im Title-Tag als auch in den oben genannten Metatags eingepflegt. Um Synergieeffekte aufzudecken, erfolgten bei 50 weiteren Seiten alle Maßnahmen zusammen. Die Keywords gelangten in den Body-Tag, in den Title-Tag und in die Metatags Description und Keywords. Das Title-Element wurde zusammen mit den Metadaten auf seine Wirkung getestet. Metadaten bedeutet im Weiteren die Metatags Description und Keywords und den Title-Tag.

Neben den schon genannten Kriterien wird außerdem der Einfluss von Hervorhebungen im Text, in diesem Fall von fettgedruckten Keywords auf das Ranking getestet. Dazu fügten die Betreiber die Keywords auf jeweils 50 Produktseiten einmal fettgedruckt im Body-Tag hinzu.

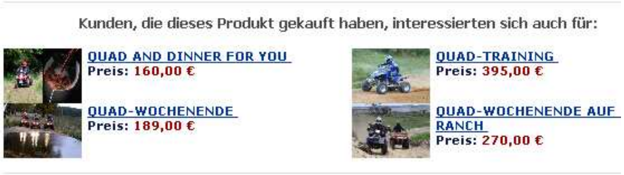
Für eine zusätzliche Menge von 50 Seiten wurde getestet, welche Synergieeffekte auftreten, wenn die oben genannten einzelnen Maßnahmen gemeinsam verwendet werden. Auf den Produktseiten werden die Keywords daher sowohl im Body-Tag und im Title-Tag sowie in den Metatags Description und Keywords verwendet und jeweils einmal durch Fettdruck im Body-Tag hervorgehoben.

Die zweite Optimierungsphase erhöhte die Keyword Density im Body-Tag von 2% bis 3% auf 3% bis 4%. Die dritte Optimierungsphase fokussierte auf die interne Verlinkung der Website. Suchmaschinen-Betreiber geben teilweise an, die Link-Analyse als Einflussfaktor für das Ranking zu bewerten [Mandl 2006]. Zu diesem Zweck erhielt jede Produktseite eine Empfehlung für vier weitere Produkte. Diese stellte je ein Bild und der Name des Produktes dar und beide Elemente verlinken auf das empfohlene Produkt. Diese Phase überprüfte zum einen, ob das Erhöhen der internen Verlinkung einen Einfluss auf das Ranking hat, und zum anderen, ob stärker verlinkte Produkte ebenfalls eine stärkere Rankingverbesserung aufweisen als diejenigen mit weniger Links. Zu diesem Zweck erhielten einige Produkte mehr Empfehlungen und damit interne Links als andere.

Im Anschluss an jede Optimierungsphase wurden die Rankingpositionen der jeweiligen Produktseiten erneut

[3] http://www.webceo.com

festgestellt, nachdem die Suchmaschinen die Seiten neu indexiert haben und die geänderten Seiten in den Index aufgenommen wurden. Auf diese Weise maß WebCEO die Effekte der Optimierungsmaßnahmen.



Kunden, die dieses Produkt gekauft haben, interessierten sich auch für:

QUAD AND DINNER FOR YOU
Preis: 160,00 €

QUAD-TRAINING
Preis: 395,00 €

QUAD-WOCHENENDE
Preis: 189,00 €

QUAD-WOCHENENDE AUF RANCH
Preis: 270,00 €

Abb. 1: Produktempfehlung durch interne Links

Die Konkurrenzseiten resp. Anzahl der Ergebnisse auf eine Suchanfrage weichen für die Keywords bzw. Keyword-Phrasen einzelner Produktseiten unterschiedlich stark voneinander ab, so dass die Veränderung der Rankingpositionen zunächst nicht vergleichbar sind. Um die Effekte einzelner Optimierungsmaßnahmen zu messen ist dies jedoch eine Vorraussetzung. Aus diesem Grund werden die Veränderungen der Rankingpositionen mit der Anzahl der gesamten Ergebnisse für die Suchanfrage ins Verhältnis gesetzt. Diese maximale Trefferzahl dient als Maß für den Wettbewerb um einen Suchbegriff. Dazu erfolgte eine Normalisierung unter Berücksichtigung der ersten 1000 Ergebnisse, die eine Suchmaschine üblicherweise anzeigt.

4 Ergebnisse

Die einzelnen Optimierungsphasen führten teilweise zu unterschiedlichen Ergebnissen. Die folgenden Abschnitt halten die Ergebnisse der drei Phasen fest.

4.1 On-Page Verfahren

Die Optimierung der Keyword Density im Body-Tag führte bei Google zu einer durchschnittlichen Rankingverbesserung von 340 Positionen, abhängig von der gewählten Suchform. Die Ergebnisse dieser Optimierungsmaßnahme weisen für Google sehr hohe Streuungen auf.

In Abb. 2 sind die Ergebnisse dieser Optimierungsphase für die deutschsprachige Suche dargestellt. Bei einem Drittel der Werte lagen sehr große Rankingverbesserungen mit Werten um 900 Positionen vor, während bei einem weiteren Drittel die Werte nahe Null liegen. Das bedeutet, dass das erste Drittel vor der Optimierungsmaßnahme eine Rankingposition jenseits der sichtbaren Ergebnisliste innehatte, und diese durch die Optimierung unter die ersten 100 Ergebnisse gelangt sind. Für das zweite Drittel ist entweder keine Verbesserungen der Rankingposition erzielt worden, oder diese konnte nicht gemessen werden, da die Produktseiten jenseits der ersten 1000 Ergebnisse gerankt waren. Die Messung der Rankingpositionen vor der Optimierungsmaßnahme ergab, dass ca. 67% der Seiten bei Google nicht innerhalb der sichtbaren Ergebnisliste, d.h. unter den ersten 1000 Ergebnissen, gerankt waren, daher ist hier von letzterem auszugehen.

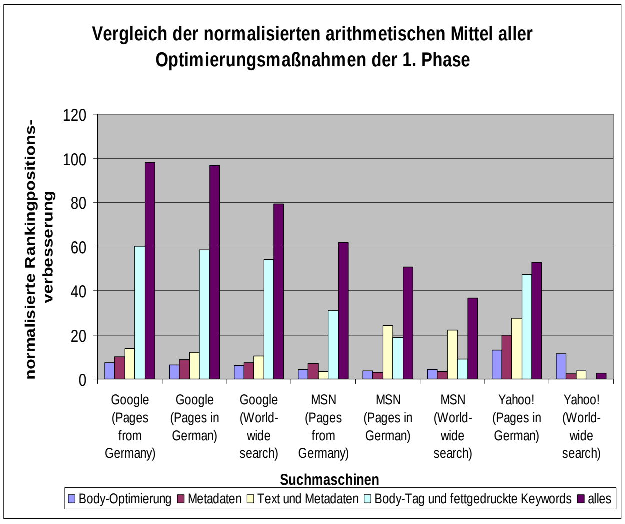


Abb. 2: Verteilung der Rankingverbesserung für Google, MSN und Yahoo bei der Suche in deutschsprachigen Dokumenten

Für die Suchmaschine MSN konnte nur eine minimale Rankingverbesserung gemessen werden. Das arithmeti-

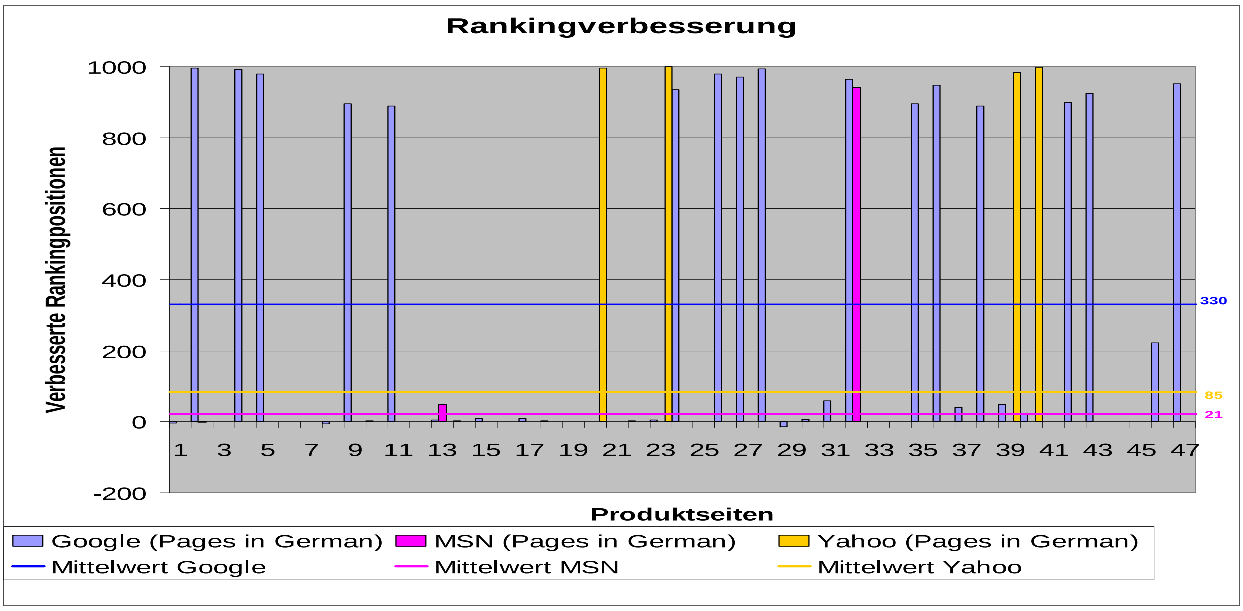


Abb. 3: Vergleich der normalisierten arithmetischen Mittel aller Optimierungsmaßnahmen der 1. Phase

sche Mittel hat hier Werte zwischen 21 und 25 in Abhängigkeit von der gewählten Suchform angenommen. Die Standardabweichung mit Werten zwischen 137 und 138 verdeutlicht auch hier die sehr starke Streuung der Ergebnisse.

Abbildung 2 zeigt, dass für die MSN-Suche in deutschsprachigen Dokumenten ein Ausreißer mit einem großen Wert auftrat. Für die übrigen Produktseiten lag bis auf einige wenige Ausnahmen keine Rankingverbesserung vor. Da die Kennzahlen der Verteilung der Rankingverbesserung für die drei Suchfunktionen von MSN ähnliche Werte aufweisen, ist davon auszugehen, dass die Verteilung für die weltweite Suche und die Suche in Dokumenten aus Deutschland ähnlich aussehen.

Für die beiden Suchmöglichkeiten der Suchmaschine Yahoo wurden durchschnittliche Rankingverbesserungen von 63 bzw. 85 Positionen erreicht, jedoch verursachten wenige Ausreißer dieses hohe arithmetische Mittel.

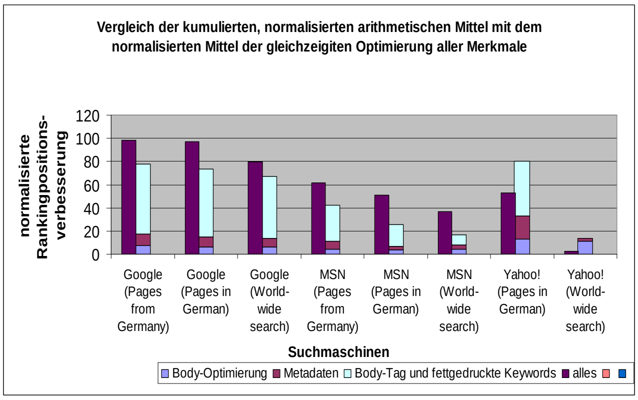


Abb. 4: Vergleich der kumulierten, normalisierten arithmetischen Mittel der Optimierung des Body-Tags, der Metadaten und des Body-Tags mit her vorgehobenem Keyword mit der Optimierung aller Merkmale

Die normalisierten Ergebnisse weisen ähnliche Verhältnisse zwischen arithmetischem Mittel und Standardabweichung auf wie die absoluten Ergebnisse. Bei der vorhergehenden Analyse trat also keine Verzerrung durch einen besonders hohe bzw. niedrige *competition* auf.

Die Analyse der übrigen Optimierungsmethoden der ersten Phase erfolgte analog. Die Ergebnisverteilungen für die Optimierung der Metadaten, der gemeinsamen Optimierung des Body-Tags und der Metadaten sowie die Optimierung des Body-Tags mit einem durch Fettdruck hervorgehobenen Keyword weisen eine ähnliche Struktur auf wie die der Optimierung der Keyword Density im Body-Tag. D.h. es konnte jeweils eine deutliche Verbesserung der Rankingposition bei Google, eine geringe Verbesserung bei MSN und keine Verbesserung bei Yahoo festgestellt werden.

Die Zielsetzung der letzten Optimierungsmethode der ersten Phase, bei der alle vorgenannten Optimierungsmaßnahmen gleichzeitig durchgeführt wurden, war das Aufzeigen eventuelle Synergieeffekte.

Abbildung 3 verdeutlicht, dass die gleichzeitige Optimierung aller Merkmale bei Google und MSN eine größere Rankingverbesserung bewirkt als die Optimierung einzelner Kriterien. Um zu überprüfen, ob ein Synergieeffekt vorliegt, werden in der folgenden Abbildung die normalisierten arithmetischen Mittel der Body-Optimierung, der Metadaten und der Optimierung mit fettgedruckten Keywords kumuliert neben dem der fünften Maßnahme dargestellt. Die gleichzeitige Optimierung des Body-Tags und der Metadaten wird nicht mit betrachtet, da die Wirkung der Keyword Density und der Metadaten sonst doppelt berücksichtigt würde.

Die kumulierten Mittelwerte der Optimierung des Body-Tags, der Metadaten und des Body-Tags mit einem im Text hervorgehobenen Keyword für die Suchmaschinen Google und MSN weisen je einen kleineren Wert auf, als die der gleichzeitigen Optimierung aller Merkmale wie Abbildung 4 zeigt. Es scheint daher ein Synergieeffekt für diese Suchmaschine vorzuliegen. Wegen der hohen Standardabweichung bei den einzelnen Merkmalen wird zusätzlich noch das Verhältnis der kumulierten Mediane der vorgenannten Optimierungsmaßnahmen zum Median der Optimierung aller Merkmale betrachtet.

Da der Wert der kumulierten Mediane der drei Optimierungsmaßnahmen für die Suchmaschine Google ebenfalls kleiner ist, als der Wert des Medians für die gleichzeitige Optimierung der Merkmale, liegt wie oben vermutet ein Synergieeffekt vor (vgl. Abb. 4). Ausgehend davon, dass die Mediane für MSN und Yahoo jeweils einen Wert von null aufweisen, kann hier nicht von einem Synergieeffekt ausgegangen werden.

4.2 On-Page Verfahren mit höherer Keyword-Dichte

Die zweite Phase der Optimierung erhöhte die Keyword-Dichte gegenüber der ersten Phase. Die 50 Produktseiten, bei denen in der ersten Phase der Body-Tag mit einer Keyword Density von 2% bis 3% optimiert wurde, erhielten im Title-Tag, in den Metatags Description und Keywords und im Inhalt erneut die entsprechenden Keywords resp. Keyword-Phrasen.

Ziel des erneuten Überprüfens der Metadaten ist es, den Einfluss von Metadaten bei im Body-Tag bereits optimierten Seiten zu ermitteln und die Ergebnisse der vorangegangenen Optimierungsphase zu verifizieren.

Die Analyse der Daten hat ergeben, dass lediglich für die Rankingpositionen bei Google eine Rankingverbesserung in der Mehrheit der Fälle eingetreten ist, da hier sowohl die Mediane als auch die Mittelwerte deutlich positive Werte aufweisen. Die Standardabweichung zeugt zwar von einer hohen Streuung um den Mittelwert, jedoch streuen die Werte stärker oberhalb des Mittelwertes wie die Schiefe der Verteilungen belegt.

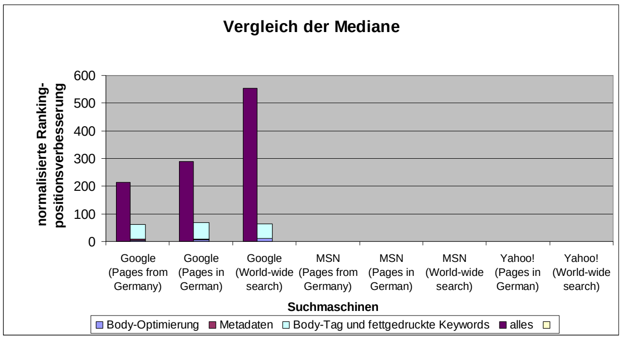


Abb. 5: Vergleich der kumulierten Mediane der Optimierung des Body-Tags, der Metadaten und des Body-Tags mit fettgedrucktem Keyword mit dem Median der Optimierung aller Merkmale

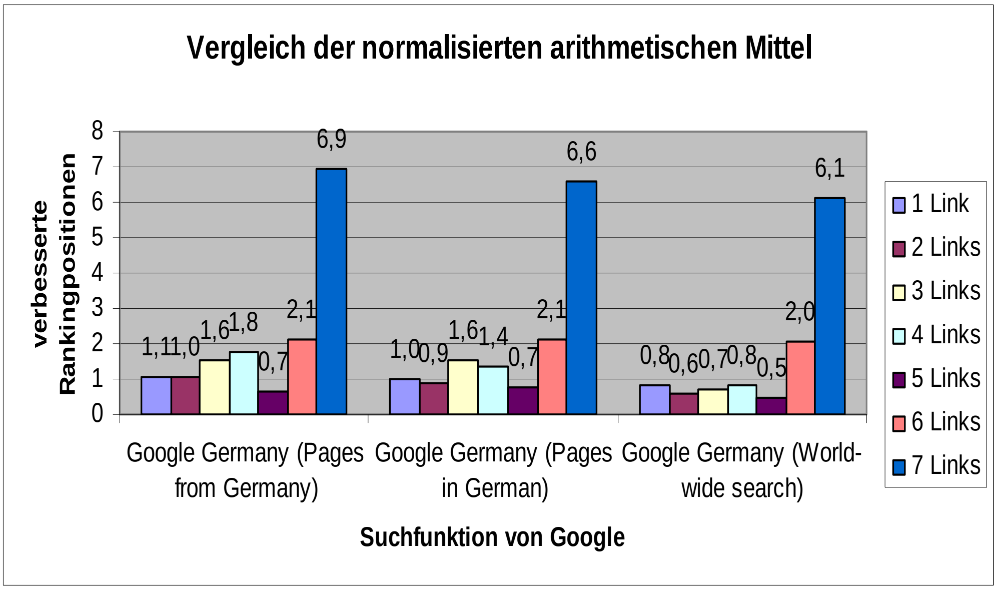


Abb. 6: Vergleich der normalisierten arithmetischen Mittel der Suchfunktionen von Google

Die Mittelwerte der Ergebnis für die Suchmaschinen Yahoo und MSN legen ebenfalls eine positive Veränderung der Rankingposition nahe, aber durch die hohen Streuungen um den Mittelwert, lässt sich keine zuverlässige Aussage darüber treffen, ob bei der Durchführung vorgenannter Optimierungsmaßnahme in jedem Fall eine Rankingverbesserung eintritt.

Da die Optimierung des Body-Tags in der ersten Optimierungsphase gezeigt hat, dass sehr wenige Produktseiten bei Yahoo und MSN unter den ersten 1000 Ergebnissen gerankt wurden, wird die Keyword Density auf 3% bis 4% erhöht. Des Weiteren wird mit dieser Maßnahme überprüft, ob bei Google die Produktseiten durch die höhere Keyword Density eine bessere Rankingposition erreichen. Ist dies der Fall, liegt die optimale Keyword Density für Google über 3%. Verschlechtern sich die Rankingpositionen so liegt die optimale Keyword Density für diese Suchmaschine zwischen 2% und 3%.

Das arithmetische Mittel deutet mit Werten zwischen 6 und 7 zunächst auf eine Verbesserung der Rankingposition bei Google hin. Bei der Betrachtung des Medians, der für alle Google Sucharten null beträgt, und der Standardabweichung, die mit 17, 19 und 20 im Verhältnis zum arithmetischen Mittel sehr hoch ist, zeigt sich, dass bei wenigen Produktseiten eine Verbesserung der Rankingposition eingetreten ist. Zieht man außerdem die normalisierten Werte hinzu, ergibt sich für die Suche in deutschen Dokumenten bzw. in Dokumenten aus Deutschland eine negative Rankingverbesserung. Diese resultiert daraus, dass bei der Normalisierung der Daten die Rankingverbesserungen mit der *competition* ins Verhältnis gesetzt werden. Daraus lässt sich schließen, dass relativ große positive Rankingverbesserungen bei einer großen Anzahl von Gesamtergebnissen erfolgt sind, und betragskleine negative Rankigverbesserungen bei einer geringen Anzahl von Gesamtergebnissen vorliegen. Eine Veränderung der Rankingposition um ein oder zwei Plätze nach oben oder unten ist jedoch nicht zwangsläufig auf die durchgeführte Optimierungsmaßnahme zurückzuführen. Sie kann ebenfalls dadurch zustande kommen, dass ein Mitbewerber seine Seite optimiert hat und deshalb eine bessere Rankingposition erreicht hat. Insgesamt ist bei den Produktseiten in den Ergebnislisten bei Google weder eine Verbesserung noch eine Verschlechterung der Rankingposition zu messen.

Bei MSN betragen die Werte des arithmetischen Mittels null, und die Standardabweichungen sind mit Werten um eins sehr klein. Die Erhöhung der Keyword Density wirkte sich bei MSN nicht auf die Rankingposition aus. Der Median betrug für alle drei MSN-Optionen ebenfalls null. Die normalisierten Ergebnisse zeigen sogar eine geringfügige Verschlechterung der Position an. Diese kann auf die Veränderungen eines Mitbewerbers zurückgeführt werden. Des Weiteren sind nur 37 von 43 Produktseiten in der sichtbaren Ergebnisliste vorhanden. Das Ziel, für die Produktseiten eine Rankingposition unter den ersten 1000 Ergebnissen zu erreichen, konnte nicht erreicht worden. Dies gilt ebenso für Yahoo.

4.3 Link-Struktur

Die dritte Phase der SEO optimierte die interne Verlinkung der Website. Ziel dieser Optimierungsphase war es festzustellen, ob

- die interne Verlinkung im Allgemeinen eine Verbesserung der Rankingposition bewirkt
- bei einer unterschiedlichen Anzahl von Links die Veränderung der Rankingposition ebenfalls variiert.

Zu diesem Zweck stellte jede Produktseite vier weitere Produkte vor, die manuelle ausgewählt worden waren, um die Anzahl der Links zu kontrollieren.

Um einen möglichen Zusammenhang zwischen der Anzahl der eingehenden Links und der Verbesserung der Rankingposition aufzudecken stellt Abbildung 6 die Kennzahlen für jede Suchmaschine getrennt dar.

In wird deutlich, dass insgesamt gesehen ein Aufwärtstrend der Rankingverbesserung für eine zunehmende Anzahl interner Links bei Google vorliegt. Die weltweite Suche weist bei bis zu fünf eingehenden Links noch keinen Aufwärtstrend auf, vielmehr schwankte die Rankingverbesserung um einen Wert von 0,7. Ab einer Anzahl von sechs eingehenden Links wird jedoch ein starker Aufwärtstrend deutlich. Eine zunehmende Anzahl von internen Links erwirkt ein besseres Ranking.

Bei den normalisierten arithmetischen Mitteln für die drei Suchfunktionen von MSN ließ sich ebenfalls ein stetiger Aufwärtstrend feststellen. Die Werte streuen um eine Gerade, die Abbildung 7 für jede Suchfunktion von MSN zeigt. Für Yahoo ergab sich keine signifikante Rankingverbesserung. Weitere Details der Evaluierung finden sich in [Schulz 2007].

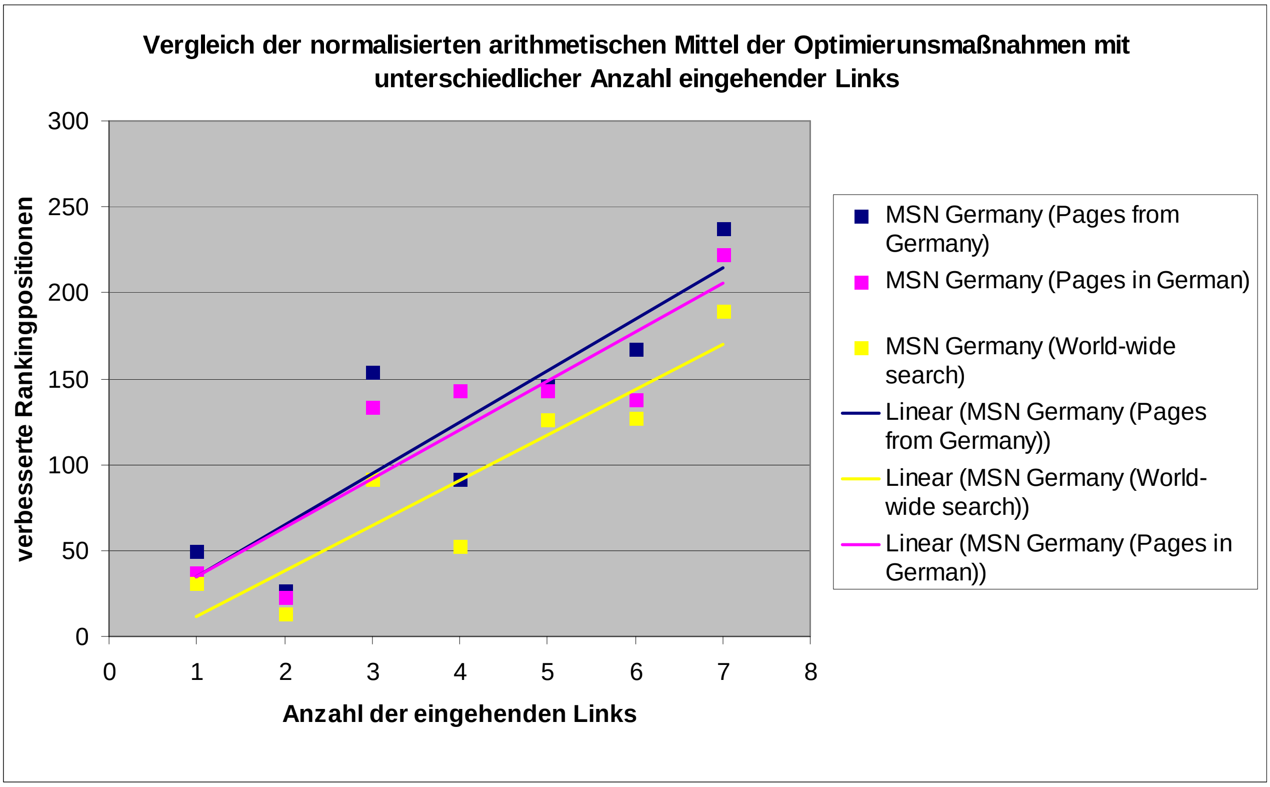


Abb. 7: Vergleich der normalisierten arithmetischen Mittel der unterschiedlichen Anzahl eingehender Links

5 Fazit und Ausblick

Die Ergebnisse zeigen, dass vor allen die Suchmaschine Google sehr sensitiv und zeitnah auf die Maßnahmen reagiert. Dagegen konnten bei Yahoo kaum Veränderungen gemessen werden. Von den Suchmaschinenbetreibern erlaubte und vorgeschlagene Maßnahmen auf der Seiten wie die häufigere Nennung des Suchbegriffs für den optimiert wird, verbessern die Position der Seiten in den Trefferlisten. Eine Keyword Density von 4% wertet Google noch positiv. Auch das Einfügen zusätzlicher site-interner Links auf zu optimierende Seiten führt zu Verbesserungen.

Die durchgeführte Studie zeigt, dass in die Bewertung auch der erreichte Platz in der Trefferliste mit eingehen sollte. In den Suchmaschinen betrachten die Benutzer oft nur die erste Seite der Ergebnisse und erwarten eine hohe Precision. SEO Maßnahmen welche zu Treffern unter den ersten zehn oder zwanzig Seiten führen sollten demnach höher gewichtet werden, als Rangverbesserungen um die gleiche Anzahl von Positionen auf hinteren Rängen.

Die Suchmaschinenoptimierung stellt ein informationsethisches Problem dar. Die Sichtbarkeit von Informationsangeboten kann über wirtschaftlichen Erfolg oder Misserfolg der Anbieter entscheiden. Suchmaschinenbetreiber legen fest, welche Praktiken sie für gerechtfertigt halten und welche nicht. Letztere sanktionieren sie durch Entfernen der Seiten aus dem Index. Derartige Sanktionen haben erhebliche Auswirkungen, sind jedoch in keiner Weise transparent, gesellschaftlich legitimiert oder anfechtbar.

Literatur

[Fetterly et al. 2004] Dennis Fetterly; Mark Manasse; Marc Najork: Spam, damn Spam, and Statistics. In *Proc of the Seventh International Workshop on the Web and Databases* (WebDB 2004) Paris. http://webdb2004.cs.columbia.edu/papers/1-1.pdf

[Fischer 2006] Mario Fischer. *Website Boosting. Suchmaschinen-Optimierung, Usability, Webseiten-Marketing*. 1. Aufl., Heidelberg: Mitp

[Grappone and Couzin 2006] Jennifer Grappone; Gradiva Couzin: *Search Engine Optimization: An Hour a Day*. Sybex 2006

[Komrey 1998] Helmut Komrey: *Empirische Sozialforschung*. 8. Aufl., Opladen: Leske und Budrich

[Mandl 2006] Thomas Mandl. Implementation and Evaluation of a Quality Based Search Engine. In *Proc 17th ACM Conference on Hypertext and Hypermedia (HT '06)* Odense, Denmark, August 22nd-25th. ACM Press. 2006. S. 73-84.

[Moritz 2005] André Moritz: *Suchmaschinen-Ranking optimieren*. Düsseldorf: Data Becker. 2005

[Schulz 2007] Julia Maria Schulz. Suchmaschinenoptimierung – Eine empirische Studie zur Optimierung des Rankings am Beispiel einer Erlebnisgeschenkefirma. *Magisterarbeit, Internationales Informationsmanagement, Universität Hildesheim*. 2007

[Womser-Hacker and Mandl 2007] Christa Womser-Hacker; Thomas Mandl: Information Retrieval. In *WISU: Das Wirtschaftsstudium*. 5/2007. S. 692-697.

Evaluation of an Information Retrieval System for the Semantic Desktop using Standard Measures from Information Retrieval

Peter Scheir[1,2], **Michael Granitzer**[2], **Stefanie N. Lindstaedt**[2]
[1]Graz University of Technology, Austria
peter.scheir@tugraz.at
[2]Know-Center Graz, Austria
slind@know-center.at

Abstract

Evaluation of information retrieval systems is a critical aspect of information retrieval research. New retrieval paradigms, as retrieval in the Semantic Web, present an additional challenge for system evaluation as no off-the-shelf test corpora for evaluation exist. This paper describes the approach taken to evaluate an information retrieval system built for the Semantic Desktop and demonstrates how standard measures from information retrieval research are employed for evaluation.

1 Semantic Web information retrieval and evaluation

Despite the youthfulness of Semantic Web information retrieval, a growing amount of proposed models and implemented systems, leading from indexing triples together with textual data [Shah *et al.*, 2002], over modeling documents as parts of knowledge bases [Zhang *et al.*, 2005], to ranking search results in semantic portals [Stojanovic *et al.*, 2001], exist. Nevertheless, Semantic Web information retrieval could benefit from the experience made in information retrieval system evaluation in the past 50 years of the information retrieval discipline.

Within this paper we give an example of how standard information retrieval measures can be applied to the evaluation of retrieval performance in a Semantic Desktop environment. We aim at providing a guideline the developers of systems for the Semantic Desktop on the one hand and raise the awareness about parallels of this new domain of information retrieval to classical information retrieval on the other hand.

At present information retrieval in the Semantic Web (on the Semantic Desktop) is an inhomogeneous field (c.f. [Scheir *et al.*, 2007b]. Although a good amount of approaches does exist, different information is used for the retrieval process, different input is accepted and different output is produced. This complicates to define generally applicable rules for the evaluation of an information retrieval system for the Semantic Web (or the Semantic Desktop) and to create a test collection for this application area of information retrieval.

This paper is structured as follows: in section 2 we briefly introduce the concept of the Semantic Desktop (section 2.1) and the characteristics of our system (section 2.2). In section 3 we present the test corpus used for system evaluation. In section 4 we talk about the evaluation of the system, which measures were used (section 4.1), the queries employed for evaluation (section 4.2), how we collected relevance judgments (section 4.3) and the ranking of system configurations we have obtained (section 4.4). Finally we discuss our approach to evaluation in section 5 and conclude with section 6.

2 The evaluated system

We have built an information retrieval system for the Semantic Desktop. We will now briefly introduce the concept of the Semantic Desktop and then focus on the characteristics of the evaluated system. A detailed description of the system can be found in [Scheir *et al.*, 2007a][1]. In this paper we treat the system as a black-box and only elaborate on the input and output values of the system.

2.1 Semantic Desktop

The Semantic Desktop [Sauermann *et al.*, 2005] [Decker and Frank, 2004] paradigm stems from the Semantic Web [Berners-Lee *et al.*, 2001] movement and aims at applying technologies developed for the Semantic Web to desktop computing. In recent years the Semantic Web movement led to the development of new, standardized forms of knowledge representation and technologies for coping with them such as ontology editors, triple stores or query languages. The Semantic Desktop founds on this set of technologies and introduces them to the desktop to ultimately provide for a closer integration between (semantic) web and (semantic) desktop.

2.2 Characteristics of the evaluated system

The evaluated system relies on both, information in an ontology and the statistical information in a collection of documents. The system is queried by a set of concepts from the ontology and returns a set of documents. Documents in the system are (partly) annotated with ontological concepts if a document *deals with* a concept. For example, if the document is an introduction to use case models it is annotated with the corresponding concept in the ontology. The annotation process is performed manually but is supported by statistical techniques (e.g. identification of frequent words in the document collection) [Pammer *et al.*, 2007].

Concepts from the ontology are used as metadata for documents in the system. Opposed to classical metadata, the ontology specifies relations between the concepts. For example, class-subclass relationships are defined as well

[1] Available online under: `http://www.know-center.tugraz.at/media/files/wissensbilanz/publications_wm/papers/2007_scheir_improving_search_on_the_semantic_desktop_pdf` (01.09.2007)

as arbitrary semantic relations between concepts are modeled (e. g. `UseCase isComposedOf Action`). The structure of the ontology can be utilized for calculating the similarity between two concepts in the ontology. This similarity can be used to extend a query by similar concepts before retrieving documents dealing with a set of concepts. After retrieval of documents was performed, the result set can be extended by means of textual similarity. Different combinations of query and result expansion were evaluated against each other.

3 The test corpus

A major obstacle in the easy evaluation of Semantic Web technology based information retrieval systems is the absence of standardized test corpora, as they exist for text-based information retrieval.

Therefore we have built our own test corpus based on the data available in the first release of the APOSDLE system [Lindstaedt and Mayer, 2006]. The first version of APOSDLE was built for the domain of Requirements Engineering. This resulted into a domain ontology for this field and a set of documents dealing with various topics of Requirements Engineering. The document base was provided by a partner in the APOSDLE project, with expertise in the field of Requirement Engineering, while the ontology was modeled by another partner. Together these two partners sign responsible for the annotation of the document base with concepts from the ontology. The ontology contains 70 concepts and the document set consists of 1016 documents. 496 documents were annotated using one or more concepts. 21 concepts from the domain ontology were used to annotate documents.

In its size our test collection is comparable to test collections from early information retrieval experiments as the Cranfield or the CACM collections.

In addition to the absence of corpora for Semantic Web information retrieval we are unaware of any standard text-retrieval corpora for evaluating a system with characteristics similar to ours. We considered treating the ontological concepts used for querying our system equivalent to query terms of a text-retrieval system to be able to use a standard corpus. Therefore we would have needed some structure relating the terms contained in the documents, as it is the case with the ontology in our system which relates concepts. For this task we could have used a standard thesaurus. As this knowledge structure is different to the ontology originally used (and therefore different similarity measures had to be applied to it), this would have led us to evaluating a system with different properties than our original one.

We also considered the INEX[2] test collection for evaluating our system. INEX provides a document collection of XML documents which would have provided us with textual data associated with XML structure information. Unfortunately again an ontology relating the metadata used as XML markup is unavailable. This would have prevented us from employing (and evaluating) the functionality provided by the query expansion technique, which founds on the ontology.

4 Evaluation

In this section we describe the evaluation that we performed. We talk about the evaluation measures, the queries used for evaluation, how we collected relevance judgments and about the system configuration rankings obtained.

4.1 Measures used for evaluation

The central problem in using classic IR measures as *recall* or *mean average precision* is that they require complete relevance judgments, which means that every document is judged against every query [Buckley and Voorhees, 2004]. [Fuhr, 2006] notices that recall can not be determined precisely with reasonable effort. Finally [Carterette *et al.*, 2006] states that: *Building sets large enough for evaluation of realworld implementations is at best inefficient, at worst infeasible.*

Therefore we opted for using evaluation measures that do not require hat every document is judged against every query. We decided for using precision (P) at rank 10, 20 and 30. In addition we made use of infAP [Yilmaz and Aslam, 2006] which approximates the value of average precision (AP) using random sampling.

For calculating the evaluation scores we have used the `trec_eval`[3] package, which origins from the Text REtrieval Conference (TREC) and allows for calculating a large number of standard measures for information retrieval system evaluation.

4.2 Queries used for evaluation

The queries that were used for the evaluation of the system are formed by sets of concepts.

The first version of the APOSDLE system presents resources to knowledge workers to allow them to acquire a certain competency. To realize search for resources that are appropriate to build up a certain competency, competencies are represented by sets of concepts from the domain ontology. These sets are used as queries for the search for resources. For the evaluation of the APOSDLE system all distinct sets of concepts representing competencies[4] were used as queries. In addition all concepts from the domain model not already present in the set of queries were used for evaluation purposes.

4.3 Collecting relevance judgments

8 different system configuration were tested and compared against each other based on the chosen evaluation measures. 79 distinct queries were used to query every system configuration. Queries were formed by sets of concepts stemming from the domain ontology.

For every query and system configuration the first 30 results were stored in a database table, with one row for every query-document pair. Query-document pairs returned by more than one system configuration were stored only once. The query-document pairs stored in the database-table were then judged manually by a human assessor. All query-document pairs were judged by the same person. The assessor was not involved in defining the competency to concept mappings uses as queries (c.f. section 4.2.

After relevance judgment, both, the results obtained by the different system configurations and the global relevance judgments have been stored into text files in a format appropriate for the `trec_eval` program. We then calculated the

[2] http://inex.is.informatik.uni-duisburg.de/

[3] http://trec.nist.gov/trec_eval/

[4] different competencies can be represented by the same concepts

P(10), P(20), (P30) and infAP scores for the different system configurations.

4.4 The obtained system configuration ranking

Table 1 shows the calculated P(10), P(20), (P30) and infAP scores for the different system configurations. Table 2 shows the system configuration ranking based on the obtained evaluation scores.

Configuration 1 (conf_1) is the baseline configuration of our system. It equals a database query with a ranked list of results. Results to the query are ranked using an idf-like measure between concepts and documents.

All other configurations make use of query expansion based on semantic similarity or result expansion based on text-based similarly. Configurations 3, 4, 5, 6, 7 and 8 perform query expansion. Configurations 2, 6, 7 and 8 perform result expansion.

Configuration 4 and 5 are essentially the same approach with minor internal differences. The same holds for configuration 7 and 8.

Conf.	P(10)	P(20)	P(30)	infAP
conf_1	0.2418	0.2051	0.1700	0.1484
conf_2	0.3089	0.2778	0.2502	0.2487
conf_3	0.3165	0.2608	0.2131	0.2114
conf_4	0.3114	0.2582	0.2097	0.2001
conf_5	0.3114	0.2582	0.2097	0.2000
conf_6	0.3848	0.3405	0.3046	0.3253
conf_7	0.3924	0.3494	0.3089	0.3326
conf_8	0.3911	0.3487	0.3080	0.3318

Table 1: Evaluation scores of system configurations calculated using P(10), P(20), P(30) and infAP

Rank	P(10)	P(20)	P(30)	infAP
1 (best)	conf_7	conf_7	conf_7	conf_7
2	conf_8	conf_8	conf_8	conf_8
3	conf_6	conf_6	conf_6	conf_6
4	conf_3	conf_2	conf_2	conf_2
5	conf_4	conf_3	conf_3	conf_3
6	conf_5	conf_4	conf_4	conf_4
7	conf_2	conf_5	conf_5	conf_5
8 (worst)	conf_1	conf_1	conf_1	conf_1

Table 2: Ranking of system configurations based on P(10), P(20), P(30) and infAP

5 Discussion

We now discuss the evaluation measures used and why we think that the amount of relevance judgments collected is sufficient for a proper evaluation of our system.

5.1 P(10), P(20) and P(30)

[Buckley and Voorhees, 2000] evaluate the stability of evaluation measures. They calculate the error rate of measures based on the number of errors occurring whilst comparing two systems using a certain measure. They divide the number of errors by the total number of possible comparisons between two different systems. Based on previous research they state that an error rate of 2.9% is minimally acceptable. They find that P(30) exactly reaches this error rate of 2.9% in their experiment with 50 queries used. Finally they suggest that the amount of queries should be increased for P(n) measures, where $n < 30$. And suggest that 100 queries would be safe if the measure P(20) is used.

We performed our experiment with 79 distinct queries and used the measures P(10), P(20) and P(30). Following the results of [Buckley and Voorhees, 2000] the size of our query set should be appropriate for P(30). We are fortified in this assumption as the ranking of the 8 system configurations is identical for P(20), P(30) and infAP.

5.2 infAP

The Trec 8 Ad-Hoc collection consists of 528,155 documents and 50 queries which make a total amount of 26,407,750 possible relevance judgments. 86830 query-document relevance pairs are actually judged. This set of pairs is created by depth-100 pooling of 129 runs. Therefore 0.33% of the possible relevance judgments are performed.

Our collection consists of 1026 documents and 79 queries, which results in a total of 81,054 possible relevance judgments. This set of pairs is created by depth-30 pooling of 8 runs and 498 additional relevance judgments that were performed for runs that were not part of the experiment. 1938 query document pairs were actually judged. Therefore 2,39% of all possible relevance judgments were performed.

The depth-100 pool for the 8 evaluated runs would consist of 4138 query-document pairs. As we judged 1938 query-document pairs, we judged 46,83% of our potential depth-100 pool. [Yilmaz and Aslam, 2006] report a Kendall's tau based rank correlation of above 0.9 between infAP and AP with as little as 25% of the maximum possible relevance judgments of the depth-100 pool of the Trec 8 Ad-Hoc collection. They consider two rankings with a rank correlation of above 0.9 as equivalent.

With 46,83% of our potential depth-100 pool judged, we are confident that the infAP measure produces an estimation sufficiently accurate. Again our confidence in the results of infAP is assured by the equivalence of the ranking of the 8 system configurations for P(20), P(30) and infAP.

6 Conclusion

We have evaluated an information retrieval system for the Semantic Desktop using standard measures for information retrieval system evaluation. As classic measures for evaluation as recall and average precision require that every document is judged for every query we have chosen precision at ranks 10, 20 and 30 as evaluation measures. In addition we made use of the random sampling approach performed by the infAP measure. We are confident that our chosen approach reflects the actual relation between the system configurations as the ranking of the system configurations remains identical for the measures P(20), P(30) and infAP.

Acknowledgments

We thank the anonymous reviewers of our submission for their constructive feedback.

This work has been partially funded under grant 027023 in the IST work programme of the European Community.

The Know-Center is funded by the Austrian Competence Center program Kplus under the auspices of the Austrian Ministry of Transport, Innovation and Technology (www.ffg.at/index.php?cid=95) and by the State of Styria.

References

[Berners-Lee *et al.*, 2001] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, May 2001.

[Buckley and Voorhees, 2000] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA, 2000. ACM Press.

[Buckley and Voorhees, 2004] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA, 2004. ACM Press.

[Carterette *et al.*, 2006] Ben Carterette, James Allan, and Ramesh Sitaraman. Minimal test collections for retrieval evaluation. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275, New York, NY, USA, 2006. ACM Press.

[Decker and Frank, 2004] Stefan Decker and Martin R. Frank. The networked semantic desktop. In *WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, 2004.

[Fuhr, 2006] Norbert Fuhr. Information Retrieval: Skriptum zur Vorlesung im SS 06, 19. Dezember 2006, 2006.

[Lindstaedt and Mayer, 2006] Stefanie N. Lindstaedt and Harald Mayer. A storyboard of the aposdle vision. In *Innovative Approaches for Learning and Knowledge Sharing, First European Conference on Technology Enhanced Learning, EC-TEL 2006, Crete, Greece, October 1-4, 2006*, pages 628–633, 2006.

[Pammer *et al.*, 2007] Viktoria Pammer, Peter Scheir, and Stefanie Lindstaedt. Two protégé plug-ins for supporting document-based ontology engineering and ontological annotation at document level. In *10th International Protégé Conference - July 15-18, 2007 - Budapest, Hungary*, 2007.

[Sauermann *et al.*, 2005] Leo Sauermann, Ansgar Bernardi, and Andreas Dengel. Overview and outlook on the semantic desktop. In *Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference*, 2005.

[Scheir *et al.*, 2007a] Peter Scheir, Chiara Ghidini, and Stefanie N. Lindstaedt. Improving search on the semantic desktop using associative retrieval techniques. In *Proceedings of I-SEMANTICS 2007 (accepted for publication)*, 2007. Available online under: `http://www.know-center.tugraz.at/media/files/wissensbilanz/publications_wm/papers/2007_scheir_improving_search_on_the_semantic_desktop_pdf` (01.09.2007).

[Scheir *et al.*, 2007b] Peter Scheir, Viktoria Pammer, and Stefanie N. Lindstaedt. Information retrieval on the semantic web - does it exist? In *LWA 2007, Lernen - Wissensentdeckung - Adaptivität, 24.-26.9. 2007 in Halle/Saale (in this volume)*, 2007.

[Shah *et al.*, 2002] Urvi Shah, Tim Finin, and Anupam Joshi. Information retrieval on the semantic web. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 461–468, New York, NY, USA, 2002. ACM Press.

[Stojanovic *et al.*, 2001] Nenad Stojanovic, Alexander Maedche, Steffen Staab, Rudi Studer, and York Sure. Seal: a framework for developing semantic portals. In *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001), October 21-23, 2001, Victoria, BC, Canada*, pages 155–162, 2001.

[Yilmaz and Aslam, 2006] Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 102–111, New York, NY, USA, 2006. ACM Press.

[Zhang *et al.*, 2005] Lei Zhang, Yong Yu, Jian Zhou, ChenXi Lin, and Yin Yang. An enhanced model for searching in semantic portals. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 453–462, New York, NY, USA, 2005. ACM Press.

Workshop on Knowledge and Experience Management

at the LWA 2007 (Lernen–Wissen–Adaption) in Halle/Saale, 24.–26.9.2007

Objectives

The workshop on Knowledge and Experience Management is held as part of the LWA Workshop series and is organized by the Special Interest Group on *Knowledge Management (FGWM)* of the German Computer Science Society (GI).

The Special Interest Group on Knowledge Management addresses automated methods for the capture, the development, the utilization, and the maintenance of knowledge for organizations and networks of people. The main goal of the workshop is the exchange of innovative research ideas and experiences with practical applications in the area of knowledge and experience management.

Thereby, it aims to provide an interdisciplinary forum for researchers and practitioners for exchanging ideas and innovative applications with respect to knowledge and experience management.

We encouraged submissions describing ongoing research efforts as well as demonstrations of recent research software prototypes. The topics of interests of the FGWM workshop series are:

- Approaches for experience and knowledge management (e.g., case–based reasoning, logic–based approaches, text–based approaches, semantic portals)
- Applications of experience and knowledge management (e.g., corporate memories, e–commerce, design, tutoring/e–Learning, e–Government, software engineering, robotics, medicine)
- (Semantic) web services for knowledge management
- Agile approaches of knowledge management
- Agent-based & peer-to-peer knowledge management
- Just-In-Time Retrieval and Just-In-Time Knowledge Capturing
- Knowledge representation (e.g., ontologies, similarity, retrieval, adaptation)
- Support for formalization and maintenance of knowledge
- Evaluation of/in knowledge management systems
- Practical experiences (”Lessons–Learned”) of IT-based approaches
- Integration of knowledge management and business processes

Presentations

The following workshop proceedings include in total nine original submissions reporting on basic and applied research. Furthermore, we have included the abstracts of four previously published papers that were also presented at the FGWM workshop.

Acknowledgments

We would like to thank all authors for submitting papers to the workshop and the reviewers for giving helpful comments on the submissions within a very short time period. Our thanks go also to the organizers and supporters of the LWA workshop series, especially Alexander Hinneburg.

Program Committee

- Klaus-Dieter Althoff, Universität Hildesheim
- Joachim Baumeister, Universität Würzburg
- Ralph Bergmann, Universität Trier
- Harald Holz, DFKI GmbH
- Ioannis Iglezakis, DaimlerChrysler AG
- Michael Kohlhase, Jacobs University Bremen
- Mirjam Minor, Universität Trier
- Markus Nick, Fraunhofer IESE
- Ulrich Reimer, FHS St. Gallen
- Bodo Rieger, Universität Osnabrück
- Thomas Roth-Berghofer, DFKI GmbH
- Martin Schaaf, Universität Hildesheim
- Rainer Schmidt, Universität Rostock
- Gerhard Schwabe, Universität Zrich
- Steffen Staab, Universität Koblenz-Landau
- York Sure, AIFB Universität Karlsruhe

Contact: http://www.fgwm.de

Joachim Baumeister & Martin Schaaf
August 2007

Original Submissions

1. Applying Case-Based Reasoning for Missing Medical Data in ISOR
   Rainer Schmidt, Olga Vorobieva
2. Semantic Perspectives on Knowledge Management and E-Learning
   Andrea Kohlhase
3. Towards Improving Interactive Mathematical Authoring by
   Ontology-driven Management of Change
   Normen Müller, Marc Wagner
4. A Domain Independent System Architecture for Sharing Experience
   Kerstin Bach, Meike Reichle, Klaus-Dieter Althoff
5. Visualizing Patient Similarity in Clinical Decision Support
   Alexey Tsymbal, Martin Huber, Sonja Zillner, Tamás Hauer, Kevin Zhou
6. TCR Textual Coverage Rate
   Kerstin Bach, Alexandre Hanft
7. panta rhei
   Christine Müller, Michael Kohlhase
8. Managing Variants in Document Content and Narrative Structures
   Michael Kohlhase, Achim Mahnke, Christine Müller
9. Know the Right People? Recommender Systems for Web 2.0
   Sergej Sizov, Stefan Siersdorfer

Re–Submissions

1. CheckMATE – Erfahrungsmanagement für ServiceRoboter zur Realisierung von
   Self-Healing in Produktionsanlagen [Projektbericht]
   Markus Nick, Sören Schneickert, Jürgen Grotepaß
2. Schlagwort ”Experience Management”
   Markus Nick, Klaus-Dieter Althoff, Ralph Bergmann
3. Using Knowledge Wikis to Support Scientific Communities
   Joachim Baumeister, Jochen Reutelshoefer, Karin Nadrowsk, Axel Misok
4. Representation and Structure-Based Similarity Assessment for Agile Workflows
   Mirjam Minor, Alexander Tartakovski, Ralph Bergmann

Applying Case-Based Reasoning for Missing Medical Data in ISOR

Rainer Schmidt and Olga Vorobieva
University of Rostock
D-18055, Rostock, Germany
rainer.schmidt@uni-rostock.de

Abstract

In this paper, a CBR approach that deals with missing data is presented. In the conversational ISOR system different knowledge sources are involved, including medical experts. In the case base rules and formulae are stored that support the restoration of numerous missing values. The task is to restore missing values in an observed medical data set. The presented method is used for a set of physiological and biochemical measurements of dialysis patients. The measurements were taken at four time points during a year in which the patients participated at an especially developed physical training program. To analyse the obtained data a restoration of missing values is really necessary.

1 Introduction

Databases with many variables have specific problems. Since it is very difficult to overview their content, usually a user a priory does not know how complete the set is. Are any data missing? How many of them and where are they located?

We have a set of observed medical data and want to extract as much information as possible out of it. Extraction is planed stepwise. At each step we put and solve a data-mining problem. The first problem, "why do some exceptional patients do not fit a specific hypothesis", was presented in our paper at the last Knowledge Management Conference in Potsdam [Vorobieva, Rumyantsev, and Schmidt, 2007]. When solving this problem, we were confronted with further problems, especially with a missing data problem.

In this paper, we deal with data that are missing randomly and without any regularity. There can be different reasons for randomly missing data. Generally, the most frequent causes are input failures, misprints, lost data, and data that were not measured at all. Here we deal with a situation where the main cause is that many measurements did not happen. Since many data were missing, we want to fill in these empty spaces in our data set.

We assume that the data set contains groups of interdependent variables but a priory we do not know how many such groups there are, what kind of variables are dependent, and in which way they are dependent. However, we intend to make use of all possible forms of dependency to restore missing data.

One possibility to obtain information from observed data is modelling, especially creating mathematical models. Our way of modelling is a combination of a relatively simple initial model with a rather demanding CBR approach [Vorobieva, Rumyantsev, and Schmidt, 2007]. It requires main factors (selected from the list of observed parameters) to set up an initial model and additional data, especially from parameters not used in the model, to find an explanation for exceptional cases.

Therefore, the more complete our observed data base is, the easier it is to find explanations and the better the explanations should be. Otherwise, missing data force a researcher either to shorten the sample by excluding patients with missing data or to select those parameters as main factors with few missing data, which are not necessarily the most important parameters.

Data analysis methods are often valued according to its tolerance to missing data as in [McSherry, 2001]. In principle, there are two main approaches to the missing data problem. The first approach is the direct restoration of missing values. Statistical restoration of missing value is usually based on non-missing values from other records. A missing value is substituted by some function of existent values for example it can be an average of some values.

The second approach suggests methods that accept the absence of some data. Those methods can be differently advanced, from simply excluding cases with missing values up to rather sophisticated statistical models [Little and Rubin, 1987; Fleiss, 1986].

Gediga and Düntsch [Gediga and Dürtsch, 2003] propose to use CBR to restore missing data. Since their approach does not require any external information, they call it a non-invasive imputation method. Missing data are supposed to be replaced by their correspondent values of the most similar retrieved cases [Gediga and Dürtsch, 2003].

Why don't we just apply statistical methods? Firstly, statistical methods require homogeneity of the sample. We have no reasons to expect the set of our patients to be a homogenous sample. Instead, we are going to find out the common tendency of the sample, fix all exceptions of the main tendency and study every exception separately. Our observed data has the additional advantage that many variables were measured. From given values for many variables sometimes missing values can be calculated or estimated. Therefore, for calculation of missing values we do not use given values of other cases but given values of the actual case.

Secondly, statistical methods are developed for a closed information system. It means that no external knowledge is considered besides those containing in the statistical model, whereas in ISOR external knowledge can be considered. In fact, ISOR was especially designed to make use of various external knowledge sources.

2 Dialysis and Fitness

We deal with medical data observed on a group of dialysis patients. Dialysis means stress for a patient's organism and has significant adverse effects. Fitness is the most available and a relative cheap way of support. It is meant to improve a physiological condition of a patient and to compensate negative dialysis effects. At our University clinic in St. Petersburg, a specially developed complex of physiotherapy exercises including simulators, walking, swimming etc. was offered to all dialysis patients but only some of them actively participated, whereas some others participated but were not really active. The purpose of this fitness offer was to improve the physical conditions of the patients and to increase the quality of their lives.

One of the intended goals is to convince the patients of the positive effects of fitness and to encourage them to make efforts and to go in for sports actively. This is important because dialysis patients usually feel sick, they are physically weak, and they do not want any additional physical load [Davidson et al., 2005].

The theoretical hypothesis is that actively participating in the fitness program improves the physical condition of dialysis patients. Instead of reliable theoretical knowledge and intelligent experience, we have just this theoretical hypothesis and a set of measurements. In such situations the usual question is, how do measured data fit to theoretical hypotheses. To statistically confirm a hypothesis it is necessary, that the majority of cases fit the hypothesis. Mathematical statistics determines the exact quantity of necessary confirmation [Kendall and Stuart, 1979]. However, usually a few cases do not satisfy the hypothesis. We examine them to find out why they do not satisfy the hypothesis. Our system, ISOR, offers a dialogue to guide the search for possible reasons in all components of its data system. These exceptional cases belong to the case base. This approach is justified by a certain mistrust of statistical models by doctors, because modelling results are usually unspecific and "average oriented" [Hai, 2002], which means a lack of attention to individual "imperceptible" features of concrete patients.

In our former work [Vorobieva, Rumyantsev, and Schmidt, 2007] we made first steps to combine classical statistical methods with Case-based Reasoning. A rather simple but easily interpretable statistical model and a limited number of all measured parameters were used. In [Vorobieva, Rumyantsev, and Schmidt, 2007] a binary model is described that involves just three selected main variables. It is reasonable to assume that such a simple model based on very few variables should have some inaccuracies. However, it is supposed to express the general tendency, especially as we managed to overcome most model inaccuracies by applying CBR techniques and considering additional measured parameters from the observed data set.

The idea to combine CBR with other methods is not new. For example Care-Partner resorts to a multi-modal reasoning framework for the co-operation of CBR and Rule-based Reasoning (RBR) [Bichindaritz, Kansu, and Sullivan, 1998]. Another way of combining hybrid rule bases with CBR is discussed by Prentzas and Hatzilgeroudis [Prentzas and Hatzilgeroudis, 2002]. The combination of CBR and model-based reasoning is discussed in [Shuguang, Qing, and George, 2000]. Statistical methods are used within CBR mainly for retrieval and retention (e.g. [Corchado et al., 2003; Rezvani and Prasad, 2003]). Arshadi proposes a method that combines CBR with statistical methods like clustering and logistic regression [Arshadi and Jurisica, 2005].

2.1 Data

For each patient a set of physiological parameters is measured. These parameters contain information about burned calories, maximal power, oxygen pulse (volume of oxygen consumption per heartbeat), lung ventilation and many others. Furthermore, there are biochemical parameters like haemoglobin and other laboratory measurements. All these parameters are supposed to be measured four times during the first year of participating in the fitness program. There is an initial measurement followed by a next one after three months, then after six months, and finally after a year. Since some parameters, e.g. the height of a patient, are supposed to remain constant within a year, they were measured just once. The other ones are regarded as factors with four grades, they are denoted as F0 – the initial measurement of factor F, and F3, F6, and F12 – the measurements of factor F after 3, 6, and 12 months.

All performed measurements are stored in the observed database, which contains 150 records (one patient – one record) and 460 variables. 12 variables are constants the other 448 variables represent 112 different parameters.

The factors can not be considered as completely independent from each other, but there are different types of dependency among specific factors. Even a strict mathematical dependency can occur, as e.g. in this triple: time of controlled training, work performed during this time and average achieved power, expressed as Power = Work/Time. Less strict are relations between factors of biochemical nature. For example, an increase of parathyroid hormone implies an increase of blood phosphorus.

Those and many other principally existent relations between factors enable us to control recorded measurements and to fill in numerous missing data in the data set. Unfortunately, not all planned measurements took really place, some of them were made somehow wrong, and some measurements were wrongly recorded or even lost. Therefore, the observed database contains many missing and definitely wrong data.

It is necessary to note that measurements of dialysis patients essentially differ from measurements of parameters of non-dialysis patients, especially of healthy people, because dialysis destroys the natural relationships between physiological processes in an organism. In fact, for dialysis patients all physiological processes behave abnormal. Therefore, the correlation between parameters differs too. For statistics, this means difficulties in applying statistical methods based on correlation and it limits the usage of a knowledge base developed for normal people.

2.2 The Role of ISOR

ISOR has been developed as conversational system that helps a medical expert to find explanations for exceptional cases that do not fit the (mostly statistical) model. Its role is to organise an exchange of information among all "members" of the conversation. Two members are humans: a medical expert and the "case-based reasoner" (the system developer). The other "members" are various databases, the two main ones are the observed data set and the case base.

We did not attempt to create a medical knowledge base, because the list of observed variables is rather long and there are many possible relationships among the variables. A knowledge base that contains information about possible relations would be too large. Furthermore, we do not know in advance which variables required for the model and/or further research may have missing data. A complete knowledge base should contain all possible relations between the variables, whereas just a small part is required.

ISOR is a dialogue system that co-operates with a medical expert, who – when urgent - provides ideas, suggestions, and theoretical knowledge. So the expert can be seen as a source of external medical knowledge, but not in the traditional way as a knowledge provider to build up a knowledge base.

From the CBR point of view, the help of the medical expert is required at the adaptation stage. Since his help is considered as "expensive", the adaptation process should be as much automatically as possible.

The second human member of the conversation is the program developer, called the "case-based reasoner". Of course, this conversation part is not restricted to one person. Its role is to assist during the adaptation, especially to make verbal statements of the medical expert understandable for ISOR. This is not just a translation into predefined ISOR inputs but the "case-based reasoner" may have to make changes in the knowledge sources of the program.

3 Restoration of Missing Data

In ISOR, CBR is applied to restore missing data, the calculated values are filled in the observed database. The whole knowledge is contained in the case base, namely in form of solutions of former cases.

Since the number of cases is rather limited, a statistical restoration does not seem to be appropriate. So, there are three types of numerical solutions: exact, estimated, and binary. Some examples and restoration formulas are shown in table 1. All types of solutions are demonstrated by examples below in section 3.1.

When a missing value could be completely restored, it is called exact solutions. Exact solutions are based on other parameters. A medical expert has defined them as specific relations between parameters. He has done it during the use of ISOR. The developer has translated them as input for ISOR. As soon as they have been used once, they are stored in the case base of ISOR and can be retrieved for further cases.

Since estimated solutions are usually based on domain independent interpolation, extrapolation, or regression methods, a medical expert is not involved. An estimated solution is not considered as full reconstruction but just as estimation.

Missing parameter	Type of solution	Description	Time points
PTH	Binary	If P(T) >= P(t) then PTH(T) >= PTH(t) Else PTH(T) < PTH(t)	0 and 6
HT	Exact	HT = 100 * (1–PV/0.065 * Weight)	6
HT	Estimated	Y(6) = Y(3)*0.66 + Y(12) * 0.33	3 and 12
WorkJ	Exact	WorkJ = MaxPower * Time * 0.5	12
BC	Exact	BC = BF * BV	12
Oxygen pulse	Estimated	Linear regression	0 and 3 and 12

Table 1. Some solutions and of restoration formulas. Abbreviations: BC = Breath consumption, BF = Breath frequency, BV = Breath volume, HT = Hematocrit, P = Phosphorus, PTH = Parathyroid hormone, PV =plasma volume

A binary solution is a partly reconstruction of a missing value. Sometimes ISOR is not able to construct neither an exact nor an estimated solution, but the expert may draw a conclusion about increasing/decreasing of the missing value. So, a binary solution expresses just the assumed trend. "1" means that the missing value should have increased since the last measurement, whereas "0" means that it should have decreased. Binary solutions are used in the qualitative models of ISOR [Vorobieva, Rumyantsev, and Schmidt, 2007]. Of course, binary solutions are rather poor for reasoning. However, even such a restoration may help the expert user as a sort of indication.

3.1 Examples

By three typical examples we want to demonstrate how missing data are restored in the ISOR system.

First example: Exact solution.
The value of hematocrit (HT) after 6 months is missing. Hematocrit is the proportion of the blood volume that consists of red blood cells. So, the hematocrit measurements are expressed in percentage.

The retrieved solution (the third line of table 1) requires two additional parameters, namely plasma volume (PV) and the weight of the patient. For the query patient these values (measured after six months) are "weight = 74 kg and PV = 3,367". These values are inserted in the formula and the result is a hematocrit value of 30%.

This restoration is domain dependent, it combines three parameters in such a specific way that it can not be applied to any other parameter. However, the formula can of course be transformed in two other ways and so it can be applied to restore values of PV and the weight of the patient. The formula contains specific medical knowledge that was once given as a case solution by an expert.

Second example: Estimated solution.
It is the same situation as in the first example. The value of hematocrit that should have been measured after six months is missing. Unlike the first example, now the PV value that is required to apply the domain dependent formula is also missing. Since no other solution for exact calculation can be retrieved, ISOR attempts to generate an estimated solution. Of course, estimated solutions are not as good as exact ones but are acceptable. ISOR retrieves a domain independent formula (fourth line of table 1) that states that a missing value after six months should calculated as the sum of two-thirds of the value measured after three months and one-third of the value measured after twelve months. This general calculation can be used for many parameters.

Third example: Binary solution.
The value of parathyroid hormone (PTH) after six months is missing and shall be restored. The retrieved solution involves the initial PTH measurement and the additional parameter phosphorus (P), namely the measurement after six months, P(6), and the initial measurement, P(0). Informally, the solution states that with an increase of phosphorus goes along an increase of PTH too. More formal the retrieved solution states:

```
If P(6) >= P(0)
        then PTH(6) >= PTH(0)
        else PTH(6) < (PTH(0)
```

So, here a complete restoration of the missing PTH value is not possible but just a binary solution that indicates the trend, where "1" stands for an increase and "0" for a decrease.

3.2 Applying Case-Based Reasoning

In ISOR, cases are mainly used to explain further exceptional cases that do not fit the initial model. Just a sort of secondary application is the restoration of missing data. The solutions given by the medical expert are stored in form of cases so that they can be retrieved for solving further missing data cases. Such case stored in the case base has the following structure.

1. Name of the patient
2. Diagnosis
3. Therapy
4. Problem: missing value
5. Name of the parameter of the missing value
6. Measurement time point of the missing value
7. Formula of the solution (the "description column of table 1)
8. Reference to the internal implementation of the formula
9. Parameters used in the formula
10. Solution: Restored value
11. Type of solution (exact, estimated, or binary)

Since the number of stored cases is rather small, there are no real retrieval problems. The retrieval is performed by keywords. The four main keywords are: Problem code (here: "missing value"), diagnosis, therapy, and time period. As an additional keyword the parameter where the value is missing can be used. Solutions that are retrieved by using the additional parameter keyword are domain dependent. They contain medical knowledge that has been provided by the medical expert. The domain independent solutions are retrieved by using just the four main keywords.

What happens when the retrieval provides more than one solution? Though only very few solutions are expected to be retrieved at the same time, only one solution should be selected. At first ISOR checks whether the required parameters values of the retrieved solutions are available. A solution is accepted if all required values are available. If more than one solution is accepted, the expert selects one of them. If no solution is accepted, ISOR attempts to apply the one with the fewest required parameter values.

Each sort of solution has its specific adaptation. A numerical solution is just a result of a calculation according to a formula. This kind of adaptation is performed automatically. If all required parameter values are available, the calculation is performed and the query case receives its numerical solution.

The second kind of adaptation modifies a restoration formula. This kind of adaptation can not be done entirely automatically but the expert is involved. When a (usually short) list of solutions is retrieved, ISOR at first checks whether all required values of the exact calculation formulae are available. If required parameter values are not available, there are three alternatives to proceed. First, to find an exact solution formula where all required parameter values are available, second to find an estimation formula, and third to attempt to restore the required values too. Since for the third alternative there is the danger that this might lead to an endless loop, this process can be manually stopped by pressing a button in a dialogue menu. When for an estimated solution required values are missing, ISOR asks the expert.

The expert can suggest an exact or an estimated solution. Of course, such an expert solution has also to be checked for the availability of the required values.

However, the expert can even provide just a numerical solution, a value to replace the missing data – with or without an explanation of this suggested value.

Furthermore, adaptation can be differentiated according to its domain dependency. Domain dependent adaptation rules have to be provided by the expert and they only applicable to specific parameters. Domain independent adaptation uses general mathematical formulae that can be applied to many parameters. Two or more adaptation methods can be combined.

In ISOR a revision occurs. It is the attempt to find better solutions. An exact solution is obviously better than an estimated one. So, if a value has been restored by estimation and later on (for a later case) the expert has provided an appropriate exact formula, this formula should be applied to the former case too. Some estimation rules are better than other. So it may happen that later on a more appropriate rule is incorporated in ISOR. In principle holds, the more new solution methods are included into ISOR, the more former already restored values are attempted to revise.

Artificial cases. Since every piece of knowledge provided by a medical expert is supposed to be valuable, ISOR saves it for future use. If an expert solution cannot be used for adaptation for the query case (required values for this solution might be missing), the expert user can generate an artificial case. In ISOR exists a special dialogue menu to do this. Artificial cases have the same structure as real ones, and they are also stored in the case base.

3.3 Results

Since ISOR is a dialogue system and the solutions are generated within a conversation mainly between the system and the user, the quality of the solutions does not only depend on ISOR but also on an expert user.

To test our method we deleted a random set of parameter values from the observed data set. Subsequently, we applied our method and attempted to restore the missing values of this simulated data set.

So far, we can just summarise how and how many missing values could be restored (table 2). Since for those 12 parameters that were only measured once and remain constant no values were missing, they are not considered in table 2. More than half of the missing values could be at least partly restored, nearly a third of the missing values could be completely restored, about 58% of restoration occurred automatically. However, 39% of the missing values could not be restored at all. The main reasons are that for some parameters no proper method is available and that specific additional parameter values are required that sometimes are also missing.

Number of Parameters	112
Number of values	448
Number of missing values	97
Number of completely restored values	29
Number of estimated values	17
Number of partly restored values (binary)	13
Number of automatically restored values	34
Number of expert assistance	25
Number of values that could not be restored	38

Table 2. Counts of missing and restored values.

4 Conclusion

In this paper, a CBR approach to the missing data problem is presented. Here, an application of the ISOR system to the problem of fitness and dialysis patients is shown. A statistical model is combined with Case-based Reasoning. The statistical model supports the hypothesis that fitness can improve the physical conditions of dialysis patients, whereas with the help of Case-based Reasoning the exceptional cases that contradict this hypothesis can be explained.

Unfortunately, many data are missing. Since the fewer data are missing the better the model, we attempt fill in the missing data. This is done in a conversational process between a medical expert, ISOR, and the system developer. Since the time of a medical expert is valuable, we attempt to make demands on him as less as possible. Only when absolutely necessary he is asked. We do not ask him to create a knowledge base, because it wood be much too time consuming. So, the main work is done by CBR.

In ISOR, all main CBR steps are performed: retrieval, adaptation, and revision. Retrieval (of usually a list of solutions) occurs by the help of keywords. Adaptation is an interactive process between ISOR, a medical expert, and the system developer. In contrast to many CBR systems, in ISOR revision plays an important role. The whole knowledge is contained in the case base, namely as solutions of former cases. No further knowledge base is required but just the knowledge we really need is stored in the case base.

Acknowledgments

We thank Professor Alexander Rumyantsev, Pavlov State Medical University, St.Petersburg, for his cooperation and we thank Professor Aleksey Smirnov, director of the Institute for Nephrology of St.Petersburg Medical University and Natalia Korosteleva, researcher at the same Institute for collecting and managing the data.

References

[Arshadi and Jurisica, 2005] Arshadi and IgorJurissica. Data Mining for Case-based Reasoning in high-dimensional biological domains. *IEEE Transactions on Knowledge and Data Engineering* ,17 (8):127-1137, 2005.

[Bichindaritz, Kansu, and Sullivan, 1998] Isabelle Bichindaritz, Emin Kansu, and Keith Sullivan. Case-based Reasoning in Care-Partner. *Proceedings of European Workshop on Case-Based Reasoning*, pages 334-345. Springer-Verlag, Berlin, 1998.

[Corchado et al., 2003] Juan M Corchado, Emilio S Corchado, Jim Aiken et al. Maximum likelihood Hebbian learning based retrieval method for CBR systems. *Proceedings of International Conference on Case-Based Reasoning*, pages 107-121, Springer-Verlag, Berlin, 2003.

[Davidson et al., 2005] AM Davidson, JS Cameron, J-P Grünfeld et al. (eds.). Oxford Textbook of Nephrology, Vol. 3. Oxford University Press, 2005.

[Fleiss, 1986] J. Fleiss. *The Design and Analysis of Clinical Experiments*. John Wiley & Sons, 1986.

[Gediga and Dürtsch, 2003] G. Gediga and I. Dürtsch. Maximum Consistency of Incomplete Data via Non-Invasive Imputation. *Artificial Intelligence Review*, 19 (1): 93-107, 2003.

[Hai, 2002] G.A. Hai. *Logic of Diagnostic and Decision Making in Clinical Medicine*. Politheknica publishing, St. Petersburg, 2002.

[Kendall and Stuart, 1979] *The advanced theory of statistics*. Macmillan publishing, New York, 1979.

[Little and Rubin, 1987] R. Little and D. Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 1987.

[McSherry, 2001] David McSherry. Interactive Case-Based Reasoning in sequential diagnosis. *Applied Intelligence*, 14 (1): 65-76, 2001.

[Prentzas and Hatzilgeroudis, 2002] Jim Prentzas and Ioannis Hatzilgeroudis. Integrating Hybrid Rule-Based with Case-Based Reasoning. Proceedings of European Conference on Case-Based Reasoning, pages 336-349. Springer-verlag, Berlin, 2002.

[Rezvani and Prasad, 2003] Sina Rezvani and Girijesh Prasad. A hybrid system with multivariate data validation and Case-based Reasoning for an efficient and realistic product formulation. *Proceedings of International Conference on Case-Based Reasoning*, pages 465-478, Springer-Verlag, Berlin, 2003.

[Shuguang, Qing, and George, 2000] L. Shuguang, J. Qing, C.George. Combining case-based and model-based reasoning: a formal specification. Proceedings of APSEC'00, pages 4-16, 2000).

[Vorobieva, Rumyantsev, and Schmidt, 2007] Olga Vorobieva, Alexander Rumyantsev, and Rainer Schmid*t*Development of an Explanation Model for Exceptional Cases. *Proceedings of Conference on Professional Knowledge Management*, pages 93-100, Potsdam, March 2007. GITO-Verlag, Berlin.

Semantic Perspectives on Knowledge Management and E-Learning

Andrea Kohlhase
Digital Media in Education (DiMeB)
University Bremen
D-28359 Bremen, Germany
a.kohlhase@jacobs-university.de

Abstract

Knowledge Management (KM) and E-Learning (EL) applications interface more and more as their objects of concern consist in 'captured knowledge' resp. 'learning objects', i.e. content. In this paper, we want to discuss the potential synergy from their fusion with respect to two area-specific difficulties: KM's Authoring Problem ("Where are the content authors?") and EL's Control Problem ("Who is in charge?"). In order to understand the underlying assumptions when developing and using such systems, we will point to the relevance of their interaction design (in contrast to interface design) and introduce the notion of "Semantic Interaction Design" requiring a closer look at the evaluation of software products. We will distinguish between the micro- and macro-perspective on KM and EL, particularly on the retrieval question of precision and recall and the motivational optimization problem for software adoption. We suggest that KM's Authoring Problem as well as EL's Control Problem are implications of above semantic perspectives. Moreover, it turns out that KM and EL's resp. strengths and weaknesses are complementary, so that they offer potential resolutions.

1 Introduction

Unfortunately, KM as well as E-Learning weren't as successful as expected (with occasional exceptions). Therefore a joint venture was undertaken to harvest synergy effects as both deal in their own specific way with content on the one hand and the user on the other. In this paper we address the question, which arises with respect to each field's specific problems: can the fusion remedy them?

Knowledge Management (KM) systems as well as E-Learning (EL) systems are built on *knowledge*[1] blocks that contain reified knowledge, i.e. information about knowledge (often called content or learning object (LO)). As objects these knowledge chunks can e.g. be managed, shared, reused, or aggregated - which are KM tasks. In contrast, as reified knowledge they can be used pedagogically as e.g. REINMANN declares them to be "*the link between learning and teaching*" [Reinmann, 2005, p. 117] - these are EL tasks. In particular, software can construct or help to construct learning contexts based on them: knowledge contexts (like ontologies or intersubjective knowledge), didactical contexts (like learning paths), or subjective contexts (like personal learning environments), for examples and ideas we suggest [Kohlhase, 2006c], [Libbrecht and Gross, 2006], or [Maus *et al.*, 2005, p. 53].

At first sight the strength of KM consists in offering the data of their underlying databases to its customers, which we comprise under "effectiveness", whereas EL's strength builds on enabling users to learn what they need to learn, which we subsume with "customization". Merging the two, we theoretically obtain adaptable systems that are effective as well as customizable. Even though this evolving synergy from an intertwinement of KM and EL is quite intuitive, the technological infrastructures are (historically) incompatible: KM developed from the field of Information Technologies and heads for a *"corporate storybook"* [Dunn and Iliff, 2005, p. 3], whereas EL grew out of Human Resources and aims at a *"class-in-a-box model"* [ibid.]. Moreover, a fusion of two fields typically doesn't involve their strengths alone, their weaknesses have to be addressed as well. So, what are these weaknesses?

The essential *bottleneck for Knowledge Management* consists in its **"Authoring Problem"**: the content for databases is not as voluntarily generated as one might have hoped. It is a follow-up problem of the more general *"Knowledge Acquisition Problem"* in the field of Artificial Intelligence, that appeared in the early eighties with its heat on expert systems. Here, so-called knowledge engineers were to extract knowledge from human experts and feed it into a database, which then represented a knowledge pool from which (with fitting algorithms) just the right expertise at just the right time could be delivered automatically. Essentially it turned out, that *people* didn't know how (or weren't keen) to formalize knowledge down for machines or they didn't want to share their expertise 'publicly'. With the still-growing acceptance of the World Wide Web, especially its participative aspect, the latter hurdle seems to be lowered considerably. For the former, KM took up the topic of knowledge representation and in the mean time has provided many authoring tools. But the problem remained: who is actually using them? There are still strong difficulties in motivating real people (not just first adopters, e.g. [Moggridge, 2007]) to share and explicate their knowledge.

The learning paradigm of Constructivism lies at the heart of (most) *E-Learning systems' weaknesses*: there are many

[1] In [Kornwachs, 2005] KORNWACHS critically discusses the use of the terms 'knowledge' versus 'information' and points to their "*fundamental difference*"[p. 34]. In particular, he points to the "*self-referential characteristics*"[p. 36] of knowledge that makes its handling via technological systems problematic. Keeping this (as well as [Probst *et al.*, 1997, p. 16], [Liessmann, 2006, 27ff.], and [Brown and Duguid, 2000, p. 125]) in mind, we use the term "knowledge" nevertheless.

things to learn in E-Learning applications, but how are they learned by real people?[2] Constructivism as learning theory states that the learning process is steered by the learner herself by adaptation and accommodation processes [Piaget, 1996]. That is, how does EL software present the content to its user steering her to a prefixed learning goal at the same time as encouraging self-steered learning processes with the same goal in mind? We speak of EL's **"Control Problem"**. The implementation of the constructivistic approach in E-Learning systems is rather antagonistic, as it focuses on a learner's guidance via didactical steering methods and underlying (already constructed) ontologies for learning objects, sometimes enhanced by simple user modeling techniques that principally are not adequate for a user's individual adaptation frame.

2 Semantic Interaction Design

Principally, we start out with the assumption that software is actively appropriated by the user (e.g. [Sesink, 2004; Schelhowe, 2007; Lunenfeld, 1999]). That means that data conveyed via a computer can be considered as mere semiotic signs, they only become meaningful when being interpreted by human beings. This activity allows to fit software into real life by interpreting their meaning and thereby relevance for the 'here-and-now'. Therefore, we are interested in the **micro-perspectives of users**, i.e. perspectives that evolve within concrete situations that are evaluated individually by each user. This view from within allows to understand the *rationality of taking action* when using a software product.

Although "having the user in mind" seems very natural, it really isn't. What feels natural about it, is that generally every software designer has the good of the end user in mind as otherwise there is no (acceptable) reason for developing such programs. What is not natural, is that software designers are not trained in understanding other people's life context (even though some are trained with respect to analyzing work contexts). Thus, the micro-perspectives view goes beyond mere 'user-centred design' which most software products claim for themselves nowadays.

In order to understand the (semantic) relationship between user and software, we contrast the micro-perspectives discussed above with the **macro-perspective**, i.e. a global view from without. We argue that software is typically designed from the macro-perspective, whereas the "use of software"-action is decided from the micro-perspective of each potential user — explaining unforeseen roadblocks for using software. By investigating the differences between macro- and micro-perspectives in more detail we will be enabled to understand the conditions behind specific situations. From a macro-perspective the benefits of using an application might seem to be very convincing, from a micro-perspective, there is often also motivation against taking action: The personal costs might be just too high. The essence of an occuring problem frequently lies in its assumptions — knowing these explicitly provides helpful keys for the problem's resolution.

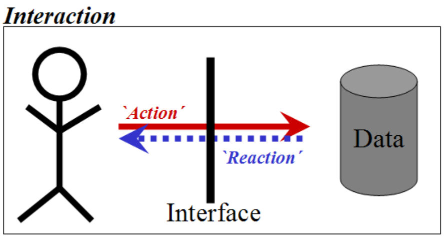


Figure 1: Designing Interaction is More than Designing Interfaces

These perspectives are especially interesting if we don't apply them to the interface design, but to the underlying interaction design as this represents the setting for the relationship between user and data: *"Designing interaction rather than interfaces means that our goal is to control the quality of the interaction between user and computer: user interfaces are the means, not the end"* [Beaudouin-Lafon, 2004, p. 4]. We depicted the situation between user and data in Figure 1. The (red) full arrow represents not only the input action, but also the user's expectations and attitude towards the system in the interaction. Likewise the (blue) dotted arrow marks not only software's (re)action but also the approach towards the user inscribed in the interaction design. This way, for instance a concept like "autonomy" can be represented even though it is not reified in the interface. For ease of terminology use[3], we divide the interaction process into an '**action**' part where the user is in focus, depicted with the full (red) arrow, and a '***reaction***' part, in which the stress is on the data/software, depicted by the dotted (blue) arrow. Note that interaction design rather connects user and software, whereas interface design tentatively separates them as the involved subjects and objects are not considered holistically.

To understand KM and EL systems we will look into their interaction design with a focus on the way users attribute meaning to the inherent actions and reactions. We will speak of the **Semantic Interaction Design** of a system. Note that this must take the user's situatedness into account as a critical component, as interaction relies on the human's capability of interpreting data so that they not only become meaningful but even carry agency (by their in-

[2] We believe that Constructivism has been taken up so broadly as learning paradigm because of its more modern "Menschenbild", i.e. idea of man, (for an overview, see [Reinmann, 2005, 146ff] or `http://beat.doebe.li/bibliothek/f00048.html`, last seen at 2007/08/24).

- Up to the middle of the 20th century *Behaviorism* [Skinner, 1999; Pavlov, 2007; Bandura, 1976] with its knowledge *transfer model* was the leading paradigm. Here, the idea of man is coined by its stimulus-response model which induces rather over-directed, inautonomous human beings.
- It was replaced by *Cognitivism* [Tolman, 1932] with its knowledge *tutoring model*. Here, in a nutshell, human brains are thought of as computers without acknowledging essential aspects of learning like motivation and emotion.
- In the 1980s this technocratic, engineering approach started to be superseded by the more empowering *Constructivism* [Piaget, 1996; Maturana and Varela, 1992] and the according knowledge *coaching model*. Here, people are considered as creators of their own reality, which indeed fits much better to the current understanding of men, e.g. think of the "N-Gen" [Tapscott, 1997] or "Digital Natives" [Prensky, 2001]. Seymour Papert introduced a variant called *Constructionism* [Papert and Harel, 1991], which stresses the embodied aspects of learning.

[3] Interaction is a rather complex term, in which we do not want to get entangled here. For a full account of interaction we refer to e.g. [Schelhowe, 2006], for 'Interaction Design' to [Löwgren and Stolterman, 2004]

terpreted underlying semantics). Semantic interaction gets enabled on the conceptual level: how do user and software deal with each other. In anthropomorphic terminology, we can speak of "computer and user as partners" [Kohlhase, 2006a] or interaction as an ongoing "conversation" [Crawford, 2003, p. 5].

The interaction framework can be evaluated from either micro- or macro-perspective, so that observed strengths and weaknesses sometimes turn out to be contrary.

An explicit goal of Semantic Interaction Design is the alignment of micro- and macro-perspectives, where the Semantic Interaction Design *process* is characerized by the following properties:

- The micro-perspective shows a benefit of the system when approached by a user (i.e. a positive full (red) arrow, so that the user is motivated to take the action of using it).
- The macro-perspective is in favor of the delivering part of the interaction (i.e. a positive dotted (blue) arrow, so that the design lives up to the expectations of the user).

We will now discuss these semantic perspectives concerning EL's Control Problem as well as KM's Authoring problem in regard to a combination of Knowledge Management and E-Learning.

3 Semantic Perspectives in KM versus EL

In this paper, having both these fundamental problems in mind, we want to elaborate on authoring of and dealing with knowledge blocks in KM and EL using the macro- and micro-standpoints as analytical method. That is, we need to address the question how combining KM and EL effects or may effect the creation and use of formalized content.

3.1 Use of Content

PATRICK DUNN and MARK ILIFF comprise the underlying strains of both fields as follows:

> *"The big idea of knowledge management is to use technology to make the knowledge contained within the business available to all employees, when they need it. The big idea of e-learning is to use technology to put training and coaching at the disposal of employees in such a way that they can learn what they need, when they need it."* [Dunn and Iliff, 2005, p. 5]

This description of KM and EL is obviously given from a macro-perspective and not from a single user's standpoint. From this macro-perspective, KM techniques mainly try to capture the available knowledge to be able to make effective use of it (but distribute it rather as an afterthought). On the other hand, EL technology aims at delivering the content just-in-time, i.e. at the exact moment when it is needed, thereby assuming an abundance of content.

In Figure 2 the relationship between the different systems and the underlying learning objects is tentatively demonstrated. In particular, from the macro-view KM offers lots of content to a user, i.e. from this standpoint we can mark it (the blue (dotted) arrow) as KM's strength. But how it can be made use of is of lesser concern to KM, i.e. from the macro-perspective KM's weakness consists in stopping short of the goal. In contrast, here E-Learning systems want to win the user with the learning opportunities offered and take much less care in the learning process itself. That is, the full (red) arrow in Figure 2 can be labelled with EL's strength, whereas the dotted (blue) arrow has to be referred to as its weakness.

It is rather striking, that the richness of the LO database is on the one side considered as means (EL) and on the other as ends (KM). Therefore, we can interpret the interaction (depicted as arrows) as a question of retrieval quality, where high precision is taken care of by EL and high recall by KM. In other words, we argue that *from the macro-perspective*,

- *KM wants to achieve high recall* rates (so that 'all' knowledge is made use of), but that
- *EL strives for high precision* rates (so that the 'right' knowledge is made use of).

Now let us look at those big ideas from the micro-perspective. Again we turn to PATRICK DUNN and MARK ILIFF first:

> *The current implementation of knowledge management is, in essence, databases. [...] Current e-learning [...] is useful for basic-level training, compliance and information delivery.* [Dunn and Iliff, 2005, p. 5]

The abundance of available content in KM systems attracts users, even though not too many services are offered. Working examples are content management systems like "WebCT" (Blackboard) or social software like "Wikipedia". The success of such systems shows that KM systems have veritable strengths on the input aspect of their interaction design, which counteracts its weakness on the reaction aspect (offering services).

The strength of an EL system is it's ability to provide user-tailored presentations of learning objects in a given learning situation, in other words in the reaction aspect of the interaction design. But the price for this is that the user may feel alienated that she is represented in the software — if at all — via a rather simplistic, static user model (e.g. in the ACTIVEMATH system [Melis *et al.*, 2001]). That is, even though she might think of the respective EL system as an enabling technology that broadens her action-radius or heightens her competence level *eventually*, it doesn't take care of the here-and-now of her concrete context. Note the stark contrast to the macro-perspective — the perspective of the institution offering the E-Learning education here.

As a surprising consequence the requirements for retrieval from the micro-view are complementary to the ones from the macro-view. In particular, *from the micro-perspective*,

- *KM wants to achieve high precision* rates (so that a user gets what she needs at that point in time), whereas
- *EL strives for high recall* rates (so that a user can find the 'best' learning object suiting her personal needs).

Note that customization presumes potential high recall, whereas effectiveness is based on high precision. In particular (cf. Figure 2), EL's Control Problem can be interpreted as a call for high precision from the macro-perspective and a call for high recall from the micro-perspective when retrieving content.

Semantic KM addresses EL's Control Problem

The main thrust in EL has been developing techniques or models to answer the high precision requirement. But, we argue, that the micro-perspective has to take precedence as it is decisive for taking the action of using the respective system (see [Kohlhase, 2005]). The Control Problem between user and system is a natural *consequence* of favoring the micro-perspective (high recall) over the macro-perspective (high precision) in EL. Maybe we 'just' have to vary the assumptions: EL interaction design has to take the

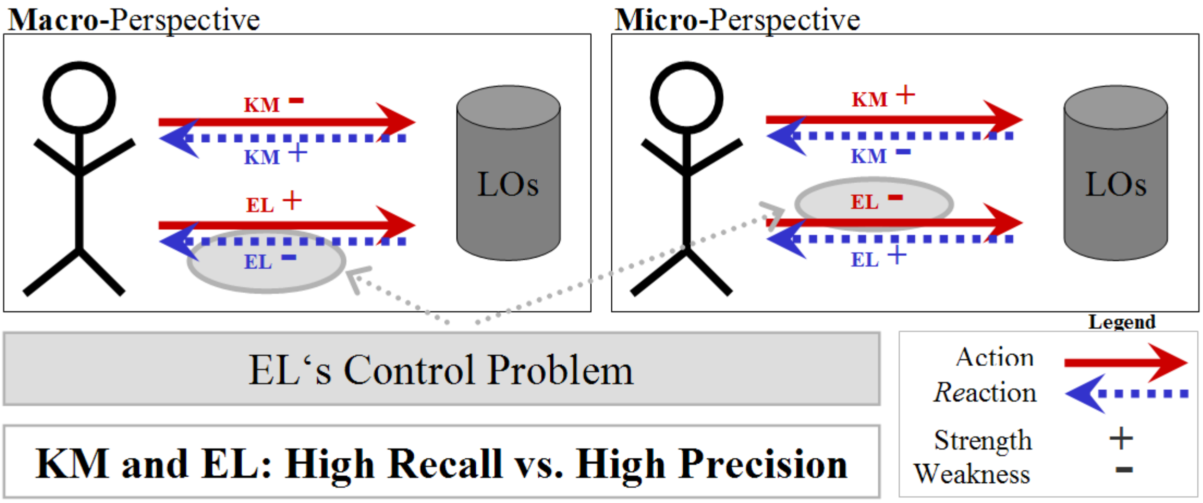


Figure 2: Distinct Perspectives on KM and EL wrt. *Use* of Content

user's demand for high recall into account. This can be delivered by *semantic KM technology*, i.e. a combination of KM- and semantic technology, building on explicitly represented knowledge objects (rather than EL text with implicit knowledge). In particular, if KM databases can offer more intelligent content, then all this content can be offered to the EL user enabling her not to choose from much too abundant content resources, but make *informed* decisions about her self-steered learning path. The constructivistic approach in KM Systems consists in managing microcontent, that can be aggregated, i.e. constructed, in various ways after its production. This way, we can make use of KM's strength to overcome EL's Control Problem.

3.2 Creation of Content

There is a common understanding in the Knowledge Management discipline that not all fabricated content can be supplied by outsiders, in the contrary that most has to be created within the context of evolvement, i.e. by collaboration. We list just a few of the arguments in the following. For one, as financial resources are evidently limited. For two (especially in organizations), since knowledge is far too dynamic and progresses fast. For three, because knowledge has a strong context component (see e.g. [Lave and Wenger, 1991; Brown and Duguid, 2000; 1991]). We can also spot a general tendency in the field of E-Learning to incorporate more and more collaborative features into their systems. Here, the reasoning is based on the social component of learning (based on e.g. [Vygotsky, 1978; Dewey, 1933; Lave and Wenger, 1991]). Together with the general trend from private data to public data on the Web, showcased by the considerable success of Social Software like DEL.ICIO.US, FLICKR, or SECONDLIFE in terms of user rates and described intriguingly in [Weinberger, 2002] and [Dourish, 2003], we restrict our analysis to *collaborative creation of content*.

In [Kohlhase and Kohlhase, 2004] we showed that the benefits of formalizing content for KM lie principally with its "readers", while the sacrifices remain with its "authors" — creating what we call the *"Authoring Dilemma"*[4] as we based our argument on the well-known "Prisoner's Dilemma" [Axelrod, 1984], which is often used for analyzing short term decision-making processes in cooperation scenarios, where the actors do not have any specific expectations about future interactions or collaborations. Moreover, in [Kohlhase, 2006b] the Authoring Dilemma was traced back to differing perspectives on the problem: the micro-perspective and the macro-perspective, where the first one is disabling content collaboration.

In particular, the problem was formulated in terms of a value 'landscape' where the action of creating can be optimized towards distinct optima. That is, here we can think about the according creation tasks in terms of optimizing action *from the macro-perspective* as follows:

- *KM wants to achieve the global optimum* (so that all available content is captured), but that
- *EL strives for the local optimum* (so that a user can progress on her chosen way).

In contrast, E-Learning software simply assumes the existence of qualitatively high learning resources and focuses on setting it up in the right context. That is, EL technology offers itself to the user notwithstanding the quality of learning objects. Again rather surprisingly, the resulting action optimizations from the micro-view are complementary to the ones from the macro-view. In particular, *from the micro-perspective*,

- *KM wants to achieve the local optimum* (so that a user gets done what needs to get done), whereas
- *EL strives for the global optimum* (comprised in a "Lifelong Learning" goal or aiming at lifting each individual's educational level).

Note that customization tends to go for the local optimum, whereas effectiveness is aimed at the global optimum.

Semantic EL addressing KM's Authoring Problem

Figure 3 visualizes the strengths and weaknesses of KM and EL with respect to the creation of content. We have argued elsewhere that the macro-perspective on KM technologies stresses its potential, but the actual offerings are rather simple and consist in authoring tools. From the micro-perspective however, the lack of supportive services prevents users from using the available authoring tools. Therefore, the Authoring Problem is a *consequence* of neglecting the micro-perspective and the failure of supplying

[4] In the field of Mathematical Knowledge Management (MKM) this is also referred to as "MKM's chicken-and-egg problem".

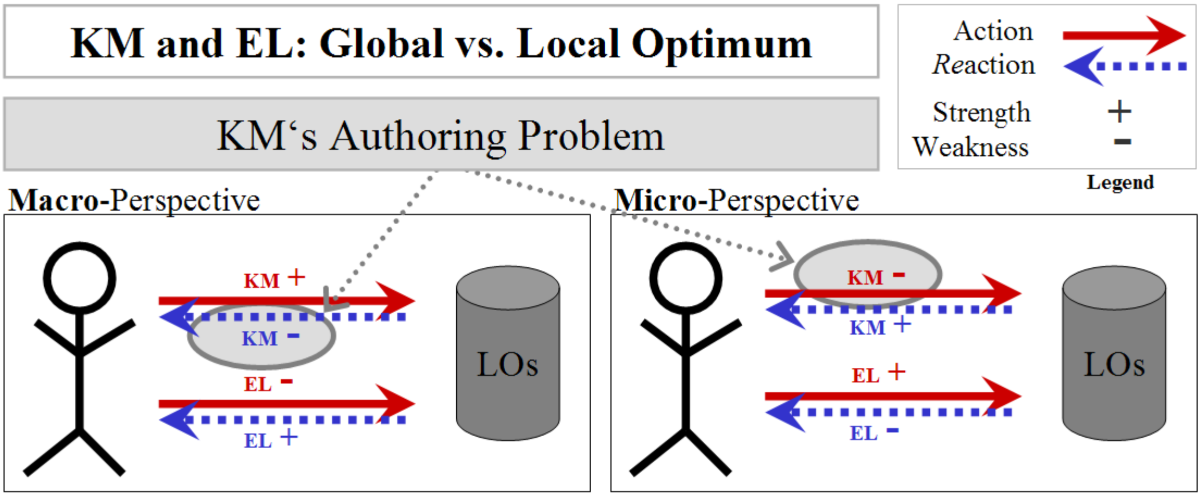


Figure 3: Distinct Perspectives on KM and EL wrt. *Creation* of Content

Added-Value Services [Kohlhase and Müller, 2007] based on the content. Even though the fashionable paradigm of the "user as producer and consumer" is appealing from a macro-perspective — from a micro-perspective, the hurdle to become a producer is a serious one and has to be acknowledged in software-design.

How can this be done? We suggest to use a combination of El and semantic technology, which we call *semantic EL technologies*. In particular, if EL data consisted of more intelligent content, then this could be tailored to the KM author enabling her to appreciate direct, intelligent services for her here-and-now situation. EL systems use the provided metadata of microcontent to construct a use context for the user. This approach makes use of EL's strength to alleviate KM's Authoring Problem.

4 Conclusion

In order to understand the benefits of a fusion of Knowledge Management and E-Learning technologies, we have focused on KM's Authoring Problem and EL's Control Problem. For this, we have looked at the strengths and weaknesses of their semantic interaction designs with respect to using and creating content within them. Here, we noted that both problems are characterized by weak user approaches from the micro-perspective and weak software deliveries from the macro-perspective. At the same time, KM and EL turned out to be rather complementary with respect to these actions under the distinct views. Therefore, we expect that merging the technologies will alleviate the respective problems. Moreover, we suggest that KM as well as EL technologies that are enriched by semantic technologies will enhance the potential resolution process.

References

[Althoff *et al.*, 2005] Klaus-Dieter Althoff, Andreas Dengel, Ralph Bergmann, Markus Nick, and Thomas Roth-Berghofer, editors. *Professional Knowledge Management*, number 3782 in LNCS. Springer Verlag, 2005.

[Axelrod, 1984] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[Bandura, 1976] Albert Bandura. *Lernen am Modell. Ansätze zu einer sozial-kognitiven Lerntheorie*. Klett-Cotta, 1976.

[Beaudouin-Lafon, 2004] Michel Beaudouin-Lafon. Designing interaction, not interfaces. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 15–22, New York, NY, USA, 2004. ACM Press.

[Brown and Duguid, 1991] John Seely Brown and Paul Duguid. Organizational Learning and Communities of Practice:Toward a Unified View of working, Learning and Innovation. *Organization Science*, 2(1):40–57, 1991.

[Brown and Duguid, 2000] John Seely Brown and Paul Duguid. *The Social Life of Information*. Harvard Business School Press, 2000.

[Crawford, 2003] Chris Crawford. *The Art of Interactive Design: A Euphonious and Illuminating Guide to Building Successful Software*. No Starch Press, 2003.

[Dewey, 1933] John Dewey. *Experience and Education*. New York: Macmillan, 1933.

[Dourish, 2003] Paul Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. MIT Press, 2003.

[Dunn and Iliff, 2005] Patrick Dunn and Mark Iliff. At cross purposes: Why e-learning and knowledge management dont get along. Online at http://www.learninglight.eu, viewed at 2007/07/04, 2005. Learning Light.

[Kohlhase and Kohlhase, 2004] Andrea Kohlhase and Michael Kohlhase. CPoint: Dissolving the Author's Dilemma. In Andrea Asperti, Grzegorz Bancerek, and Andrzej Trybulec, editors, *Mathematical Knowledge Management, MKM'04*, number 3119 in LNAI, pages 175–189. Springer Verlag, 2004.

[Kohlhase and Müller, 2007] Andrea Kohlhase and Normen Müller. Added-Value: Getting People into Semantic Work Environments. Accepted for Publication in "J. Rech, B. Decker, and E. Ras (editors): *Semantic Work Environments*", 2007.

[Kohlhase, 2005] Andrea Kohlhase. Overcoming Proprietary Hurdles: CPoint as Invasive Editor. In Fred de Vries and Graham Attwell and Raymond Elferink and Alexandra Tödt, editor, *Open Source for Education in Europe: Research and Practise*, pages 51–56. Open Universiteit of the Netherlands, Heerlen, 2005.

[Kohlhase, 2006a] Andrea Kohlhase. Media or Medea Society? Learner and Learning Technology as Full Partners. In Bundit Thipakorn, editor, *ICDML2006*, volume 1, pages 6–12. Tana Press, March 2006. Bangkok (Thailand), 2006-03-13/14.

[Kohlhase, 2006b] Andrea Kohlhase. The User as Prisoner: How the Dilemma Might Dissolve. In Martin Memmel, Eric Ras, and Stephan Weibelzahl, editors, *2nd Workshop on Learner Oriented Knowledge Management & KM Oriented e-Learning*, number 2, pages 26–31, 2006. Online Proceedings at `http://cnm.open.ac.uk/projects/ectel06/pdfs/ECTEL06WS68d.pdf`.

[Kohlhase, 2006c] Michael Kohlhase. *OMDoc: An Open Markup Format for Mathematical Documents*. Springer Verlag, August 2006. ISBN 3-540-37897-9.

[Kornwachs, 2005] Klaus Kornwachs. Knowledge + Skills + "x". In Althoff et al. [2005].

[Lave and Wenger, 1991] Jean Lave and Etienne Wenger. *Situated Learning: Legitimate Peripheral Participation(Learning in Doing: Social, Cognitive and Computational Perspectives S.)*. Cambridge University Press, 1991.

[Libbrecht and Gross, 2006] Paul Libbrecht and Christian Gross. Authoring LeActiveMath Calculus Content. In William Farmer and Jon Borwein, editors, *Mathematical Knowledge Management, MKM'06*, number 4108 in LNAI, pages 251 – 265. Springer Verlag, 2006.

[Liessmann, 2006] Konrad P. Liessmann. *Theorie der Unbildung. Die Irrtmer der Wissensgesellschaft*. Zsolnay, 2006.

[Löwgren and Stolterman, 2004] Jonas Löwgren and Erik Stolterman. *Thoughtful Interaction Design: A Design Perspective on Information Technology*. The MIT Press, 2004.

[Lunenfeld, 1999] Peter Lunenfeld, editor. *The Digital Dialectic: New Essays on New Media*. The MIT Press, 1999.

[Maturana and Varela, 1992] Humberto R. Maturana and Francisco J. Varela. *Tree of Knowledge: Biological Roots of Human Understanding*. Shambhala Publications Inc.,U.S., 1992. Originally published in 1984.

[Maus *et al.*, 2005] Heiko Maus, Harald Holz, Ansgar Bernardi, and Oleg Rostantin. Leveraging Passive Paper Piles to Active Objects in Personal Knowledge Spaces. In Althoff et al. [2005], pages 50–59.

[Melis *et al.*, 2001] E. Melis, E. Andres, A. Franke, G. Goguadse, P. Libbrecht, M. Pollet, and C. Ullrich. LE ACTIVEMATH system description. In Johanna D. Moore, Carol Luckhard Redfield, and W. Lewis Johnson, editors, *Artificial Intelligence in Education*, volume 68 of *Frontiers in Artificial Intelligence and Applications*, pages 580–582. IOS Press, 2001.

[Moggridge, 2007] Bill Moggridge. *Designing Interactions*. MIT, 2007.

[Papert and Harel, 1991] S. Papert and I. Harel. Situating Constructionism. In S. Papert and I. Harel, editors, *Constructionism*. Ablex Publishing, 1991.

[Pavlov, 2007] Ivan P. Pavlov. *Die Arbeit der Verdauungsdrüsen*. Saarbrücken : VDM, Müller, 2007. Originally published in 1898.

[Piaget, 1996] Jean Piaget. *Einfhrung in die genetische Erkenntnistheorie*. suhrkamp, 1996. First edition in 1974.

[Prensky, 2001] Marc Prensky. Digital Natives, Digital Immigrants. In *On the Horizon*, volume 9. NCB University Press,, Octobre 2001.

[Probst *et al.*, 1997] G. Probst, St. Raub, and Kai Romhardt. *Wissen managen*. Gabler Verlag, 4 (2003) edition, 1997.

[Reinmann, 2005] Gabi Reinmann. *Blended Learning in der Lehrerbildung*. Pabst, 2005.

[Schelhowe, 2006] Heidi Schelhowe. Interaktion und Interaktivität: Aufforderungen an die ITG und an die Medienpädagogik. In Werner Sesink and al, editors, *Jahrbuch Medienpädagogik 2006*. medienpaed, Juni 2006.

[Schelhowe, 2007] Heidi Schelhowe. *Technologie, Imagination und Lernen: Grundlagen für Bildungsprozesse mit Digitalen Medien*. Waxmann, 2007.

[Sesink, 2004] Werner Sesink. *In-formation: Die Einbildung des Computers*. Number 3 in Bildung und Technik. LIT Verlag Münster, 2004.

[Skinner, 1999] B.F. Skinner. *The Behavior of Organisms: An Experimental Analysis*. B.F. Skinner Foundation, 1999. Originally published in 1938.

[Tapscott, 1997] Don Tapscott. *Growing Up Digital: The Rise of the Net Generation*. McGraw-Hill Professional, 1997.

[Tolman, 1932] Edward Chace Tolman. *Purposive Behavior in Animals and Men*. New York: Century, 1932.

[Vygotsky, 1978] L.S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1978.

[Weinberger, 2002] David Weinberger. *Small Pieces Loosely Joined: a Unified Theory of the Web*. Basic Books, 2002.

5 Zitate

- The interest of the field of Mathematical Knowledge Management is predicated on the assumption that by investing into markup or formalization of mathematical knowledge, we can reap benefits in managing (creating, classifying, reusing, verifying, and finding) mathematical theories, statements, and objects. This global value proposition has been used to motivate the pursuit of technologies that can add machine support to these knowledge management tasks. But this (rather naive) technology-centered motivation takes a view merely from the global (macro) perspective, and almost totally disregards the user's point of view and motivations for using it, the local (micro) perspective.

  In this respect, the MKM approach is similar to the much-hyped *Semantic Web*, and suffers from the same problem: before the inference-based techniques of the field can pay off, a large volume of data (mathematical documents for MKM and web content for the Semantic Web) must be semantically annotated. Both fields also agree on how this should be achieved: rather than waiting for artificial intelligence methods that can automate this, we rely on content authors or volunteers to supply the annotations. Here, we have a very important difference to the case of Formal Methods, where the connection of the sacrifices incurred in program verification are directly linked to the expected benefits, usually a radically reduced risk of liability or reduced insurance payments.

- In Section **??** we will argue that we can comprise this scheme to a "*user as a prisoner*"-concept (cf. the well-known "Prisoner's Dilemma"). The dilemma consists in two competing perspectives on taking action: the micro- and the macro-perspective, where the first one is disabling content collaboration. In [Kohlhase and Kohlhase, 2004] KOHLHASE and the author discussed this phenomenon as "Authoring Problem", in an educational context in [Kohlhase, 2005] as "User Riddle": even though the advantages of using KM systems for content collaboration seemed tremendous, no action was taken by users to invest the additional energy and effort to produce such content. So, the real problem in the "user as consumer and producer"-concept is the *micro-perspective of motivation for action* and it is not clear, whether the one or/and the other is more helpful for this.
- The current implementation of knowledge management is, in essence, databases. [...] Current e-learning [...] is useful for basic-level training, compliance and information delivery. [Dunn and Iliff, 2005, 5]
- The idea of putting things into a precise sequence versus the idea of each step being self-determining, each step following from the last but not necessarily depending on the last. Static versus dynamic.
- If we adopt the underlying characteristics of the social tagging process, we might be able to overcome the dichotomy of the user as a consumer *or* producer, of knowledge as content *or* form, and of roles like teacher *or* learner.
- Content objects are essential links between Knowledge Management and E-Learning systems. Therefore content authoring and sharing is an important, interdisciplinary topic in the resp. fields. In this paper, we want to critically elaborate on the "user as producer and consumer"-concept for content production and consumption. We address the subject by using the notion of content collaboration as example for the "*Prisoner's Dilemma*", in which the sensible way out (from a macro-perspective) is sensibly not pursued by an individual (from a micro-perspective). We will use this micro-perspective of a *user as prisoner* to analyze what the recently very successful Social Tagging processes can teach us about the user taking action as a producer and/or consumer.
- The analysis presented in this paper will form the starting point for the development of a stepwise process of content generation (working title: "Stepwise Blended Learning and Knowing"). We plan to implement and evaluate this in the context of the CPOINT system (implemented by the author)[5], leveraging a central aspect of the social tagging process: the transition from Personal KM up to a social, but self-steered E-Learning System.
- Norm Friesen: Cognitivism has traditionally seen cognition, content and context as separate; and recent attempts to overcome this separation (e.g., situated cognition) have been criticized for being insufficient (e.g., Lave, 1988).
- (Reinmann, S. 148ff.):In der Erforschung des Lernens (und Lehrens) gibt es drei groe Theoriesysteme: den Behaviorismus, den Kognitivisums und den Konstruktivismus. Die gewhlte Reihenfolge spiegelt die Chronologie der Entstehung der drei Theoriesysteme wider. ... Vielmehr zeigen neue Theoriesystem jeweils die Schwchen der vorangegangenen auf und verweisen diese auf begrenzte Geltungsbereiche.

  Aus wissenschaftstheoretischer Sicht zeigt sich, dass die drei groen Theoriesysteme zum Lernen unterscheidbare Auffassungen daszu vertreten, ob und auf welche Weise eine Realitt existiert (Ontologie), ob und wie der Mensch Wissen ber diese Realitt erlangen kann (Epistemologie), welche Forschungsfragen in den Blick genommen und welche Methoden zu deren Beantwortung befrwortet werden (Methodologie) und in welchem Verhlntnis der Mensch zu seiner Umwelt steht (Anthropologie). –¿ Lernparadigmen!

  – Behaviorismus: Stimulus-Response Modell, Black-Box-Denken (da das mentale Modell nicht interessiert), behav. um kogn. Aspekte erweitert: Lernen am Modell (beobachten und imitieren –¿ Beobachtungslernen, Imitationslernen sind beide identisch, S. 151). Das Menschenbild im Behaviorismus ist stark geprgt von Konditionierung auf und durch uere Reize (S.152)

  – Kognitivismus: Wissensreprsentation sowie Regeln der menschlichen Informationsaufnahme, -speicherung und -wiedergabe. Psychische Prozesse in technischen Modelle abbildbar (KI Newell&Simon, Informationstheorie Shannon, Kybernetik Wiener). Das menschenbild im Kognitivismus ist weniger mechanistisch als im Behaviorismus, weil man dem Mnschen auch zielgerichtetes Handeln und nicht nur reaktives Verhalten unterstellt. Kennzeichnend ist aber auch hier die Suche nach mglichst berechenbaren Beziehunene und Regeln innerhalb und zwischen kognitiven Prozessen des Menschen. Problematisch a Kogn. ist (aus heutiger Sicht) die Computer-Metapher, die zu einem technologischen Verstndnis von Lernen verleitet und dabei die Komplexitt des Gehirns unterschtzt (S. 161). Alle Aspekte menschlichen Lernens werden auf Informationsverarbeitungsprozesse reduziert — auch Fragen der motivation und Emotion; das subjektive Erleben bleibt auen vor. Ermglichung von Lernen (Arnold, 2003), p.169. Engineering- vs. Empowerment-Prinzipien. Instruktionsdesign vs. Kontextdesign.

  – Konstruktivismus: Bedeutungskonstruktion, Eigenaktivitt und Selbstorganisation statt Reaktivitt und Auensteuerung *in einem nicht-technischen Sinne* (im Ggs. zu Kogn.), s. 155. Strukturelle Koppelung. Kognition als wirksames Handeln. Neuer Konstruktivismus:

[5] CPoint is an open source, semantic, invasive editor from within MS PowerPoint that attaches semantic annotation to PPT-objects and converts this micro-content into a web-capable format. More information and download site is available under `http://kwarc.eecs.iu-bremen.de/projects/CPoint/install.html`

Terhardt 1999; Reinmann-Rothmeier & Mandl 2001). Der Mensch gilt im Konstruktivismus konsequenterweise als Erschaffer seiner eigenen Realitt, als "Welterzeuger", der niht nur reagiert oder Informationen verarbeitet, sondern gestaltend in seine Welt eingreift.

– Transfer-, Tutor-, Coachmodell.

Towards Improving Interactive Mathematical Authoring by Ontology-driven Management of Change

Normen Müller[1], **Marc Wagner**[2]

[1]Jacobs University Bremen
D-28759, Bremen, Germany
n.mueller@jacobs-university.de

[2]Universität des Saarlandes
D-66123, Saarbrücken, Germany
wagner@ags.uni-sb.de

Abstract

The interactive use of mathematical assistance systems requires an intensive training in their input and command language. With the integration into scientific WYSIWYG text-editors the author can directly use the natural language and formula notation she is used to. In the new *document-centric* paradigm changes to the document are transformed by a mediator into commands for the mathematical assistance system. This paper describes how ontology-driven management of change can improve the process of interactive mathematical authoring.

1 Introduction

Mathematical proof assistance systems have not yet achieved recognition and relevance in mathematical practice. Significant progress is still required, in particular with respect to the user-friendliness of these systems. Rather than developing a new user interface for the mathematical assistance system ΩMEGA the PLATΩ system [plato, 2007] presents a generic way of integrating proof assistance systems into scientific text-editors by using a flexible and parametric semantic annotation language. PLATΩ allows for the incremental development of mathematical documents in professional type-setting quality by propagation of changes and context sensitive service menu interaction.

The aim of the PLATΩ system is to support the complete authoring process of a mathematical document - from creation through formalization to publication - in a collaborative environment. This paper investigates the added values for the authoring process when integrating the `locutor` system [locutor, 2007; Müller, 2006; 2007] into PLATΩ. Using ontology-driven management of change and hence maintaining semantic dependencies the concrete research questions are: How can we preview the effects of a modification for the author? How can authors be informed about dependency conflicts during collaborative editing? How can we provide suggestions for conflict resolution?

2 The Mediator: PLATΩ

The development of the proof assistance system ΩMEGA is one of the major attempts to build an all-encompassing assistance tool for the working mathematician or for the formal work of a software engineer. It is a representative of systems in the paradigm of *proof planning* and combines interactive and automated proof construction for domains with rich and well-structured mathematical knowledge. The ΩMEGA-system is currently under redevelopment where, among others, it has been augmented by the development graph manager MAYA, and the underlying natural deduction calculus has been replaced with the CORE-calculus [Autexier, 2005].

The MAYA system [Autexier and Hutter, 2005] supports an evolutionary formal development by allowing users to specify and verify developments in a structured manner, it incorporates a uniform mechanism for verification in-the-large to exploit the structure of the specification, and it maintains the verification work already done when changing the specification. Proof assistance systems like ΩMEGA rely on mathematical knowledge formalized in structured theories of definitions, axioms and theorems. The MAYA system is the central component in the new ΩMEGA system that takes care about the management of change of these theories via its OMDOC-interface [Kohlhase, 2006].

The CORE-calculus supports proof development directly at the *assertion level* [Huang, 1996], where proof steps are justified in terms of applications of definitions, axioms, theorems or hypotheses (collectively called *assertions*). It provides the logical basis for the so-called TASK LAYER [Dietrich, 2006], that is the central component for computer-based proof construction in ΩMEGA. The proof construction steps are: (1) the introduction of a proof sketch [Wiedijk, 2004], (2) deep structural rules for weakening and decomposition of subformulas, (3) the application of a lemma that can be postulated on the fly, (4) the substitution of meta-variables, and (5) the application of an inference. Inferences are the basic reasoning steps of the TASK LAYER, and comprise assertion applications, proof planning methods or calls to external theorem provers or computer algebra systems (see [Dietrich, 2006; Autexier and Dietrich, 2006] for more details about the TASK LAYER).

A formal proof requires to break down abstract proof steps to the CORE calculus level by replacing each abstract step by a sequence of calculus steps. This has usually the effect that a formal proof consists of many more steps than a corresponding informal proof of the same conjecture. Consequently, if we manually construct a formal proof many interaction steps are typically necessary. Formal proof sketches [Wiedijk, 2004] in contrast allow the user to perform high-level reasoning steps without having to justify them immediately. The underlying idea is that the user writes down only the interesting parts of the proof and that the gaps between these steps are filled in later, ideally fully automatically (see also [Siekmann *et al.*, 2002]). Proof sketches are thus a highly adequate means to realize the tight integration of a proof assistance system and a scientific text-editor.

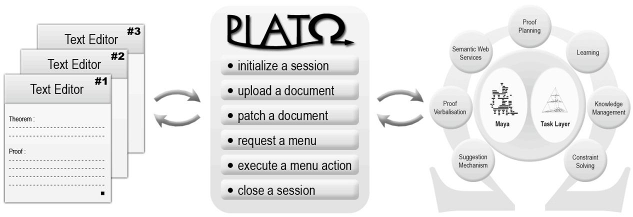


Figure 1: Architecture of the integration of the text-editor TeXMACS and the ΩMEGA system via the mediator PLATΩ

The mediator PLATΩ [Wagner *et al.*, 2006] has been designed as a support system to realize the tight integration of a proof assistance system and a text-editor (see Figure 1). PLATΩ is connected with the text-editor by an informal representation language which flexibly supports the usual textual structure of mathematical documents. This semantic annotation language, called *proof language* (PL), allows for underspecification as well as alternative (sub)proof attempts. In order to generate the formal counterpart of a PL representation, PLATΩ separates theory knowledge like definitions, axioms and theorems from proofs. The theories are formalized in the *development graph language* (DL), which is close to the OMDOC theory language supported by the MAYA system, whereas the proofs are transformed into the *tasklayer language* (TL) which are descriptions of TASK LAYER proofs. Hence, PLATΩ is connected with the proof assistance system ΩMEGA by a formal representation close to its internal data structure.

Besides the transformation of complete documents, it is essential to be able to propagate arbitrary changes from an informal PL representation to the formal DL/TL one and the way back. If we always perform a global transformation, we would on the one hand rewrite the whole document in the text-editor which means to lose large parts of the natural language text written by the user. On the other hand we would reset the data structure of the proof assistance system to the abstract level of proof sketches. For example, any already developed expansion towards calculus level or any computation result from external systems would be lost. Therefore, one of the most important aspects of PLATΩ's architecture is the propagation of changes.

The formal representation finally allows the underlying proof assistance system to support the user in various ways. PLATΩ provides the possibility to interact through context-sensitive service menus. If the user selects an object in the document, PLATΩ requests service actions from the proof assistance system regarding the formal counterparts of the selected object. Hence, the mediator needs to maintain the mapping between objects in the informal language PL and the formal languages DL and TL.

In particular, the proof assistance system supports the user by suggesting possible inference applications for a particular proof situation. Since the computation of all possible inferences may take a long time, a multi-level menu with the possibility of lazy evaluation is provided. PLATΩ supports the execution of nested actions inside a service menu which may result in a patch description for this menu.

Furthermore, the PLATΩ system provides an efficient management of user-defined notation [Autexier *et al.*, 2007] that allows the author to define her own notation inside a document in a natural way, and use it to parse the formulas written by the author as well as to render the formulas generated by the proof assistance system.

3 The Document Engineer: `locutor`

The `locutor` system aims at the development of a methodology, techniques, and tools to support a management of change (MOC) for informal but internally structured documents, i.e. to support the evolution, revision and adaptation of collections of technical documents. The system adapts and extends change management techniques from formal methods (cf. development graphs) to the informal setting. Instead of a formal semantics we assume that these documents adopt syntactical and semantic structuring mechanisms formalized in a *document ontology* (cf. Figure 2). This is an ontology that formalizes the document structure rather than the document contents and is also used to classify the type of documents. In particular we assume that it provides a notion of document fragment equivalence (cf. section 5.1). This formalization provides a notion of consistency and invariants that allows one to propagate effects of local changes to entire documents. Conversely, the ontology will provide means to localize effects of changes by introducing a notion for semantic dependencies between document parts.

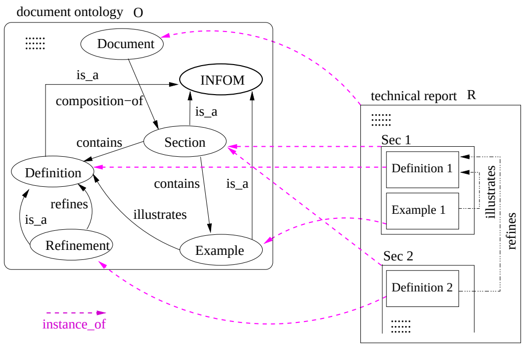


Figure 2: A document ontology $\mathcal{O}$

We regard *documents* as *self-contained structured compositions of information units*. One can pragmatically think of information units as "*tangible/visual text fragments potentially adequate for reuse*" constituting the content of documents. To distinguish the term "information unit" between common speech and the ontological concept, we call the ontological concept INFOM.

Following the OMDOC approach[1] we separate documents into two layers both under version control: A *narrative* and a *content* layer both of which consist of INFOMS and are composed via relations. The presentational order of information units in documents is represented on the narrative layer whereas the information units themselves and the ontological relations between them are placed in the content layer[2]. The connection between the narrative and the content layer is represented via *narrative relations* (analogous to symbolic links in UNIX). The information units and the ontological relations build up the "content commons" [CNX, 2007]. We use the term NARCON for the graph representations of document collections consisting of a narrative layer and a content layer.

[1] The OMDOC group does not claim to have invented this concept, it is part of the XML folklore and can already be found e.g. in [Verbert and Duval, 2004]. But the OMDOC format probably implements this idea in the cleanest way.

[2] Both the structural and the ontological relations are retrieved by the respective document ontology.

Following the initial work in the MMiSS [MMiSS, 2007] project, we also model the concept of *variants*. This expands the application area not only "in-the-breadth" but also "in-the-depth". Thus, by extending the well-known concept of *versions* and *revisions* by the concept of variants, the life-cycle of documents will no longer be only along a horizontal time line but also along a vertical line of variants. On the document level we call the concept of versions, revisions, and variants *document states*.

The computation of structural differences between two document states is based on the insights of XMLDIFF tools and the initial work of [Eberhardt and Kohlhase, 2004; Kohlhase and Anghelache, 2003]. According to this we extended the `diff`–algorithms and unification-based techniques, proposed there, to operate on NarCons resulting in a $\mathcal{M}$`diff`-algorithm, i.e. a model based `diff`–algorithm comprising an equality theory on NarCons (w.r.t. the respective document ontology). Therewith `locutor` is able to identify syntactically different InfoMs to be semantically equal and thus to minimize the number of InfoMs affected when changing InfoMs *(Equality Theory)* and to frame the syntactical representation of InfoMs and thus to help to locate changes of InfoMs relative to the internal structure *(Syntactical Structure)*.

In the first step, to compute the *long-range effect of changes* the `locutor` system prompts the author for a *classification* of the computed structural differences. Therefore the system provides a MoC ontology comprising a *taxonomy of change relations*. The idea is to capture the essence of semantically equal InfoMs in the specification of equivalence relations R on InfoMs. Then, dependencies between InfoMs are always relative to equivalence classes, i.e. changing an InfoM I within an equivalence class will not affect InfoMs that depend on I only with respect to R. The connection between a document ontology and MoC ontology is modeled in a so-called *system ontology*. The central intuition behind this approach is that *strong change management* (SCM) techniques can be based on information that can be expressed in system ontologies. We claim that the `locutor` system only needs the system ontology part of a fully formal domain semantics. Thus system ontologies will be the central means for extending the SCM methods to the structured, two-layered and two-dimensional document setting.

In the second step, to propagate changes, the `locutor` system performs a *reasoning on classified structural differences* utilizing inference rules consolidated in a *change relation calculus* based on the respective system ontology.

4 Integrating `locutor` into PlatΩ

We plan to consolidate the two systems as depicted in Figure 3. Therewith we want to gain besides collaborative authoring with version management the following benefits:

4.1 Benefits for the User

Besides version management and collaborative authoring the integration of `locutor` into the PlatΩ system should decrease conflicts and thus time-consuming recomputations.

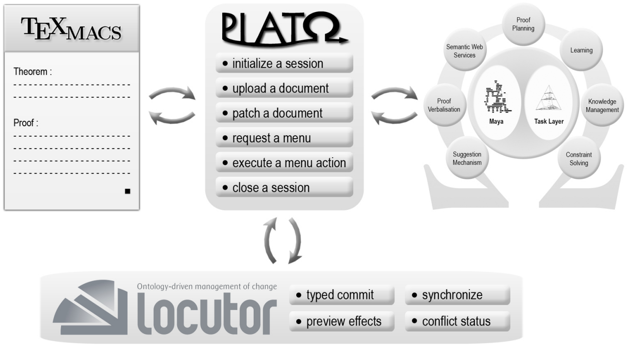


Figure 3: Integrated Architecture of `locutor` and PlatΩ

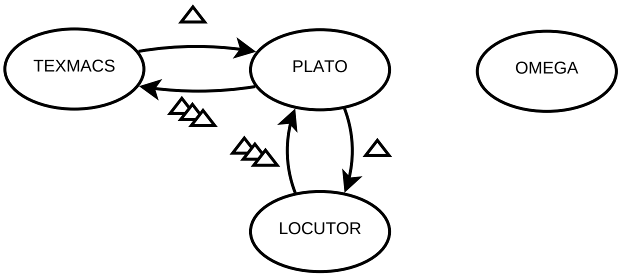


Figure 4: Preview of change effects by `locutor`

`locutor` should be able to preview the effects of a modification for the author and to improve consistency on the document level either by adaptation on demand or by automatic adaptation. Figure 4 shows a scenario where the author e.g. modified a variable name inside a formula. Let $\forall A, B.\ A = B \Leftrightarrow A \subset B \wedge B \subset A$ be the old formula and let $\forall \mathbf{C}, B.\ A = B \Leftrightarrow A \subset B \wedge B \subset A$ be the new one. This single modification is sent to `locutor` which in turn adapts all dependent variable positions in the same formula in order to preview the effects for the author. Thus the formula $\forall \mathbf{C}, B.\ \mathbf{C} = B \Leftrightarrow \mathbf{C} \subset B \wedge B \subset \mathbf{C}$ is shown as preview.

4.2 Benefits for the Proof Assistance System

Regarding PlatΩ's interface to the mathematical assistance system, `locutor` should act like a firewall blocking erroneous. Otherwise the mathematical assistance system would try to verify the erroroneous input and thus wasting the time of the author who waits for feedback. That is for example the scenario in Figure 5, where the author performs a set of modifications which produce conflicts in the document that cannot be resolved automatically. Then the author will be notified of the conflicts and may resolve or force them. As example we consider again the formula $\forall A, B.\ A = B \Leftrightarrow A \subset B \wedge B \subset A$. When the author modifies this formula to $\forall \mathbf{C}, B.\ \mathbf{D} = B \Leftrightarrow A \subset B \wedge B \subset A$ the automatic adaptation fails in the former occurrence of the variable A that has been renamed to D. Therefore the author will be asked whether or not this conflict is intended.

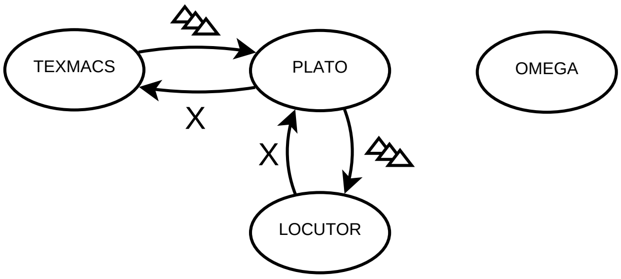


Figure 5: Notification of conflicts by `locutor`

Moreover, by identifying dependent modifications the `locutor` system should be able to return a combined meta change information. Considering the example with the old formula $\forall A, B.\ A = B \Leftrightarrow A \subset B \wedge B \subset A$ and the new formula $\forall \mathbf{C}, B.\ \mathbf{C} = B \Leftrightarrow \mathbf{C} \subset B \wedge B \subset \mathbf{C}$, `locutor` is able to identify the α-conversion of the variable A to C. Instead of propagating the renaming of each single occurrence of the variable A in that formula to the mathematical assistance system, this set of modifications is sent to `locutor` as shown in Figure 6 who combines them to one meta modification, the *replacement* $\{A \mapsto C\}$, which is finally applied in the mathematical assistance system directly on the whole formula. The mathematical assistance system then takes care about the more complex dependencies between the variable names in formulas occurring in different proofsteps.

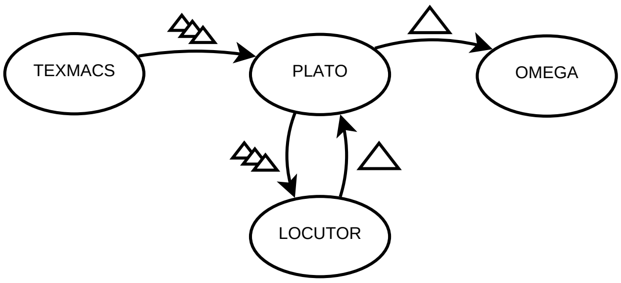


Figure 6: Combining a set of dependent changes to a meta change information

4.3 Ontology-driven Management of Change

First of all, the author uses the document ontology of the PLATΩ system, the *proof language* PL, to semantically annotate the document in the text-editor. This ontology formalizes the document structure and is used by PLATΩ for the communication with the proof assistance system and the efficient propagation of changes. Beside that, the semantics of that document ontology allows to define an ontology inside individual documents: A concept like the predicate $\subset$ can be introduced together with alternative notations using the following annotation format.

```
\begin{definition}{Predicate $\in$}
  The predicate \concept{\in}{elem \times
  set \rightarrow bool} takes an individual
  and a set and tells whether that
  individual belongs to this set.
\end{definition}

\begin{notation}{Predicate $\in$}
  Let \declare{x} be an individual and
  \declare{A} a set, then we write
  \denote{x \in A}, \denote{x is element
  of A} or \denote{A contains x}.
\end{notation}
```

Concepts are implicitely related by their type information. Furthermore, they can be grouped and ordered by precedence using PLATΩ's ontology. All introduced concepts can directly be used in the same document for writing formulas, for example in axioms, theorems and proofsteps. With the integration of `locutor` into PLATΩ we aim at a management of change for documents that is driven by an ontology which is dynamically defined by the documents themselves.

In the following we will discuss the equivalence and change relations of `locutor` and how they can be adapted to the needs of PLATΩ with illustrative examples.

5 Equivalence Relations

A stronger notion of equality leads to more compact, less intrusive edit scripts. For instance, if we know that ordering of elements carries no meaning in a document format (think of BibTeX entries), two documents are considered equal, even if they differ in every single line (w.r.t. to the element order). Consequently, with this notion of equality, the computed edit script (a representation of the document differences) would be empty. This motivates the need to identify syntactically different but semantically equal INFOMs and thus to generate less intrusive edit scripts.

5.1 Primitive Equivalence Relations

In order to support such an efficient and less interfering management of change, `locutor` provides *primitive* equivalence relations. These are abstract specifications of equivalence relations, which have to be implemented in the respective document ontology. `locutor` provides a sophisticated plug-in mechanism for them, such that each plug-in provides its own $\mathcal{M}$`diff`-algorithm. These primitives specify and utilize semantic aspects of documents (INFOMs), in particular, they are used to identify syntactically different INFOMs as semantically equal. Changes are only propagated if they change the semantics. Typical examples are parsers of programming languages which will ignore, for instance, the number of white spaces between lexical symbols. A classical way to introduce such a classification on INFOMs is the use of equivalence relations R on INFOMs. Two INFOMs are considered as "semantically" equal w.r.t. an equivalence relations R iff both are in the same equivalence class. Labelling a dependency between two INFOMs with R, a change of one INFOM would be only propagated if that would also change the equivalence class the INFOM belongs to. Currently identified primitives are:

Include Normalization (`INC`)

Many document formats provide some kind of literal inclusion mechanism. For instance, OMDOC provides the `ref` element, TEX provides the `\include` and `\input` macros, and XML provides the construct of parsed entities and XINCLUDE [W3C, 2007]. We call the process of replacing an `include` element by its target in a document *include-reduction*, and the document resulting from the process of systematically and recursively reducing all the `include` elements the *include-normal form* of the source document. As include-normalization may not always be possible, e.g. if the targets do not exist or are inaccessible, we call a document *include-reducible*, iff its `include`-normal form exists, and *include-valid*, iff the `include` normal form exists and is a valid document of the class. Arbitrary *include-valid* documents are considered to be semantically equal to the respective *include-normal form*.

White-Spacing (`WHI`)

In various document formats, multiple consecutive white spaces do not carry any further semantics, but are for readability only. For instance, think of LATEX users using double white spaces, or a double newline to mark off the beginning of a new paragraph. Regarding XML documents, however, the indent level and white space only nodes do matter (cf. `xml:space` attribute). Finally, there is the application/operating system dependent white space and newline character encoding: the carriage return (`\r` or ch(13)), the linefeed (`\n` or ch(10)), the tab (`\t`), and the spacebar (' '). We subsume all these issues under the term “white-spacing”. By this equivalence relation white-spacing within arbitrary documents is ignored, i.e. documents only differing in white-spacing are considered to be equal.

Ordering ($\mathtt{ORD}_{\mathtt{DocFor}}$)
The order of elements matters in almost all document formats (DocFor), for instance, the raw XML snippet <root><A/><B/></root> is not equal to <root><B/><A/></root>. Additionally the order restriction may change between different elements, i.e. for some elements the order matters but is irrelevant for others in the same document format. Thus this primitive equivalence relation is relative to the grammar of the respective document format, in particular, relative to single elements. For example, key-value pairs in OMDOC do have a strict order, but CMP elements can be freely ordered. We differentiate between *loosely* and *strictly* specified elements. For *loosely* specified elements the logical dependency graph is mandatory to the structural dependency graph. That is documents with the same logical dependency graph but a different structural dependency graph w.r.t. the order of the elements are considered to be equal. For *strictly* specified elements the structural dependency graph implied by the respective grammar is at the front.

URI-Normalization (URI)
This primitive equivalence relation causes locutor to consider relative path statements to be equal to normalized path statements, i.e. to the corresponding resolved absolute path. For example, within a document the relative path statement URI(../../lwa07.tex) might normalize to URI(https://www.kwar.info/nmueller/conferences-lwa07/lwa07.tex) and both paths are considered to be $\equiv_{\text{URI}}$ equivalent.

Formulae (FOR)
Content representations of mathematical formulae like OPENMATH or MATHML come with their own equivalences. We subsume α-conversion, dispensable variables and nested attribution under the term “formulae-equivalence”. *α-conversion* Regarding the OPENMATH specification [Buswell *et al.*, 2004] binding objects are constructed from an OPENMATH object, and from a sequence of zero or more variables followed by another OPENMATH object. The first OPENMATH object is the “binder” object. Arguments 2 to $n-1$ are always variables to be bound in the “body” which is the n^{th} argument object. We write $\beta(b, x_1, \ldots, x_m, O)$ where β denotes the OPENMATH binding operation, $x_1, \ldots, x_m$ the bound variables, and O the body. For example, activating this equivalence relation leads locutor to consider the following α-equality: $\beta(\lambda, z, y, x, z(yx)) \equiv_{\text{FOR}} \beta(\lambda, f, g, t, f(gt)) \equiv_{\text{FOR}} \beta(\lambda, x, y, z, x(yz))$.

Dispensable Variables In OPENMATH, repeated occurrences of the same variable in a binding operator are allowed. Thus a binding with multiple occurrences of the same variable is considered to be semantically equivalent to the binding in which all but the last occurrence of each variable is replaced by a new variable which does not occur free in the body of the binding. That is $\beta(\lambda, v, v, v \times v) \equiv_{\text{FOR}} \beta(\lambda, v', v, v \times v)$.

Nested Attribution An OPENMATH attribution decorates an object with a sequence of one or more pairs made up of an OPENMATH symbol, the “attribute”, and an associated object, the “value”. We write $\alpha(O, (k_i \mapsto v_i)_i)$ where α denotes the OPENMATH attribution operations, k_i are OPENMATH symbols, and v_i and O are OPENMATH objects. As the value can be an OPENMATH attribution object itself, compositions of attributions are allowed and are considered semantically equivalent to a single attribution. That is, $\alpha(\alpha(O, (k_i \mapsto v_i)_i), (k'_j \mapsto v'_j)_j) \equiv_{\text{FOR}} \alpha(O, (k_i \mapsto v_i, k'_j \mapsto v'_j)_{i,j})$.

Theory-Refactorization (THE)
Top-level OMDOC theories containing nested theories are semantically equal to the so-called *refactored* theories (modulo theory renaming). That is,

```
<theory xml:id=”A”>
  <symbol name=”a”/>
  <theory xml:id=”B”>
    <symbol name=”b”/>
  </theory>
</theory>
```

is equivalent to

```
<theory xml:id=”A’”>
  <symbol name=”a”/>
</theory>
<theory xml:id=”B’”>
  <imports from=”A’”/>
  <symbol name=”b”/>
</theory>
```

The top-level theory A containing a nested theory B is equivalent to the refactored, top-level theories A' (w/o B) and B' where B' now imports A'. Note, the refactorization has to be performed from the outermost to the innermost theory.

We are currently investigating further primitive equivalence relations, like in an XML document, attributes having default values and attributes being absent are considered equal. This is important because many applications fill in default values automatically.

5.2 Conjoint Equivalence Relations

Figure 7 in the first column summarizes the previously described primitive equivalence relations. By postulating $\forall \rho, \sigma \in Eq. \rho\sigma = \sigma\rho \Rightarrow \rho\sigma = (\rho \cup \sigma)^*$ where $\rho\sigma$ and $(\rho \cup \sigma)^*$ are again equivalence relations, these primitives may be composed to computable *conjoint* equivalence relations. Default implementations for the document formats XML, OPENMATH, and OMDOC are under construction. The predefined conjunctions $\equiv_{\text{XML}}$, $\equiv_{\text{OpenMath}}$ and $\equiv_{\text{OMDoc}}$ are contained in the respective document ontologies. A conjoint equivalence relations is interpreted as the transitive closure of the union of the implemented primitives, e.g. $\equiv_{\text{XML}} := (\mathtt{INC} \cup \mathtt{URI})^*$. Note, as for example $\equiv_{\text{XML}} \subseteq \equiv_{\text{OpenMath}} \subseteq \equiv_{\text{OMDoc}}$ holds, the plug-in specification of locutor also provides (and encourages) the reuse of implementations of primitives in “larger” conjunctions, i.e. inheritance between document ontologies, in

	$\equiv_{\text{XML}}$	$\equiv_{\text{OpenMath}}$	$\equiv_{\text{OMDoc}}$	$\equiv_{\text{CNXML}}$	$\equiv_{\text{TeX}}$
INC	√	√	√	√	√
WHI		√	√		√
$\mathtt{ORD}_{\mathtt{OMDoc}}$			√		
FOR		√	√		
URI	√	√	√	√	√
THE			√		

Figure 7: Equivalence relation matrix of locutor

particular regarding $\mathcal{M}$`diff`-algorithms. To demonstrate the flexibility of the emerging equivalence relation matrix, we appended the conjunctions $\equiv_{\text{CNXML}}$ and $\equiv_{\text{TeX}}$ to emphasize the potential support of further document formats, e.g. the support of CNXML [Hendricks and Galvan, 2007], XHTML [W3C, 2000], MATHML [W3C, 2003], or even TEX.

6 Change Relations

The following change relations serve as classifications for computed structural differences and as such constitute the input of the change relation calculus. Depending on the classified modifications and the type of the dependency between the elements, `locutor` first reasons on and then propagates the changes w.r.t. the dependency types specified in the system ontology. We propose to annotate each dependency relation by a set of primitive equivalence relations on which they are sensitive to. In addition we propose to annotate each change relation by a set of primitive equivalence relations which they are violating. Thus, if the intersection of two such sets is not empty the change has to be propagated. Subject to the level of resulting consistency the author either retrieves precise (accumulated) locations within the document to manually re-check or the document is automatically adapted to a consistent state (w.r.t the system ontology). In the later case `locutor` returns the meta-diff[3] information to the PLATΩ system which is then able to call a meta-command in the mathematical assistance system instead of calling multiple commands for each single modification. Currently identified change relations are:

6.1 Conservative(φ)

Given a primitive equivalence relation φ, the parametrized change relation *Conservative* denotes a φ-equivalence preserving change. That is, modifying an element A to A' such that $A \equiv_\varphi A'$ still holds, `locutor` generates a *meta-diff* Δ_φ comprising both automatically adapted elements and those elements whose relation to A is violated by this modification. For example, a formula $\forall A, B.\ A = B \Leftrightarrow A \subset B \wedge B \subset A$ referring to $\forall A, B.\ A \subset B \Leftrightarrow \forall x.x \in A \Rightarrow x \in B$ via an operator name is not affected by an α-conversion, like $\forall D, C.\ D \subset C \Leftrightarrow \forall y.y \in D \Rightarrow y \in C$. However, renaming $\subset$ to $\sqsubset$ would violate the "referencing-by-name" relation from $=$ to $\subset$. In this case `locutor` infers a refactorization (cf. section 6.2) and propagates the change along the reverse dependency from $\subset$ to $=$.

6.2 Refactored(ϱ)

Given a sub-type of a refactoring $\varrho \in \{$`Renamed`, `Moved`, `Inlined`, `Deleted`, `Replaced`$\}$, the parameterized change relation denotes a syntatic change of type refactorization with sub-type ϱ. In this case `locutor` fully automatically propagates the changes and adapts the dependency graph. The in addition generated accumulated meta-diff Δ_ϱ comprising all adapted elements w.r.t. to ϱ is returned to the PLATΩ system.

If an element has been `Renamed`, then `locutor` automatically updates all dependent elements by adapting the respective OMDOC `ref` elements. For example, let the document D at revision 512 (denoted by D_{512}) contain $\forall A, B, C.\ A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ and the definition of $=$. If an author syntactically modifies the definiendum $=$, the `locutor` system will automatically propagate the changes along the logical dependency graph to all affected elements and return the respective meta-diffs. Thus PLATΩ is able to trigger the whole renaming in the mathematical assistance system instead of propagating each single modification which in general invalidates previously computed verifications. However, if an author accidentally renames a bound variable, for example, the bound variable A in $=$ to C, `locutor` infers the conservative change relation *Conservative*(FOR) and will automatically adjust the respective definition.

If an element has been `Moved`, for example, from one theory to another, then `locutor` will by this classification automatically update all dependent elements. For example, all values of OMDOC `ref` elements are adapted to the new location.

If an element has been `Inlined` at one certain location, i.e. expanding an element at call side and removing the element itself, then `locutor` automatically propagates this change to all dependent elements and updates the respective `ref` elements. For example, instead of entirely removing $=$ the author may perform an inlining, i.e. replacing each occurrence of $=$ by its definition. In this case `locutor` performs a parallel capture-avoiding substitution along the relations of the dependency graph. That is, we obtain $\forall A, B, C.\ (A \cap (B \cup C) \subset (A \cap B) \cup (A \cap C)) \wedge ((A \cap B) \cup (A \cap C) \subset A \cap (B \cup C))$. Regarding OMDOC, inlining means replacing all references to a `symbol` element (e.g. OPENMATH `OMS` elements) by its corresponding definition.

If an element has been completely `Removed`, then `locutor` accumulates all occurrences and notifies the author. For example, removing the definition of $=$ because of an existing eq funtion, leads `locutor` again to propagate the changes to all affected elements, resulting in $\forall A, B, C.\ \mathrm{eq}(A \cap (B \cup C), (A \cap B) \cup (A \cap C))$.

If an element has been `Replaced` by a new one, then `locutor` updates all elements depending on the removed one by adapting their references. For example, replacing $\wedge$ by an already existent type-conform connective land (e.g. *Refactored(*`Replaced[∧/land]`*)*), causes `locutor` to substitute $\wedge$ by land in all affected elements, resulting in $\forall A, B.\ A = B \Leftrightarrow \mathrm{land}(A \subset B, B \subset A)$.

6.3 Semantics

If an element is semantically changed, then `locutor` will accumulate all occurrences and notify the author to reverify the respective dependent elements. For example, if $\subset$ is semantically changed (e.g. semantic modification on the definiens), `locutor` will accumulate all occurrences of $\subset$ and notify the author to re-verify the respective elements (and $\subset$ itself). In addition, if one changes the usage of $\subset$, e.g. by mistake, then `locutor` notifies the author, before PLATΩ has to re-compute the internal data structures.

7 Conclusion and Outlook

We have outlined the integration of the ontology-driven management of change system `locutor` into the elaborated interactive mathematical authoring framework PLATΩ. Automatic classification of changes can be worthwhile comparing to time-consuming computations in the worst case accounted to "slips of the pen".

[3] The concrete specification of a meta-diff is still under investigation.

The integration is at the moment at an early stage of development: The communication between the two systems has been discussed so far. That is, the requirements of the PLATΩ system to the `locutor` system are well understood. The next step is the specification and implementation of the herein described conflation. By accomplishing this task, the authors are confident in both identifying further requirements regarding the communication and continuing improving both systems.

Acknowledgments

The authors would like to thank Michael Kohlhase, Serge Autexier, and Florian Rabe for stimulating discussions in the first phase of this work.

References

[Autexier and Dietrich, 2006] S. Autexier and D. Dietrich. Synthesizing proof planning methods and Ωants agents from mathematical knowledge. In J. Borwein and B. Farmer, editors, *Proceedings of MKM'06*, volume 4108 of *LNAI*, pages 94–109. Springer, 2006.

[Autexier and Hutter, 2005] S. Autexier and D. Hutter. Formal software development in Maya. In D. Hutter and W. Stephan, editors, *Festschrift in Honor of J. Siekmann*, volume 2605 of *LNAI*. Springer, February 2005.

[Autexier *et al.*, 2007] Serge Autexier, Armin Fiedler, Thomas Neumann, and Marc Wagner. Supporting user-defined notations when integrating scientific text-editors with proof assistance systems. In Manuel Kauers, Manfred Kerber, Robert Miner, and Wolfgang Windsteiger, editors, *Towards Mechanized Mathematical Assistants*, LNAI. Springer, june 2007.

[Autexier, 2005] S. Autexier. The CoRe calculus. In R. Nieuwenhuis, editor, *Proceedings of CADE-20*, LNAI 3632, Tallinn, Estonia, July 2005. Springer.

[Buswell *et al.*, 2004] Stephen Buswell, Olga Caprotti, David P. Carlisle, Michael C. Dewar, Marc Gaetano, and Michael Kohlhase. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004. `http://www.openmath.org/standard/om20`.

[CNX, 2007] CONNEXIONS. Project homepage at `http://www.cnx.org`, seen February 2007.

[Dietrich, 2006] D. Dietrich. The Task Layer of the ΩMEGASystem. Diploma thesis, Saarland University, Saarbrücken, Germany, 2006.

[Eberhardt and Kohlhase, 2004] Frederick Eberhardt and Michael Kohlhase. A Document-Sensitive XML-CVS Client. unpublished KWARC blue notes, 2004.

[Hendricks and Galvan, 2007] Brent Hendricks and Adan Galvan. The Connexions Markup Language (CNXML). `http://cnx.org/aboutus/technology/cnxml/`, 2007. Seen June 2007.

[Huang, 1996] X. Huang. *Human Oriented Proof Presentation: A Reconstructive Approach*. Number 112 in DISKI. Infix, Sankt Augustin, Germany, 1996.

[Kohlhase and Anghelache, 2003] Michael Kohlhase and Romeo Anghelache. Towards collaborative content management and version control for structured mathematical knowledge. In Andrea Asperti, Bruno Buchberber, and James Harold Davenport, editors, *Mathematical Knowledge Management, MKM'03*, number 2594 in LNCS, pages 147–161. Springer Verlag, 2003.

[Kohlhase, 2006] M. Kohlhase. *OMDOC - An Open Markup Format for Mathematical Documents [Version 1.2]*, volume 4180 of *LNAI*. Springer, August 2006.

[locutor, 2007] `locutor`: An Ontology-Based Management of Change, seen June 2007. system homepage at `http://www.kwarc.info/projects/locutor/`.

[MMiSS, 2007] MMiSS: Multimedia in Safe and Secure Systems. Web site at `www.mmiss.de`, seen July 2007.

[Müller, 2006] Normen Müller. An Ontology-Driven Management of Change. In *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) conference proceedings*, 2006.

[Müller, 2007] Normen Müller. Towards an Ontology-Driven Management of Change. Exposé of PhD research proposal, June 2007.

[plato, 2007] Interactive Mathematical Authoring with PLATO, seen June 2007. System homepage at `http://www.ags.uni-sb.de/plato/bin/view.pl`.

[Siekmann *et al.*, 2002] J. Siekmann, C. Benzmüller, A. Fiedler, A. Meier, and M. Pollet. Proof development with OMEGA: $\sqrt{2}$ is irrational. In M. Baaz and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 9th International Conference, LPAR 2002*, number 2514 in LNAI, pages 367–387. Springer, 2002.

[Verbert and Duval, 2004] Katrien Verbert and Erik Duval. Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models. In *Proceedings of the EDMEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 202–208, 2004.

[W3C, 2000] W3C. XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). `http://www.w3.org/TR/xhtml1/`, 2000. Seen July 2007.

[W3C, 2003] W3C. Mathematical Markup Language (MathML) Version 2.0 (Second Edition). `http://www.w3.org/TR/MathML2/`, 2003. Seen July 2007.

[W3C, 2007] W3C. XML Inclusions (XInclude) Version 1.0 (Second Edition), seen July 2007. http://www.w3.org/TR/xinclude/.

[Wagner *et al.*, 2006] M. Wagner, S. Autexier, and C. Benzmüller. PLATΩ: A mediator between text-editors and proof assistance systems. In C. Benzmüller S. Autexier, editor, *7th Workshop on User Interfaces for Theorem Provers (UITP'06)*, ENTCS. Elsevier, August 2006.

[Wiedijk, 2004] F. Wiedijk. Formal proof sketches. In S. Berardi, M. Coppo, and F. Damiani, editors, *Types for Proofs and Programs: Third International Workshop, TYPES 2003*, LNCS 3085, pages 378–393, Torino, Italy, 2004. Springer.

A Domain Independent System Architecture for Sharing Experience

Kerstin Bach Meike Reichle Klaus-Dieter Althoff

University of Hildesheim
Institute of Computer Science
Intelligent Information Systems Lab
D-31141, Hildesheim, Germany
{bach|reichle|althoff}@iis.uni-hildesheim.de

Abstract

We propose SEASALT, an architecture based on the CoMES approach on developing collaborative multi-expert-systems using case-based reasoning and software agents technology. SEASALT is built on a modular structure and will allow implementing intelligent information systems for different kinds of application scenarios based on the architecture we are presenting. Introducing SEASALT furthers knowledge intensive services based on distributed knowledge sources. In our approach we integrate "knowledge work" in a community eliciting new information. Hence, in SEASALT agents act alongside with human beings on a community platform proving and receiving information.

1 Introduction

The SEASALT (Sharing Experience using an Agent-based System Architecture LayouT) project is intended as a long term project that aims to develop a domain-independent and versatile architecture for intelligent information systems and subsequently specify and develop its individual components and finally implement it for different application scenarios. Our work concerns the topics of knowledge engineering and knowledge management by describing an application independent architecture for intelligent information systems aimed at distributed multi-expert environments. Its main contribution is in the field of collaborative knowledge maintenance, knowledge acquisition from natural language communication, and agent technology in knowledge acquisition and maintenance. Our focus lies on domains that come with a surrounding community or can easily initiate an according community. This means that the overall domain should have a certain social relevance as well as it should be useful for the individual members of the community. Examples for such domains will be presented in this paper in section 5.

Since the knowledge managed by our systems is held by a community rather than by one or a few single experts we will aim at designing the systems' underlying architecture to involve collecting the knowledge from the community members and motivating them to provide their knowledge. The system is assembling and transforming the communities' knowledge into a well defined structure so it can easily be maintained and queried.

As a model for our architecture we use the Collaborative Multi-Expert-System (CoMES) approach as described in [Althoff *et al.*, 2007]. We have chosen the CoMES approach since we consider it to be both, easily applicable to different domains and also fitting in its focus on collaborative multi-expert environments. (In our opinion a community can be regarded as collaborating experts.) Our architecture has been developed in a bottom-up procedure by applying the CoMES approach to a first application scenario to receive an architecture draft. The draft was then subsequently generalized by mapping the draft to other scenarios and extending the architecture so it can also be applied to the new scenario. The scenarios used to develop the architecture were taken from already existing research projects, the criteria on which the scenarios are chosen match are described in section 4.1, two of them are presented in section 5.

By modeling the according domain in a modularized way we can easily split it in individual topics that can then be maintained by individual case factories as described in [Althoff *et al.*, 2006]. This distributed approach allows us to more easily incorporate information collected within our community, since this information mostly does not provide complete cases of the desired domain but rather partial information that has to be assembled and combined before it can be used in a conventional information system. Modeling the system's individual components as agents additionally contributes to its flexibility and helps us to accommodate the complexities of community provided knowledge.

In this paper we will firstly present prior work that has already been done in connection to advancing the CoMES approach. Next we will further specify the CoMES approach by concretizing it and transforming it into a first architecture layout and giving a detailed description of its individual components. Then we will demonstrate the architecture's adequacy by applying it to different application scenarios that have been developed within our research group. Finally we will give an outlook on the further proceedings of our work.

2 Collaborative Multi-Expert-Systems

Collaborative Multi-Expert-Systems (CoMES, see [Althoff *et al.*, 2007]) denote a new research approach that is both, a continuation of the well-known expert system approach and a research direction based on the ideas of case factory and knowledge-line [Althoff *et al.*, 2006; 2005].

In the *Knowledge-line* concept we systematically apply the software product-line approach [van der Linden *et al.*, 2007] from software engineering to the knowledge of knowledge-based systems. This enables the necessary "knowledge level modularization" for building potential variants in the sense of software product-lines. The modularization can be achieved by making use of multi-agent systems [Burkhard, 2003; Weiß, 1999] as a basic approach for knowledge-based systems. An intelligent agent – as a first approximation – is implemented as a case-based reasoning (CBR) system [Althoff, 2001], which, besides case-specific knowledge, can also include other kinds of knowledge. Each CBR agent is embedded in a case factory that is responsible for all necessary knowledge processes like knowledge inflow, knowledge outflow as well as knowledge analysis.

A *Case Factory* (CF) is an organizational unit that emulates the well-known experience factory approach [Basili *et al.*, 1994] from software engineering. Each role within an experience factory motivates the introduction of one or more software agents for carrying out automatable (sub-)tasks more and more independently. Like the CBR agents, the associated respective CF agents are intended to learn from experience. For example, they could implement machine learning techniques for analyzing, evaluating, and maintaining the case base of the CBR system agent. Usually each software agent has a human coach, namely the one being responsible for the role (in the sense of the experience factory approach), jointly taking over the respective assigned tasks. The human coach provides case-specific knowledge for the case base of the assigned CF agent(s) as well as feedback to suggested decisions. The coach's motivation for providing knowledge is to make the CF agents as competent as possible to transfer more and more routine tasks to these CF agents. Of course, overall responsibility and control remain with the human decision maker.

While many early (and also some current) expert systems had the problem of acquiring and maintaining their knowledge, the underlying idea in CoMES is to "develop CoMES where knowledge is produced": for example in knowledge communities already available in the World Wide Web. CoMESs do not necessarily try to integrate knowledge from different sources/experts but based on many experts. As a consequence, they do not have the immediate goal to formally represent the whole knowledge necessary to solve problems associated with a given task. Instead the idea is to learn step by step based on prior experiences or case-specific knowledge provided by cooperative authors. Another idea is to keep the resulting learning scenarios/tasks as simple as possible, thus having more agents and having each one learning in a rather simple way. In this context we are thinking of computer science techniques focusing on experience like case-based reasoning, experience management, case factory, machine and human learning, cognitive architectures, etc.

3 Prior Work

In this section we present prior work which supports realizing SEASALT. We will introduce techniques and explain how they can be applied to further the implementation of SEASALT. The approaches described in this section will be specific and concentrate on a single facet.

3.1 Domain Modeling for Unstructured Texts

In SEASALT we aim to process contributions given in web communities to model the provided knowledge. Since we do not know what kind of data can be expected, we have to deal with unstructured texts of different domains. We decided to use Textual Case-Based Reasoning (TCBR) to capture the given information and formalize it [Lenz *et al.*, 1998]. Before a TCBR-System can be set up, a domain model has to be defined to ensure that the data on which the system is based can be accessed. Hence, analyzing unstructured texts requires a domain model containing terms to represent texts. The terms contained in a vocabulary have to be both, domain specific and domain independent [Roth-Berghofer, 2003]. Since we do not know what kind of text (topic, used terms, etc.) is used in the contributions we have to create a vocabulary repository containing both types of terms. When dealing with large databases containing unstructured texts, building a domain model is a very time-consuming task. [Bach, 2007b] describes how domain modeling for TCBR can be realized for a case base containing more than 9.500 cases with 2.2 million words. The paper describes how heterogeneous repositories are merged to build a vocabulary repository of general and domain specific terms to facilitate the detection of unknown words. Therefore vocabulary repositories based on GermaNet[1] and Web Services provided by *Projekt Deutscher Wortschatz*[2] were used.

Since we expect SEASALT dealing with a large amount of unstructured texts we will need a semi-automatic support for searching and modeling new terms as well. In [Bach and Hanft, 2007] the Textual Coverage Rate (TCR) is introduced, which is a method to determine the IE coverage of unstructured texts using a given vocabulary. This empowers a knowledge engineer to decide which parts of the corpus should be modeled first and how much should be done to achieve a certain quality of modeling which means coverage of unstructured text through IEs. Another feature of a vocabulary repository is the provision of synonyms which allow connections to find similar words. In SEASALT we receive information of a World Wide Web community and we expect dealing with misspelling/mistyping of terms. Resolving misspelled words we will have to create a misspelling repository in order to recognize and suggest corrections. Misspelled words and their corrections will be stored during the modeling process and during semi-automated processes analyzing unknown texts, former corrections can be used to suggest the correct word to the knowledge engineer. In automated processing the vocabulary is used to find correctly spelled words and continue using them.

Although the domain modeling for unstructured text is only one part to formalize unknown unstructured texts (semi-)automatically, it will help us to gather information of the community. We still miss the definition of connections between terms beyond synonyms to find similar documents.

3.2 Combination of Distributed Information

Prior work on the combination of distributed information on a complex domain, such as Free/Libre Open Source Software (FLOSS) has been done in [Reichle, 2007]. Here information about FLOSS applications is gathered from different sources such as public FLOSS directories, GNU/Linux distributions, collaboratively maintained software tags and bug tracking systems. [Reichle, 2007] describes how to combine and unify these different information sources and transform them into a general model for FLOSS, that contains not only the most relevant information on a FLOSS application but also weights the different attributes and provides individually adapted similarity measures, so the model can be used to set up a CBR system's

[1] http://www.sfs.uni-tuebingen.de/lsd/

[2] http://wortschatz.uni-leipzig.de/Webservices/

case base.

While this work already offers useful insights on how to combine different information sources in this rather complex domain, it lacks the flexibility that the SEASALT architecture provides. In the approach presented there, all information is merged by different scripts and parsers beforehand and then inserted into a database and subsequently imported into the case base of an empolis e:IAS system. This way the combination of the different information sources is already predefined and can not be adapted to for example different kinds of questions or information needs. Also such a huge, monolithic case base is harder to maintain, especially if automatic case base maintenance is to be used [Roth-Berghofer, 2003]. An architecture like this makes it harder to automatically insert new information. This is especially true if this information does not come in complete case descriptions but is collected from a real live community and thus maybe fragmented or incomplete. The SEASALT approach of storing information in a modularized way can make use of this work's findings but also extend them to suit a more flexible system architecture.

4 The Architecture of SEASALT

4.1 Individual Architecture Units

This section describes each unit of the SEASALT architecture. The descriptions follow the logical order starting with a given task which is processed using the architecture.

The architecture can be vertically split in two parts as can be seen in figure 4.1. On the left hand side the knowledge provision and on the right hand side the knowledge acquisition. First we will focus on the knowledge provision and explain how a question to the system will be processed. A user enters a question using the *Interface* which passes the question on to the *Coordination Agent*. The *Coordination Agent* analyzes the question, looks up the matching *Topic Agent(s)* and sends its requests to them. A response based on the existing case base is created by each *Topic Agent* and passed back to the *Coordination Agent*. Finally, the response of the *Topic Agents* is used by the *Coordination Agent* to compile an answer.

To exemplify this, imagine the *Knowledge Line* as a cabinet with the *Coordination Agent* as its chancellor or president. Each member of the cabinet is concerned with a different department and has a large staff, that provides them with information, updates it and make simple decisions in the name of their respective cabinet member. In our architecture the members of the cabinet would be the *Topic Agents*, their staff would be each *Topic Agent's* individual *Case Factory*.

The knowledge acquisition process is illustrated on the right hand side of our architecture diagram. It consists of two parts: a platform that supports the community and offers communication services where community members can discuss and exchange experiences and a knowledge engineering part in which contributions are analyzed and processed. Based on the *Topic Agents* we will place *Collectors* belonging to a *Topic Agent* in the community to collect information. Since the information given in the community is not structured, a *Knowledge Engineer* has to support the *Collector's* work. The *Knowledge Engineer* will receive and formalize contributions that the *Collectors* classify to be useful for their individual topic. In the beginning this has to be a human being, but the *Knowledge Engineer* trains an *Apprentice* agent by providing it with a document set, consisting of the source contribution and the structured document. Thus the *Apprentice* will soon be able to do at least basic pre-processing for the *Knowledge Engineer*. The formalized contribution is passed on to the *Case Factory* to include the new cases and make them available to the system. The original contributions that are provided with the new cases can be used by the system to support its answers. Additionally the community platform which is used by the *Collectors* to collect information also offers agents for intelligent services that make the platform more usable, ease the communication and accelerate conclusions on topics.

It is necessary that the information provision and the community platform are presented and perceived as a single entity. Thereby the platform's intelligent services can be an additional motivation for the community members to not only passively let the system collect information, but to directly provide it with complete cases or feedback on the knowledge it already has collected. This is especially important, since it may be assumed that the system's knowledge provision component itself will mainly be used by those community members that have less knowledge on the domain, while those who are experts on the domain and have much knowledge about it (the community's *regulars* or *core group*) will use it less often. Those regulars however will make more use of the community platform and thus they will also more appreciate its intelligent services and be more motivated by them to actively provide the system with their knowledge.

Application Scenario

The motivation of users joining and working in the community is crucial for a successful application of SEASALT. Therefore, application scenarios in which an intelligent information system using our architecture is implemented should fulfill the following two conditions:

- *Socially relevant*, the topic should be relevant enough to ensure a community which addresses enough people who can join and share their knowledge.
- *Individually useful*, so each member of the community can ask their own questions and receives satisfying answers and is thus motivated to further contribute to the system's knowledge base.

Defining the scenario's tasks and domains should focus on both conditions to create a good running community.

The application scenario's knowledge domain should fit the following characteristics:

- Suitable for community maintenance: all information contained is accessible to everybody,
- Modular structure of knowledge to facilitate assignment to *Topic Agents*,
- Well-defined range of topics with typical questions.

Interface

The communication between the user and the knowledge providing components is enabled using a Human Computer Interface (HCI), for example, containing forms or text boxes. The *Interface* can be either a website or a client application which passes the question on to the *Coordination Agent*. The *Interface* depends on the structure of the domain data and supports the structuring of questions ensuring the *Coordination Agent* can handle them. After processing the question the *Interface* will be used to display the answer.

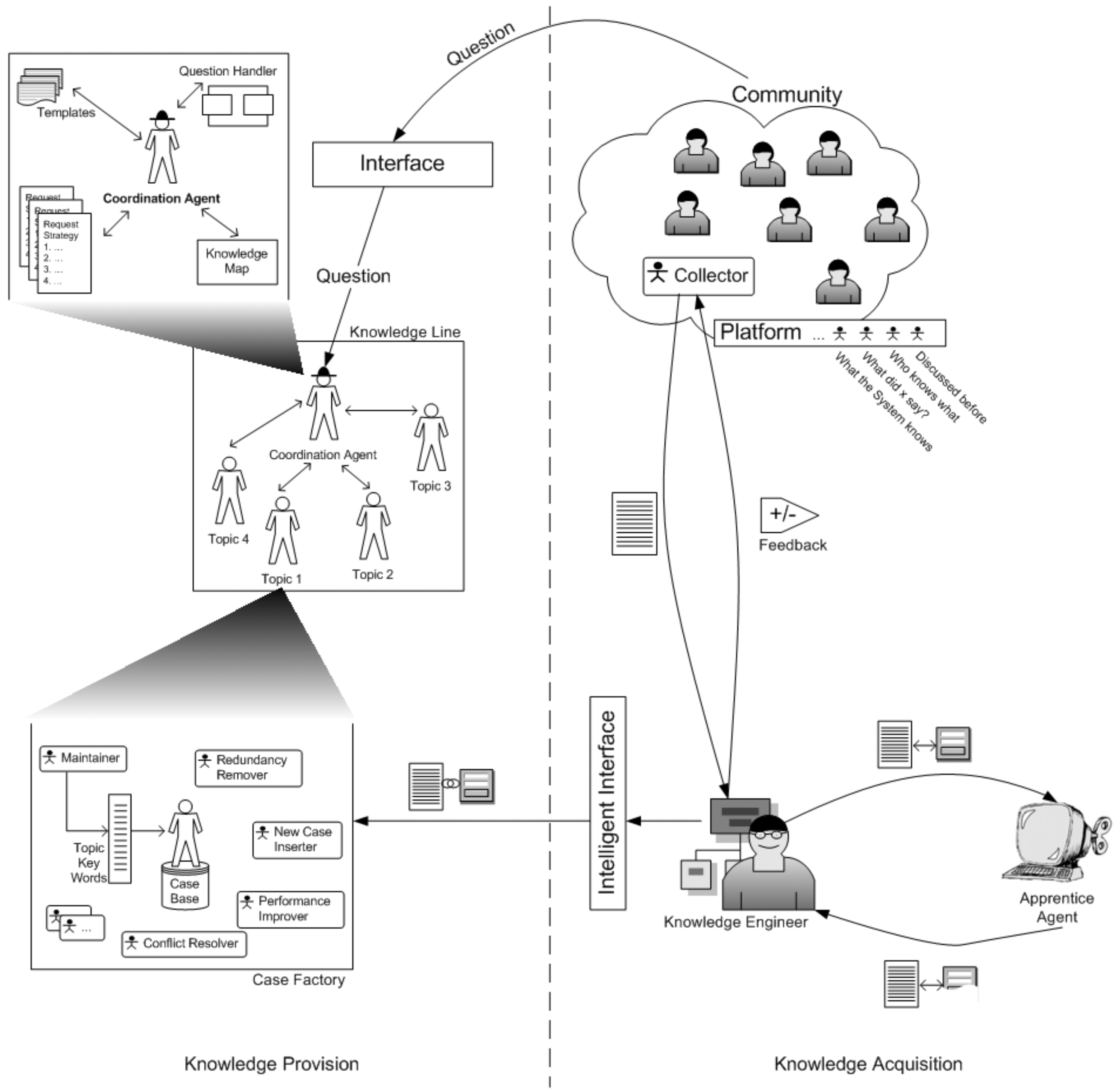


Figure 1: The SEASALT architecture

Knowledge Line
A knowledge domain that has the above mentioned characteristics allows building numerous *Topic Agents* that can be coordinated by one *Coordination Agent*. A clearly defined area of expertise for each *Topic Agent* is needed. Following a modular structure enables realizing a distributed multi-expert-system.

The Knowledge line consists of several *Topic Agents* whereas each agent is responsible for one individual topic. Each *Topic Agent* is implemented as a CBR system, has an own *Case Factory* to manage its knowledge, and is equipped with a list of topic key words to communicate its area of expertise to the *Collector* agent.

Coordination Agent
The *Coordination Agent* receives the user's questions from the *Interface* and communicates to individual *Topic Agents*. It possesses a Question Handler to distinguish the type of question and select a Request Strategy according to the question. Request Strategies indicate in what order to query individual *Topic Agents* and how to combine their results for creating a human readable answer. Furthermore the *Coordination Agent* has a Knowledge Map indicating which *Topic Agent* is competent on which topic and how to contact it. To generate the human readable answer, templates are provided which will be filled with the combined responses of the contacted *Topic Agents*.

Case Factory
The individual *Topic Agents'* case bases are maintained by several *Case Factory Agents* which serve different tasks such as inserting new cases, removing redundancies, resolving conflicts, improving the case base's overall performance, maintaining the topic keywords list, etc. Additionally each *Topic Agent* has a collector agent that is equipped with the keyword list and tries to identify relevant contri-

butions or contributions on the community platform which are delivered to the *Knowledge Engineer*. Each contribution passed on to the *Knowledge Engineer* is analyzed and the *Knowledge Engineer* will give a feedback about its actual relevance to the *Collector* so the agent can improve its collection strategy.

Furthermore, each *Topic Agent* has a case base log where every *Case Factory Agent* notes all actions performed on the *Topic Agent's* case base and why they were carried out, ensuring all decisions and actions can be retraced.

Collector

One *Collector* agent per *Topic Agent* is placed in the community. The *Collectors'* task is to collect community contributions that are relevant to their respective *Topic Agent's* area of expertise. Their choices are based on the respective *Topic Agent's* key word list. The contributions identified as relevant are passed on to the *Knowledge Engineer*. The *Collectors* then receive feedback on their choices from the *Knowledge Engineer*, thus improving their performance over time.

Knowledge Engineer

The role of the *Knowledge Engineer* connects the *Case Factories* and the *Community* enabling the exchange of knowledge between both of them. The *Knowledge Engineer* is an expert on the system's tasks and also takes part in the Community. He receives contributions collected on the community platform by the different *Collector* agents and gives each *Collector* agent a feedback on its decisions. Furthermore the *Knowledge Engineer* formalizes relevant contributions, so the according *Case Factory Agent* can insert the new cases into its *Topic Agent's* case base. The formalization is supported by an intelligent interface that offers features like controlled vocabulary, spell checks, synonyms, etc. as it is presented in [Bach, 2007b; Bach and Hanft, 2007].

The *Knowledge Engineer* is supported by an *Apprentice* agent. To train the *Apprentice* the *Knowledge Engineer* uses relevant contributions and their formalizations (document sets). In return the *Apprentice* is able to do at least basic pre-processing and can take more and more work off the *Knowledge Engineer* as its training continues.

Since these are only roles, the tasks of the *Knowledge Engineer* can of course also be executed by more than one person as well as the *Apprentice* can be more than one agent.

Apprentice

The *Apprentice* is an agent or numerous agents, that is trained with document sets by the *Knowledge Engineer*. We aim to automate as many tasks as possible. The *Apprentice* is at least able to do preparatory pre-processing and is meant to take more and more work off the *Knowledge Engineer*.

Community

The Community consists of a group of people who discuss several topics of a common domain. A communication *Platform* is used to discuss and exchange opinions. It is extended with several *Collector* agents, that collect relevant contributions on their topics. The people participating in the community will be able to use the given system.

Platform

The *Platform* is used by the *Community* to discuss and exchange opinions and by the *Collector* agents to collect relevant contributions. It offers agents of its own that perform intelligent services for the platform's users such as:

- „This has been discussed before" – Pointing out to users if similar discussions have been had before
- „X might be able to help you on this" – Learning which user is interested in or an expert on a certain topic. (Interested users start threads on topic, experts answer in threads on topic.)
- „What does X think about that?" – Selecting all contributions by a specific person according to a specified topic
- „What the system knows" – Can tell what data the knowledge-based system has on a specific topic

An example for such intelligent services is presented in [Feng *et al.*, 2006].

5 Practical Applications of the SEASALT Architecture

After introducing the architecture we will now describe two application scenarios which will be implemented using SEASALT. Both application scenarios should substantiate the architecture and clarify the usage of the CoMES approach. Further their implementation and testing will serve as a first evaluation and help to improve the architecture and detect flaws in the concept.

5.1 docQuery - An Intelligent Information System on Travel Medicine[3]

Nowadays it has never been easier to travel to different places, experience new cultures and get to know new people. In preparation for a healthy journey it is important to acquire high quality of reliable information about travel medicine prevention. Travel medicine is specialized on medical issues like diseases, vaccinations, etc. which might occur before, during and after a journey.

Currently the World Wide Web offers many websites, discussion forums and services where a traveler can gather information. Usually those websites do not contain all medical information a traveler might need and the editors are mostly unknown. Furthermore the information is spread over hundreds of websites and it is challenging and time-consuming to find appropriate information [Bach, 2007a].

Together with a team of certified doctors of medicine with a strong background of travel related medicine, we will use SEASALT to implement *docQuery*, an application to provide travel information based on travelers' key data such as travel period, destination, age(s) of traveler(s), activities, etc. [Bach, 2007a]. We aim at establishing a community of experts exchanging knowledge on their expertise and getting new information from their colleagues. *docQuery* will be supervised by experts and integrate the experiences of travelers using the given advices to improve the quality and early identify new issues.

Implementing *docQuery* using SEASALT will help both, creating a community to exchange knowledge and offering an multi-expert-system on travel medicine tasks. In

[3] This is a project in cooperation with Thomas Schmidt (mediScon worldwide).

the travel medicine domain various topics like medications, countries, diseases, etc. can be distinguished and will each be found in an individual *Topic Agent*.

5.2 FLOSSWALD – Information System on Free and Open Source Software

Free/Libre and open source software (FLOSS) has produced a large and diverse range of software which offers numerous and high quality alternatives to almost all commercial software applications. Popular FLOSS like Firefox[4], Thunderbird[5], OpenOffice.org[6], vlc[7] or the GNU/Linux operating system are steadily gaining users both in the private and commercial sector.

However the FLOSS community is a complex social and technical network that consists of tens of thousands of individual groups and projects that produce software in all degrees of quality. Research [Reichle, 2007] shows, existing FLOSS directories are mostly used by expert users and FLOSS insiders, while less experienced users prefer general search engine and the advice of friends when choosing software. The success of choosing software in this way is limited though. A general web search for software for a specific purpose will most likely yield the most popular software (or rather that with the most popular website) but not necessarily the one that is best suited for the given purpose and user. Asking friends or colleagues for software advice is also of only limited use, since those usually have the same level of knowledge as the seeking person and can thus not offer qualified advice.

The FLOSSWALD System [Reichle and Hanft, 2006; Hanft and Reichle, 2007] aims to be an intelligent information system on free/libre open source software (FLOSS), that offers the community's knowledge and experiences with different software to unexperienced users and can be queried using natural language and simple menus. To use the system, the user does not have to be an expert on software or computers in general. In order to achieve this the system combines information collected from different FLOSS directories by the FLOSSmole Project [Howison *et al.*, 2006] with data provided by GNU/Linux distributions (in this case the Debian Project's Package data). This knowledge can be enriched using the DebTags, a collection of collaboratively maintained tags covering different aspects of software [Zini, 2005]. Additionally more user-friendly attributes such as tasks that a software can be used for "vague attributes" like user friendliness, flexibility or stability that can be learned from the community. The combination of these different knowledge sources and the problem of keeping them up-to-date provides an adequate application scenario for the SEASALT architecture.

5.3 Implementation Details

Both application scenarios, along with others, will base on SEASALT and use basically the same technologies. Raw data will be contained in a DBMS like PostgreSQL[8] and the *Case Factory* will be realized using e:IAS.

e:IAS is an information and knowledge management suite developed by empolis[9], a subsidiary of Bertelsmann Arvato [empolis GmbH, 2005]. e:IAS consists of several different components for information and knowledge processing and management. Among these there is a rule engine, which can be used to model business processes and classification tasks (via rules). Hence, it contains a text miner that can be used for analyzing documents as well as it offers free text user queries, which can further be combined with a downstream pattern matching component.

e:IAS also includes a powerful CBR engine and the so called creator module, which is based on the free IDE eclipse[10]. The creator is used to model the cases for the CBR engine. A case is modeled as an aggregate of attributes. The creator is used to model the required attributes, their domains and also underlying concepts and taxonomies and the respective attributes' similarity functions. The model is stored using RDF respectively OWL, all further information is stored in an XML format. The case models in e:IAS can be filled with imported data from a multitude of sources. The data import and their further processing is done using a modular pipeline system in which the different functionalities can be freely combined using individual pipelets. Pipelets offer for example the import from simple text files, documents, databases or websites using an integrated crawler, and their subsequent processing such as breaking the input data into single values and assigning them to their respective attributes, analyzing texts with a text miner or spell checker or stripping input of html or xml elements. Once the data are imported and processed the system's knowledge base is ready and can be used by the e:IAS knowledge server. Pipelines are however not only used for importing data, but also for integrating the aforementioned functionalities and making them available to the Knowledge Server. e:IAS offers pipelets for text mining, the creation and application of rule sets, searching, automated classification and the generation of dialogs. Additionally the Knowledge Server is able to use external knowledge sources such as already existing dictionaries. Different types of clients including simple web clients but also rich clients or JavaBean applications can be used to access the Knowledge Server. Communication between the server and its clients can be implemented using various languages such as XML, COM or RMI.

6 Conclusion and Future Work

6.1 Conclusion

In this paper we have presented an architecture that follows the CoMES approach and connects case-based reasoning, software agent technologies and the acquisition of knowledge distributed in a community.

We also described the collaboration of different experts as well as the integration of software agents in a (given) community. To further knowledge exchange we extend common community platform features with several intelligent services executed by software agents. Alongside integrating knowledge work in a community, we have described how to use the elicited knowledge in an intelligent information system. Developing our architecture we follow a modular approach by creating *Topic Agents* for each specific topic and combining them to create more complex answers. Hence we specified each unit of SEASALT and assigned tasks they have to perform. To evaluate our architecture we presented two application scenarios which can be implemented using SEASALT.

[4] http://www.mozilla-europe.org/en/products/firefox/

[5] http://www.mozilla-europe.org/products/thunderbird/

[6] http://www.openoffice.org/

[7] http://www.videolan.org/vlc/

[8] http://www.postgresql.org/

[9] http://www.empolis.com

[10] http://www.eclipse.org/

6.2 Scientific Contribution

In this work we focused on knowledge engineering and knowledge management by describing an application independent architecture for intelligent information systems aimed at distributed multi-expert environments. Agent technology is applied in the case factory agents, the collector agent, and the apprentice agent. The work of the knowledge engineer and its apprentice agent concerns the field of knowledge acquisition from natural language communication. Collaborative knowledge maintenance is realized in the implementation of the topic agents, case factory, and collector agent.

Our approach is characterized by our focus on communities and the modular structure of knowledge as represented in our topic agents which also distinguishes the SEALSALT architecture from general search engines and more monolithic approaches. Furthermore the architecture is more topic and community oriented. The use of case-based reasoning allows for easy maintenance of the knowledge base and powerful retrieval.

6.3 Future Work

Future work will go into the detailed specification and implementation of the described units of SEASALT to realize CoMES. We will first have to define general functions and units which have to be provided for any application, then implement those units and in a final step adapt them to the different application scenarios like FLOSSWALD and *docQuery*. As mentioned before, the *Case Factory* will be implemented using e:IAS representing one *Topic Agent*. The communication between agents will be realized using RMI which is already provided for e:IAS. Also the communication and tasks of the *Coordination Agent* have to be defined and implemented. We have also created the role of a *Knowledge Engineer* which trains the *Apprentice Agent*. To support the *Knowledge Engineer* by pre-processing the communities' documents we will use TCBR, but in advance we have to clarify which tasks a *Knowledge Engineer* has to fulfill and how agents can support it and adapt to those tasks. We will also have to undertake further research into issues arising from collaborative knowledge maintenance such as contradicting experiences, alternate solutions or incomplete data.

To ensure successful community work, community members have to be motivated to share their experiences and use the platform. In exchange platform services such as specific platform agents should ease communication. Hence, questions should be answered correctly so users can trust in it. Existing communities have to be introduced to our approach and the platform features need to be discussed.

References

[Althoff *et al.*, 2005] Klaus-Dieter Althoff, Jens Mänz, and Markus Nick. Integrating Case-Based Reasoning and Experience Factory: Case Studies and Implications. In Uli Furbach, editor, *Proceedings of the 28th German Conference on Artificial Intelligence Workshop on Knowledge Engineering and Software Engineering*, pages 1–12, Koblenz, Germany, 11. - 14. September 2005. Springer, LNAI 3698.

[Althoff *et al.*, 2006] Klaus-Dieter Althoff, Alexandre Hanft, and Martin Schaaf. Case Factory – Maintaining Experience to Learn. In Mehmet H. Göker, Thomas Roth-Berghofer, and H. Altay Güvenir, editors, *Proc. 8th European Conference on Case-Based Reasoning (ECCBR'06), Ölüdeniz/Fethiye, Turkey*, volume 4106 of *Lecture Notes in Computer Science*, pages 429–442, Berlin, Heidelberg, Paris, 2006. Springer Verlag.

[Althoff *et al.*, 2007] Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Regis Newo, Meike Reichle, Martin Schaaf, and Karl-Heinz Weis. Collaborative Multi-Expert-Systems – Realizing Knowlegde-Product-Lines with Case Factories and Distributed Learning Systems. Technical report, University of Osnabrück, Osnabrück, September 2007.

[Althoff, 2001] Klaus-Dieter Althoff. Case-Based Reasoning. In S.K. Chang, editor, *Handbook on Software Engineering and Knowledge Engineering. Vol.1, World Scientific*, pages 549–587. 2001.

[Bach and Hanft, 2007] Kerstin Bach and Alexandre Hanft. Domain Modeling in TCBR Systems: How to Understand a New Application Domain. In David C. Wilson and Deepak Khemani, editors, *Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR) 2007, Workshop on Knowledge Discovery and Similarity*, pages 95–103, Belfast, Northern Ireland, 2007.

[Bach, 2007a] Kerstin Bach. docQuery – Reisemedizinisches Informationssystem. Internal project report, 2007.

[Bach, 2007b] Kerstin Bach. Domänenmodellierung im Textuellen Fallbasierten Schließen. Master's thesis, Institute of Computer Science, University of Hildesheim, 2007.

[Basili *et al.*, 1994] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. Experience Factory. In John J. Marciniak, editor, *Encyclopedia of Software Engineering, vol 1*, pages 469–476. John Wiley & Sons, 1994.

[Burkhard, 2003] Hans-Dieter Burkhard. Software-Agenten. In Günther Görz, Claus-Rainer Rollinger, and Josef Schneeberger, editors, *Handbuch der Künstlichen Intelligenz, 4. Auflage*, pages 943–1020. Oldenbourg, 2003.

[empolis GmbH, 2005] empolis GmbH. Technisches White Paper e:Information Access Suite. Technical report, empolis GmbH, September 2005.

[Feng *et al.*, 2006] Donghui Feng, Erin Shaw, Jihie Kim, and Eduard Hovy. An intelligent discussion-bot for answering student queries in threaded discussions. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 171–177, New York, NY, USA, 2006. ACM Press.

[Hanft and Reichle, 2007] Alexandre Hanft and Meike Reichle. The FLOSSWALD Information System on Free and Open Source Software. In Norbert Gronau, editor, *Proceedings of the 4th Conference on Professional Knowledge Management - Experiences and Visions*, pages 135–142, Berlin, March 2007. Gito Verlag.

[Howison *et al.*, 2006] James Howison, Megan S. Conklin, and Kevin Crowston. FLOSSmole: A collaborative repository for FLOSS research data and analyses. *International Journal of Information Technology and Web Engineering*, 1(3):pages 17–26, Juli - September 2006.

[Lenz *et al.*, 1998] Mario Lenz, André Hübner, and Mirjam Kunze. Textual CBR. In Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard, and Stefan Wess, editors, *Case-Based Reasoning Technology – From Foundations to Applications*, Lecture Notes in Artificial Intelligence, LNAI 1400, pages 115–137. Springer-Verlag, Berlin, 1998.

[Reichle and Hanft, 2006] Meike Reichle and Alexandre Hanft. The FLOSSWALD Information System on Free and Open Source Software. In Martin Schaaf and Klaus-Dieter Althoff, editors, *Lernen - Wissensentdeckung - Adaptivität Proceedings of the LWA 2006, FGWM 2006 Workshop on Knowledge and Experience Management*, volume 1/2006 of *Hildesheimer Informatik-Berichte*, pages 229–233. University of Hildesheim, Oktober 2006.

[Reichle, 2007] Meike Reichle. Entwicklung und Implementierung eines Modells zum Retrieval von Free/Libre Open Source Software unter Verwendung eines Case-Based Reasoning Systems. Master Thesis (Magisterarbeit), 2007.

[Roth-Berghofer, 2003] Thomas Roth-Berghofer. *Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology, Dissertation.* PhD thesis, Universität Kaiserslautern, 2003.

[van der Linden *et al.*, 2007] Frank van der Linden, Klaus Schmid, and Eelco Rommes. *Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering*. Springer, Berlin, Heidelberg, Paris, 2007.

[Weiß, 1999] Gerhard Weiß, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999.

[Zini, 2005] Enrico Zini. A cute introduction to Debtags. In Alexander Schmehl, editor, *Proceedings of the 5th annual Debian Conference*, pages 59–74, Helsinki, Finland, 10. - 17. Juli 2005. The Debian Project.

Visualizing Patient Similarity in Clinical Decision Support

Alexey Tsymbal, Martin Huber, Sonja Zillner
Corporate Technology Division
Siemens AG, Erlangen, Germany
{alexey.tsymbal; martin.huber; sonja.zillner}@siemens.com

Tamás Hauer
CCS Research Centre, CEMS Faculty
University of the West of England
Bristol, UK
tamas.hauer@cern.ch

Kevin Zhou
Integrated Data Systems Department
Siemens Corporate Research
Princeton, NJ, USA
kzhou@scr.siemens.com

Abstract

As clinical research and practice rely more and more on analytic approaches to patient data, there is a growing challenge to build those tools that efficiently manage, analyze and visualize the vast information collected for each patient. Clinical decision support systems may well be a significant step in the right direction as long as the results are easily accessible to the clinicians. We have evaluated a number of visualization techniques which, we believe, hold promise in being adopted in the clinical workflow. Here we present a comparative analysis of three of these which we found suitable: treemaps, relative neighbourhood graphs and combined distance-/heat-maps.

1 Introduction

There is growing interest in the use of computer-based clinical decision support systems (DSSs) to reduce medical errors and to increase health care quality and efficiency [Berlin *et al.*, 2006]. Clinical DSSs vary greatly in design, functionality, and use. According to the reasoning method used in clinical DSS, one important subclass is that of Case-Based Reasoning (CBR) systems – systems which have reasoning by similarity as the central element of decision support [Berlin *et al.*, 2006; Nilsson and Sollenborn, 2004].

CBR is a recognized and well established method for building medical systems [Nilsson and Sollenborn, 2004]. It is, however, commonly acknowledged that CBR has not yet become as successful in medicine as in some other application domains. The field is evolving steadily, but slowly [Nilsson and Sollenborn, 2004; Schmidt and Vorobieva, 2005].

One known reason for this is the especial complexity of medical data, and the resulting difficulty in defining a meaningful distance function on them and adapting the final solution [Schmidt and Vorobieva, 2005]. Medicine is a domain where large and complex heterogeneous data sets are commonplace. Today, a single patient record may include, for example, demographic data, family history, laboratory test results, images (including echocardiograms, MRI, CT, angiogram, etc.), signals (e.g. EKG), genomic and proteomic samples, and history of appointments, prescriptions and interventions. Much if not all of this data may contain important information and therefore be relevant for decision support. In [Tsymbal *et al.*, 2007] a use of ontologies was considered for defining similarity on complex medical data as a way to tackle this issue.

Another commonly reported reason for the relatively slow progress of the field is the lack of transparency and explanation in medical CBR. Often, similar patients are retrieved and their diagnoses are presented, without specifying why and to what extent the patients are chosen to be similar and why a certain decision is suggested. We believe that, one way to approach this problem is to better visualize the underlying inter-patient similarity, which is the central concept of any clinical CBR.

There is a large number of information visualization techniques which have been developed over the last few decades to support the exploration of large data sets to cope with the ever increasing data flood [Keim, 2002]. Visualization of large volumes of data may lead to the discovery of novel important knowledge; such a knowledge discovery process is called visual data mining or visual data exploration [Keim, 2002]. Shneiderman and Plaisant [2004] introduce the so-called visualization mantra, which may be considered as the main principle for visual data mining: “overview first, zoom and filter, then details on demand”.

Inter-patient similarity is perhaps the core concept in any clinical CBR. However, in known CBR systems the visualization is usually limited with the visualization of case solutions and not case similarity [Mullins and Smyth, 2001]. Our main goal with this paper is to propose and compare techniques for visualizing patient similarity, which can be useful in clinical decision support. Our focus is on three specific non-traditional techniques such as treemaps [Shneiderman, 1992], relative neighbourhood graphs [Toussaint, 1980], and combined distance-/heat-maps [Verhaak *et al.*, 2006], which are arguably the most promising candidates for visualizing patient similarity.

Health-e-Child is an EU-funded Framework Programme 6 (FP6) project aimed at improving personalized healthcare in selected areas of paediatrics, especially focusing on integrating medical data across disciplines, modalities, and vertical levels such as molecular, organ, individual and population. The results presented in this paper contribute to the development of decision support systems in the project and are based on preliminary results conducted using early samples of real-life patient data collected by participating clinicians. The examples make use of a sample of 46 neuro-oncology patients.

The paper is organized as follows. In Sections 2-4, we give an overview of three techniques suitable for similarity visualization in medical DSS; treemaps, relative neighbourhood graphs, and combined distance-/heat-maps. In Section 5, a comparative analysis is presented, summarizing limitations and benefits of each technique. In Section 6, a framework for medical similarity search is considered, which integrates the three visualization techniques, and we conclude in Section 7 with a summary, open issues and further research topics.

2 TreeMaps – a Space-Filling Visualisation Technique for Hierarchies and Not Only

Treemaps [Shneiderman, 1992] were suggested as an efficient two-dimensional (2D) space-filling approach to visualize hierarchical data structures. Space filling techniques are based on the concept of using the whole of the screen area by subdividing the space available for a node among its children [Katifori *et al.*, 2007].

Treemaps allow for an intuitive visualization of containment in non-intersecting sets; both sets and elements are represented as rectangles, with elements and subsets completely filling the enclosing rectangle of the containing set or superset, much like a rectangular Venn-diagram. The name, treemap is justified as the subset and containment relations induce a tree structure on the domain of nested nonintersecting sets and elements: these are nodes of the tree and the directed edges stand for the containment/subset relationship.

The implementation of the view makes use of a simple recursive algorithm, which partitions the rectangular visualization region into a nested sequence of smaller rectangles.

The full power of this visualization technique is exploited by letting the graphics primitives correlate with attributes of the data being displayed, thus incorporating further visual aids. In practice, the colour of the rectangles correlates with one such attribute (usually one that is independent of the tree structure) and the size of the rectangle with another. Using the size of the rectangles in such a way is especially powerful when it corresponds to an *additive* measure on the enclosing sets because the size of the enclosing rectangles is then similarly meaningful. The obvious candidate for such a measure is the node's cardinality, in which case the leaves are equal-sized. Another common example of an additive measure is in file-system views; the display size is correlated with the size of the occupied disk-area, and the corresponding node's size is proportional to the cumulative size of files belonging to the parent directory [Shneiderman, 1992]. Correlating the size of the rectangles with a non-additive attribute results in views which can be misleading but this is less of a problem for data sets with small number of leaves.

In comparison to the traditional tree views where entities are listed sequentially with additional visual aids, like indentation and collapsing/expanding branches, treemaps have the advantage of better exploiting the screen space and a more natural visualization of data set elements through navigation. The global tree structure is always visible, leaving out the rectangles smaller than the visual cutoff. To access the deeper levels, the usual navigation relies on the isomorphism of the tree structure: any subtree is again a tree so that the treemap view can be simply zoomed in and out. Curiously, numerous applications of "tree-map views" are used with completely flat data, where of course this navigation capability is not exploited.

The treemap layout algorithm is linear in the number of nodes and is reported to display up to a million nodes, depending on the screen resolution [Shneiderman, 2006]. It is usually acknowledged that tree-maps visualize the size attribute best among similar techniques [Teoh, 2007]. The list of application areas for treemaps is extensive indeed; these range from file-system browsing and stock market portfolio visualization to NBA player statistics browser and a budget viewer [Shneiderman, 2006].

Systems implementing tree-maps typically support one or more *layout algorithms* that specify the arrangement of the nodes in the tree, partitioning each parent node into child nodes. This algorithm especially changes the appearance of the trees with the big cardinality of nodes, and is usually not important for binary trees. Most real-life tree structures, however, such as file directories, are not binary, thus the choice of the layout algorithm is important.

The first algorithm, originally developed at the University of Maryland to visualize file directories, is the so called *"slice and dice" layout*. With this simple layout, each parent rectangle is divided into child rectangles with parallel lines (often the use of vertical and horizontal lines alternate from level to level). The algorithm gives best results when the number of children is approximately constant across the tree (e.g. binary trees); if this is not the case then the resulting rectangles may become too thin and node comparison becomes difficult.

At present, the layout algorithm of choice in many systems is the *squarified layout* [Bruls *et al.*, 2000] which aims for all the nodes in the tree to have aspect ratio close to one. This is acknowledged to provide the best balance of structural fidelity and aesthetics.

Aspect ratio, therefore, is an important characteristic of treemap visualization. Satisfactory aspect ratio, however, usually comes at the price of stability – while the slice and dice algorithm maintains fixed positions of the nodes, the squarified algorithm changes the layout (often drastically, as the change is non-local) as the tree or the size of the leaves change, thus making it difficult to find nodes consistently at one position. This may be unwelcome in certain situations, for example when the data set is periodically updated with new instances and it is necessary to quickly locate a certain instance. The amount of change a layout undergoes as the data set is updated (stability) is thus another important metric characterizing tree-map visualization [Bederson *et al.*, 2002]. Besides these direct metrics, often indirect subjective metrics are used to evaluate tree-map visualization, based on lab experiments and interviews. One such treemap-specific metric suggested in [Bederson *et al.*, 2002] is the number of times that the motion of the reader's eye changes direction upon scanning the treemap layout.

Treemaps are useful and can be applied virtually for any kind of data hierarchy. In the context of our paper, we are primarily interested in *anonymous hierarchies* obtained from hierarchical clustering of a certain clinical data set, and visualizing patients' similarity in the context of a distance defined by a clinician. A more typical application of treemaps, though, is to so-called *lexicographic hierarchies*, where the grouping has a certain a priori meaning and thus the subsets can be assigned a label.

The subtle difference between the two classes of hierarchies usually requires different approaches in implementation. The most common kind of lexicographic hierarchies displayed is decision trees based on certain attributes in the data. The attributes which are used to create the treemap can include virtually any attribute of interest (gender, age, family history, etc.). As opposed to the clustering case where the nested sets are defined by the data itself, the lexicographic view requires that a tree structure is imposed over the range of the selected attribute to induce the nested structure on the – a priori flat – data set. When the selected data attribute is a numeric one (or one whose range is a set with a complete order) then it needs to be discretized by recursively dividing the numeric range into nonoverlapping intervals (*binning*). The ordering over the range defines a natural tree structure over these intervals, which generates the containment hierarchy over the dat set. Another type of lexicographic hierarchies that are successfully applied with treemaps is ontologies. In this case the selected attribute is a label, assuming values from a labelled tree (the ontology). For example, named regions of the human body organize naturally in a tree based on a *part_of* relationship between them. In turn, this induces a nested set hierarchy on a flat data set of cancer patients labelled by tumour location. Treemaps were recently applied to visualize the Gene Ontology [McConnell *et al.*, 2002; Baehrecke *et al.*, 2004], which is a large naturally hierarchical data structure. The main reported benefit of tree-map visualization is the ability to rapidly see patterns in large collections of data and obtain details on demand.

Besides hierarchies, treemaps have been lately often applied to visualize virtually flat data. The appealing appearance of the treemap layout is used to visualize a group of objects with certain features that can be monitored as the size and the colour of the rectangles. For example, in the "Newsmap" visualization of Google news (www.marumushi.com/apps/newsmap/newsmap.cfm) the structure (tree) is virtually absent. Instead, several groups of news are visualized as separate sub-treemaps, and each sub-map is simply a flat collection of documents arranged using the squarified layout algorithm.

It is important to note that the strengths of treemap visualization strongly depend on the particular layout algorithm of choice, and also on the software implementation (especially GUI). Many implementations have been created, open-source and closed, domain-specific and generic. Each implementation has its own advantage and, unfortunately, limitations. We will mention here a few implementations which we considered with regards to our task of patient similarity visualization. The original *Treemap* software from the University of Maryland (the latest version is 4.1, available at www.cs.umd.edu/hcil/treemap/) is perhaps the most popular one. It has many useful features including flexible hierarchies of different depth, easy to apply binning, and easy navigation through the nodes in the tree. One important limitation though, in our opinion, is that Treemap was designed for lexicographic hierarchies only, and is not particularly suitable for anonymous hierarchical clusterings.

Another popular implementation is *JTreeMap* (jtreemap.sourceforge.net). Reportedly, it is the only open-source library for treemapping, which is maintained. *JTreeMap* is more suitable for visualizing hierarchical clusterings, and is not as good as *Treemap 4.1* in visualizing lexicographic hierarchies. Usually, the group headings are visualized in the centre of the corresponding node, and it is difficult to visualize more than one level of headings. The popular open-source visualization toolkit *Prefuse* [Heer *et al.*, 2005] also includes a treemap implementation. The part related to treemaps, however, is not yet documented, and understanding the code needs considerable learning effort.

There were a number of studies comparing treemaps with alternative techniques for various visualization-related tasks. Interestingly, treemaps are usually the winner, or are at least comparable with other techniques, even with the limited implementations used. Kobsa [2004] compares treemaps with five well-known hierarchy visualization systems, and Windows Explorer as a baseline system, in the context of browsing file directories. Fifteen simple technical questions were asked for evaluation. Treemap turned out to be the best visualization system in this study. Wang *et al.* [2006] conduct a similar study, focusing though on less trivial questions (tasks) that are not easily answerable through simple automated scripts and that could be said to be knowledge discovery tasks. Treemaps are again among the winners, and their ability to lead to the discovery of useful knowledge is confirmed. Zhang *et al.* [2004] present arguably the first medical application of treemaps. Treemaps were integrated into a large brain research data warehouse to support neurologic and neuroradiologic research at the University of California, San Francisco Medical Centre. Treemaps were demonstrated to facilitate better discovery of the hidden knowledge in medical image data warehouses. Also, treemap visualization was found to be user friendly with a short learning curve, proven by feedback from collaborating neurologists, neurosurgeons and clinical researchers.

Besides its advantages, some treemap limitations were also reported. It is usually acknowledged that treemaps may be difficult for understanding the structure of the tree [Wang *et al.*, 2006]. We, however, believe that, this limitation is strongly implementation and layout algorithm – dependent, and with certain treemap configurations it is possible to avoid (or at least to minimize) this problem.

In Figure 1a a JTreeMap visualization of a hierarchical clustering of 46 Brain Tumour patients is demonstrated. The Cobweb/Classit algorithm from the Weka machine learning library [Witten and Frank, 2000] is used for clustering. The current patient is highlighted with a tooltip. Imaging-related features are used for clustering. The size of rectangles represents age at diagnosis, and the colour – the quality of life for patients after treatment [Garre *et al.*, 1994]. For the figures in this paper, the patient data was anonymized, to conform with the data confidentiality requirements.

Figure 1b displays the same patients using a two level lexicographic hierarchy: patients are divided by tumour grade and tumour location (Treemap 4.1 implementation). Again, the size of rectangles represents the age at diagnosis, whereas the colour the quality of life for patients. E.g. in the upper right corner, all patients suffering of a highly malignant grade 4 tumour are shown. All of them died of the disease as indicated by the black colour.

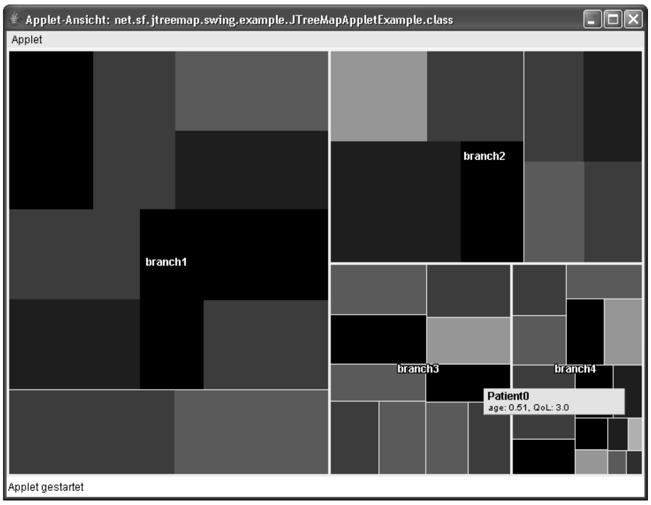


Figure 1a JTreeMap visualization of a hierarchical clustering of Brain Tumour patients

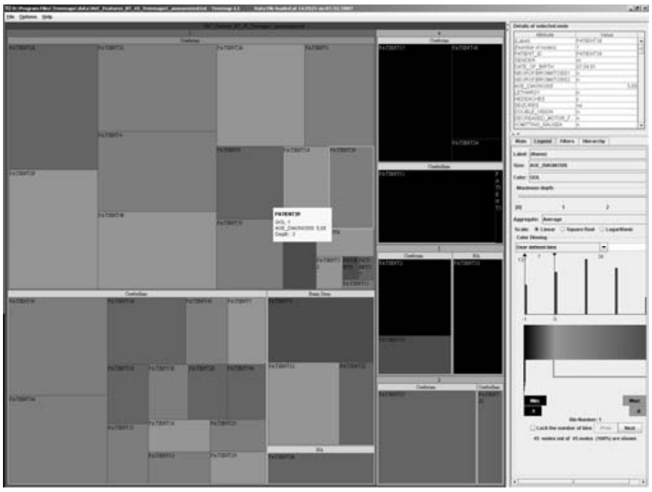


Figure 1b Treemap 4.1 visualization of a lexicographic hierarchy of Brain Tumour patients. Using a four level scale, patients with no sequelae are displayed green, patients with life threatening sequelae red, patients that died are displayed black.

3 Relative Neighbourhood Graphs

In a relative neighbourhood graph, two vertices corresponding to two instances A and B in a data set are connected with an edge, if there is no other instance C which is closer to both A and B with respect to a certain distance metric *d* [Jaromczyk and Toussaint, 1992]:

$$d(A,B) \le \min_{C \neq A,B} \max\{d(A,C), d(B,C)\} \tag{1}$$

Originally, relative neighbourhood graphs were defined for two-dimensional data (planar sets) with Euclidean distance metric, but later they were generalized and applied to multiple dimensions and other distance metrics [Toussaint, 1980; Jaromczyk and Toussaint, 1992; Muhlenbach and Rakotomalala, 2002].

Besides the relative neighbourhood graphs we focus on, there are some other known node-link (graph-based) visualizations of instance proximity. These include the Minimum spanning tree (MST), the Gabriel graph, and the Delanay tessellation [Jaromczyk and Toussaint, 1992]. We believe that the relative neighbourhood graph is the best to visualize patient proximity in a DSS among the ones mentioned above. The MST has usually too few edges to spot groupings/patterns in the data, while the Gabriel graph and the Delanay tessellation are, vice versa, usually too overcrowded, which becomes a problem with already more than 10 instances.

In machine learning, neighbourhood graphs find various applications, including clustering, outlier removal, and even supervised discretization [Muhlenbach and Rakotomalala, 2002]. In other areas, other, more exotic applications may also be found. In [Marcotegui and Beucher, 2005], for example, the minimum spanning tree of a neighbourhood graph was applied to contrast-based hierarchical image segmentation. In [Li and Hou, 2004] a directed relative neighbourhood graph and directed minimum spanning tree are successfully applied to topology control, in order to create a power-efficient network topology in wireless multi-hop networks with limited mobility.

One serious limitation of relative neighbourhood graphs which affects their scalability is the computational complexity of the algorithms that construct them. There are known solutions for speed-up for dimensionalities up to three, but in the general case the computational complexity remains $O(n^3)$, where *n* is the number of nodes (instances). One possible solution for too large data sets is to cluster them into a number of groupings, for which calculating the relative neighbourhood graph would be feasible, and then to create a neighbourhood graph separately for each of them, also connecting close clusters (the corresponding neighbourhood graphs) with each other.

In Figure 2, an example relative neighbourhood graph is presented, for the previously considered Brain Tumour data. Each node corresponds to one patient with an ID, and the colour represents gender. The visualization of relative neighbourhood graphs is implemented with the Prefuse open-source visualization toolkit [Heer *et al.*, 2005].

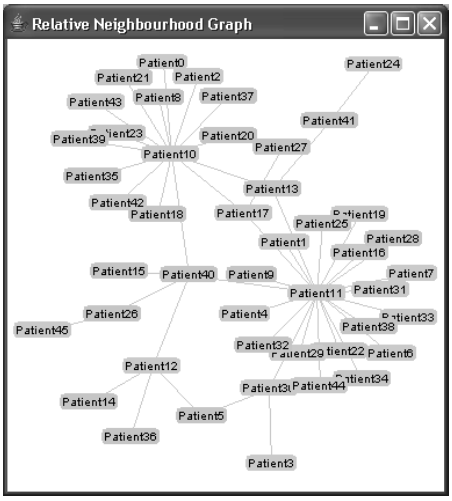


Figure 2 Relative neighbourhood graph for the Brain Tumour patients (based on MRI features)

4 Combined Distance-/ Heat-maps

Heatmaps and distance maps are two more orthodox ways of data visualization than the previously considered treemaps and neighbourhood graphs. *Distance map* (often called a correlation map or a correlation plot, depending on what criterion is used to calculate proximity) is usually a square matrix (unless only the non-redundant half of it is visualized), each cell of which displays proximity between two entities (these could be genes, other arbitrary variables, patients, etc). The proximity (or the correlation) may be calculated using e.g. Pearson's correlation, Spearman's correlation or the inverse normalized Euclidean distance. Correlation or anti-correlation is indicated by a colour scale, e.g. from blue to red. With distance maps, although details on individual feature values are lost, similarity between any pair of instances can easily be inspected.

Nowadays *heatmaps* are most actively used by the bioinformatics community, in order to visualize expression values, but they were originally used and in principle can be applied to visualize arbitrary attributes. NASDAQ, for example, uses them to show the NASDAQ 100 index volatility.

Wikipedia defines heatmaps as "a graphical representation of data where the values taken by a variable in a two-dimensional map are represented as colours". In this definition, a variable can be e.g. a gene expression value, and the 2D map can be defined as an intersection of genes and their samples. Both distance maps and heatmaps are commonly used in microarray data processing systems to visualize the results of experiments with gene expression values.

[Verhaak *et al*., 2006; Verhaak, 2006] present a visualization system called *HeatMapper* that elegantly combines these two techniques to visualize gene expression profile correlations, genotypes, phenotypes and sample characteristics.

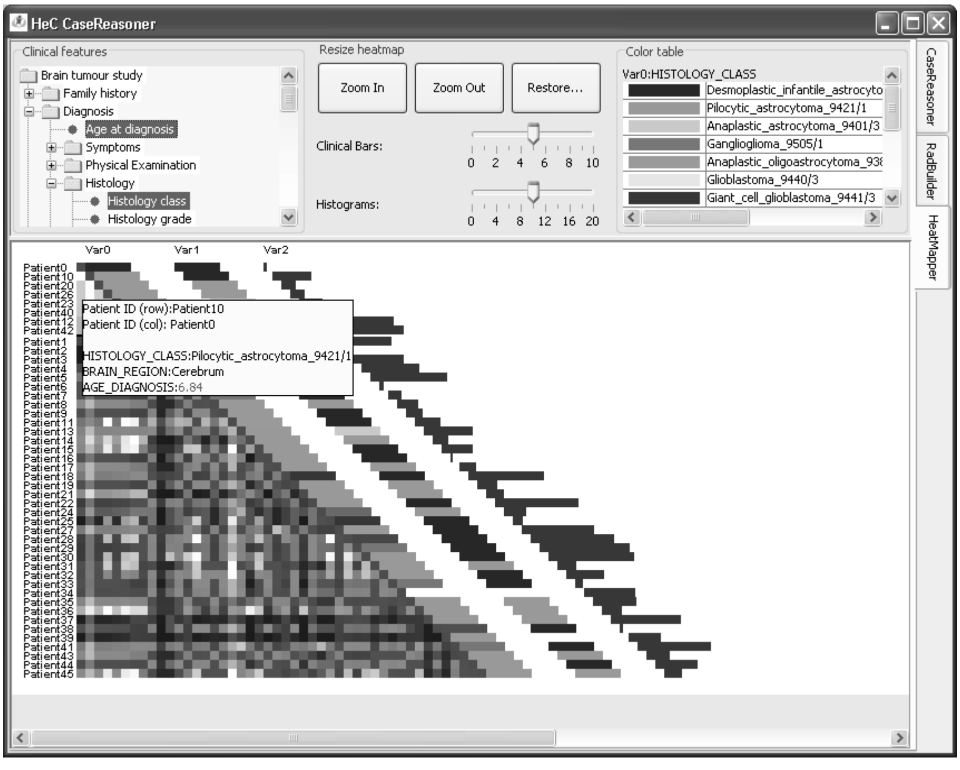
HeatMapper displays a triangular, non-redundant distance map for gene expression profile correlations, and each line representing a certain sample continues with a heatmap presenting clinical features, gene expression values and sample characteristics. Different entries in one sample characteristic are mapped to different colours, or, in the case of numeric data, shown as bars of which the size is proportional to the value. This type of visualization, combined with a clustering technique to group the samples, helps to easily spot sub-classes of samples sharing a certain commonality and has lead to many interesting discoveries, such as a mutation in particular gene and high similarity in expression profile and clinical features. [Verhaak *et al*., 2006; Verhaak, 2006]. [Verhaak, 2006] gives a thorough overview of different applications of this type of visualization and obtained results in the area of acute myeloid leukemia.

HeatMapper is an open-source Java platform-independent system, which is available free of charge with its source code, and there is no restrictions for use by non-academics.

We believe that distance maps and heatmaps may also be useful in visualizing patient similarity in medical decision support, in the context which is considered in this paper. While HeatMapper focuses on gene expression profiling and the distance map is limited to visualizing gene expression profile correlations, the similar technique could be useful in visualizing complex inter-patient similarity defined by a clinician in a more general way, which may be defined on features of different kind, including clinical, imaging and genetic features.

We consider combined distance-/heat-maps as a separate visualization technique for visualizing patient similarity in a medical DSS, which has its own limitations and benefits, and complements the previously considered treemaps and relative neighbourhood graphs.

In Figure 3, an example visualization is presented, using again the previous BrainTumour data set. We adapted the code from HeatMapper, making some minor modifications including support of the Weka *.arff* data sets, visualizing feature hierarchies instead of flat lists and explanation of colour coding used in the heatmap. Three features are displayed; histology class, affected brain region and age at diagnosis.



Var0:HISTOLOGY_CLASS
- Desmoplastic_infantile_astrocyto
- Pilocytic_astrocytoma_9421/1
- Anaplastic_astrocytoma_9401/3
- Ganglioglioma_9505/1
- Anaplastic_oligoastrocytoma_93
- Glioblastoma_9440/3
- Giant_cell_glioblastoma_9441/3

Patient ID (row):Patient10
Patient ID (col): Patient0

HISTOLOGY_CLASS:Pilocytic_astrocytoma_9421/1
BRAIN_REGION:Cerebrum
AGE_DIAGNOSIS:6.84

Figure 3 Combined distance-/heat-map visualization of the Brain Tumour data

5 Comparative Analysis

In Table 1, we summarize the three non-traditional techniques for similarity visualization and present limitations and benefits of each.

Table 1 Comparison of visualization techniques

	Benefits	Limitations
Treemaps	space efficient, more than 10.000 patients can be visualized; can be used with hierarchical clustering, ontologies and decision trees; intuitively embeds navigation through isomorphism; leads to knowledge discovery	only 3 features are simultaneously visualized (size, colour, label), plus the structure; needs some learning efforts to understand
Relative neighbourhood graphs	intuitive, easy to understand node-link form (semantic nets)	not space efficient, overcrowded for big numbers of patients; only 2 features are simultaneously visualized (colour and label); $O(N^3)$ complexity
Combined distance-/ heat-maps	full data matrix can be visualized; direct (“raw”) visualization of distance and feature values; can be used in combination with clustering; may lead to knowledge discovery	linear layout, not space efficient; understanding patterns in distances and feature values needs some efforts

The main benefits of treemaps are their space efficiency and the fact that a large number of patients may be visualized on the screen at the same time (usually much larger than with any other technique). This, and the natural support of interactivity and navigation through the tree isomorphism, may lead to the discovery of novel important knowledge; examples of this were already published [Zhang *et al.*, 2004; Kobsa, 2004]. Treemaps are somehow universal and may be used to visualize virtually any hierarchy appearing in the domain, be it a hierarchical clustering of patients, an ontology, or a simple decision tree defined on the features. While this universality has not yet been exploited in known implementations, we believe this is an important feature of treemaps, which, if used for decision support with the same data, may bring additional dividends in terms of better understanding of the problem domain.

The price to be paid for space efficiency is that treemaps are usually able to visualize only two or three features of each patient (size of the box, its colour, and sometimes label). Additionally, the structure of the tree may encode important information as well, sometimes also in the form of feature values. Besides, treemaps need some learning efforts, as this is something novel not usually encountered by those without IT background (although treemaps are becoming more widespread with each year). Treemap sceptics usually raise this as the most important argument against the use of treemaps in decision support. Recent studies, however, suggest that the learning curve might not be that bad after all, (the order of half an hour), both for students in lab experiments and medical personnel (neurologists, neurosurgeons, and clinical researchers) [Kobsa, 2004; Zhang *et al.*, 2004]. Also, the learning time is considerably implementation-dependent.

Both relative neighbourhood graphs and combined distance-/heat-maps are less space efficient than treemaps. They have, however, their own benefits. The node-link representation of neighbourhood graphs is usually considered as more intuitive and user-friendly. Combined distance/heat-maps are able to visualize virtually any feature value from the data set, together with the inter-patient distances, in an explicit form. This is something that both treemaps and neighbourhood graphs lack. Neighbourhood graphs are the weakest link with regards to this issue; they are able to visualize two features only (through label and colour of the node).

As can be seen from this comparison, these visualization techniques complement each other with their benefits and limitations. It is difficult to select and recommend only one technique for similarity visualization; rather they should be used in combination with each other.

6 CaseReasoner: A Framework for Medical CBR

Within the Health-e-Child project, one goal is to develop a prototype DSS *CaseReasoner*, based on similarity search, which will employ the considered visualization techniques.

The basic idea is to provide a clinician with a flexible tool for such operations as data filtering and similarity search, and also for the exploration of the resulting data sets. Two basic concepts distinguished in CaseReasoner are the *source* and *target* data sets. The source data set includes a sample of all patient cases relevant for a certain previously defined problem domain (such as atrial septal defect or brain glioma). In the source data set the *current patient* is distinguished, which is the patient under study, regarding whom decision support is needed. The target data set is the resulting data set after a series of filtering operations (optionally), and similarity search. In the target data set the first (the most similar) patient is distinguished. The aim is to let the clinician explore and compare the records related to the two patients, and to visualize their place in the corresponding distribution of the source and target data sets. We believe, that such functionality will be useful for the clinician for better understanding of the status of the current patient within the problem domain under study and, ultimately, for decision making.

In Figure 4 the main screen of CaseReasoner is presented. The source data set includes the 46 brain tumour patients, and the target data set includes 7 female patients similar to the patient under study in terms of features extracted from brain MRI. The problem domain is defined by a hierarchy of relevant features (represented as an .xml file with references to the original Health-e-Child database). In the case of brain glioma, it includes 124 features, including common clinical features, family history, brain MRI study, histological investigation, and expression values for relevant genes. CaseReasoner allows a clinician to find a set of similar patients in terms of an arbitrary subset of features of interest defined on the feature hierarchy. The heterogeneous Euclidean-overlap metric is currently implemented for the similarity search. A simple DICOM viewer is used to visualize related images, and histograms and bar charts from the WEKA open-source Java data mining toolkit [Witten and Frank, 2000] are used to represent the distribution of values in the source and target data sets for numeric and categorical features correspondingly.

Currently the combined distance-/heat-maps are already implemented in CaseReasoner, and the implementation of treemaps and neighbourhood graphs is considered. Each visualization supports the display of search results, specifying the current patient and the set of similar patients among the rest.

Each visualization technique is being implemented as a separate tab, which replaces the whole screen space of CaseReasoner when selected. Thus, we adopt the sequential visualization approach, in contrast to the parallel visualization [Teoh, 2007], for multiple visualization techniques. We believe that it is more important to have enough screen space, sacrificing the benefits of simultaneous visualization.

CaseReasoner is problem domain-independent and extensible, through defining new feature hierarchies representing problem domains under study. Future work with CaseReasoner includes the implementation of more sophisticated domain-specific distance functions and incorporation of models built using “eager” machine learning techniques for certain data attributes (in contrast to the “lazy” CBR approach).

7 Conclusions

Case-Based Reasoning (CBR) is a recognized and well established method for building medical systems. However, CBR has not yet become as successful in medicine as in some other application domains. One reported reason for this is the lack of transparency and explanation in decision making. We believe that one way to approach this problem is to better visualize the underlying inter-patient similarity, which is the central concept of any clinical CBR.

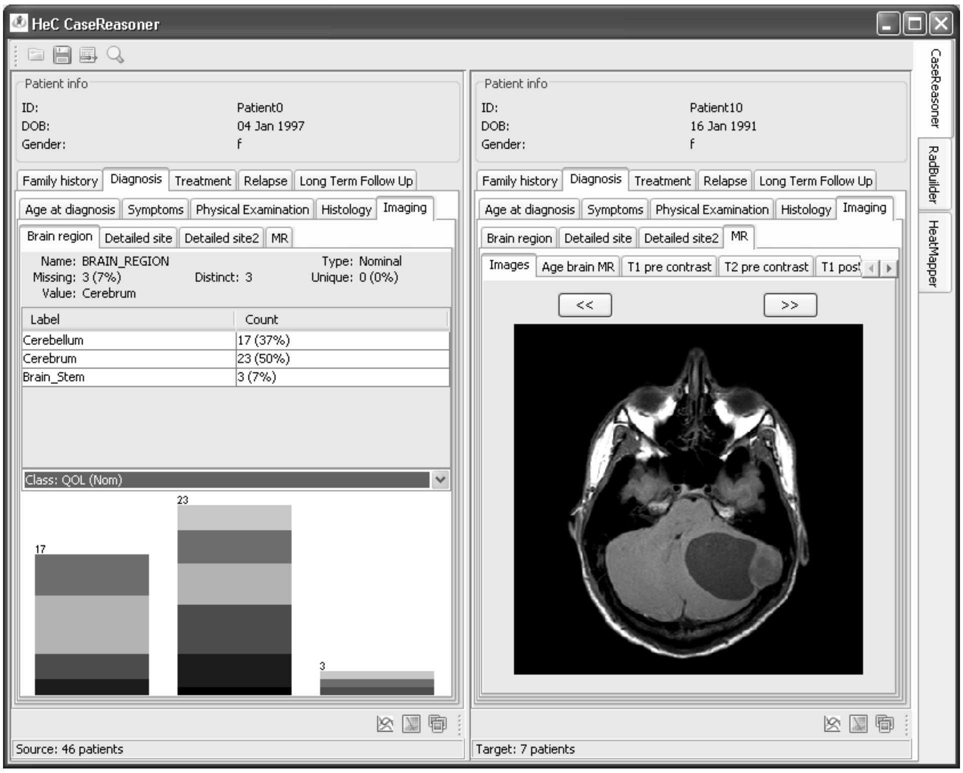
A common approach to visualise the similarity of complex multidimensional data is to use Principal Component Analysis, Independent Component Analysis, Sammon’s mapping, or any other distance-preserving dimensionality reduction space transformation technique, and to visualize the data in the extracted dimensions. One problem with this approach to be used in medical CBR is that it usually introduces a certain amount of distortion in the data, not always preserving the original distances. While it might be enough to raise certain hypotheses about the clusters in the data, it is usually not known to what extent particular local distances are preserved, which is a serious limitation for medical decision making.

In this paper we presented a comparative analysis of three techniques which do not have this limitation; treemaps, relative neighbourhood graphs, and combined distance-/heat-maps, which, we believe, hold promise in being adopted in clinical workflow. Each technique has its own benefits and limitations, which complement each other for the three techniques.

Our future work includes implementation of these techniques and evaluation in clinical workflow. Various challenges and some solutions with the task of evaluation of information visualization techniques were presented in [Plaisant, 2004].

Acknowledgements

This work has been partially funded by the EU project Health-e-Child (IST 2004-027749). The authors wish to acknowledge support provided by all the members of the Health-e-Child consortium in the preparation of this paper.



ID: Patient0
DOB: 04 Jan 1997
Gender: f

Name: BRAIN_REGION Type: Nominal
Missing: 3 (7%) Distinct: 3 Unique: 0 (0%)
Value: Cerebrum

Label	Count
Cerebellum	17 (37%)
Cerebrum	23 (50%)
Brain_Stem	3 (7%)

Class: QOL (Nom)

Source: 46 patients

ID: Patient10
DOB: 16 Jan 1991
Gender: f

Target: 7 patients

Figure 4 The main screen of CaseReasoner

References

[Baehrecke *et al.*, 2004] Eric H. Baehrecke, Niem Dang, Ketan Babaria, Ben Shneiderman. Visualization and analysis of microarray and gene ontology data with treemaps. *BMC Bioinformatics*, 5(1), 2004, 84.

[Bederson *et al.*, 2002] Benjami B. Bederson, Ben Shneiderman, Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Transactions on Graphics*, 21(4), 2002, 833-854.

[Berlin *et al.*, 2006] Amy Berlin, Marco Sorani, Ida Sim. A taxonomic description of computer-based clinical decision support systems. *J. of Biomedical Informatics*, 39(6), 2006, 656-667.

[Bruls *et al.*, 2000] Mark Bruls, Kees Huizing, Jarke J. vanWijk. Squarified treemaps. In: *Proc. Joint Eurographics and IEEE TCVG Symposium on Visualization, Eurovis'00*, Springer, 2000, 33-42.

[Garre *et al.*, 1994] M.L. Garre, S. Gandus, B. Casena, R. Haupt et al. Health status of long-term survivors after cancer in childhood. *American J. of Pediatric Hematology/Oncology, 16(2), 1994, 143-152.*

[Heer *at al.*, 2005] Jeffrey Heer, Stuart K. Card, James A. Landay. Prefuse: a toolkit for interactive information visualization. In: *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems, CHI'05*, ACM Press, 2005, 421-430.

[Jaromczyk and Toussaint, 1992] Jerzy W. Jaromczyk and Godfried T. Toussaint. Relative neighbourhood graphs and their relatives. In: *Proc. IEEE*, 80(9), 1992, 1502-1517.

[Katifori *et al.*, 2007] Akrivi Katifori, Constantin Halatsis, Georgios Lepouras, Costas Vassilakis, Eugenia G. Giannopoulou. Ontology visualization methods - a survey. *ACM Computing Surveys*, 2007 (to appear).

[Keim, 2002] Daniel A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1), 2002, 1-8.

[Kobsa, 2004] Alfred Kobsa. User experiments with tree visualization systems. In: *Proc. IEEE Symp. on Information Visualization, INFOVIS'04*, 2004, 9-16.

[Li and Hou, 2004] Ning Li and Jennifer C. Hou. Topology control in heterogeneous wireless networks: problems and solutions. In: *Proc. 23rd Annual Joint Conf. of IEEE Computer and Communications Societies, Infocom'04*, IEEE, 2004.

[Marcotegui and Beucher, 2005] Beatriz Marcotegui and Serge Beucher. Fast implementation of waterfall based on graphs. In: *Proc. 7th Int. Symp. on Mathematical Morphology, Computational Imaging and Vision*, Vol. 30, Springer, 2005, 177-186.

[McConnell *et al.*, 2002] Patrick McConnell, Kimberly F. Johnson, Simon M. Lin. Applications of Tree-Maps to hierarchical biological data. *Bioinformatics*, 18(9), 2002, 1278–1279.

[Muhlenbach and Rakotomalala, 2002] Fabrice Muhlenbach and Ricco Rakotomalala. Multivariate supervised discretization, a neighbourhood graph approach. In: *Proc. IEEE Int. Conf. on Data Mining, ICDM'02*, IEEE Computer Society, 2002, 314-321.

[Mullins and Smyth, 2001] Mark Mullins and Barry Smyth. Visualization methods in case-based reasoning. In: *Workshop Proc. 4th Int. Conf. on Case-Based Reasoning*, 2001.

[Nilsson and Sollenborn, 2004] Markus Nilsson, Mikael Sollenborn. Advancements and trends in medical case-based reasoning: an overview of systems and system development. In: *Proc. 17th Int. FLAIRS Conf. on AI, Special Track on CBR*, AAAI Press, 2004, 178-183.

[Plaisant, 2004] Catherine Plaisant. The challenge of information visualization evaluation. In: *Proc. Working Conf. on Advanced Visual Interfaces, AVI'04*, ACM Press, 2004, 109-116.

[Schmidt and Vorobieva, 2005] Rainer Schmidt and Olga Vorobieva. Adaptation and medical case-based reasoning focusing on endocrine therapy support. In: *Proc. Int. Conf. on AI in Medicine, AIME'05*, LNCS, Vol. 3581, Springer, 2005, 300-309.

[Shneiderman and Plaisant, 2004] Ben Shneiderman, Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison Wesley, 2004.

[Shneiderman, 1992] Shneiderman B. Tree visualization with Tree-Maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1), 1992, 92-99.

[Shneiderman, 2006] Ben Shneiderman. Treemaps for space-constrained visualization of hierarchies. Tech report, University of Maryland, USA, 2006. (available at http://www.cs.umd.edu/hcil/treemap-history/)

[Teoh, 2007] Soon Tee Teoh. A study on multiple views for tree visualization. In: *Proc. SPIE-IS&T Electronic Imaging, Visualization and Data Analysis, VDA'07*, Vol. 6495B, SPIE Press, 2007, 1-12.

[Toussaint, 1980] Godfried T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4), 1980, 261-268.

[Tsymbal *et al.*, 2007] Alexey Tsymbal, Sonja Zillner, Martin Huber. Ontology- supported machine learning and decision support in biomedicine. In: *Proc. 4th Workshop on Data Integration in the Life Sciences, DILS'07*, LNBI, Springer, 2007, 156-171.

[Verhaak *et al.*, 2006] Roel G.W. Verhaak, Mathijs A. Sanders, Maarten A. Bijl, Ruud Delwel, Sebastiaan Horsman, Michael J. Moorhouse, Peter J. van der Spek, Bob Löwenberg, Peter J.M. Valk. HeatMapper: powerful combined visualization of gene expression profile correlations, genotypes, phenotypes and sample characteristics. *BMC Bioinformatics*, 7, 2006, 337.

[Verhaak, 2006] Roel G. W. Verhaak. *Gene Expression Profiling of Acute Myeloid Leukemia*. PhD Thesis, Erasmus University Medical Centre, Netherlands, 2006.

[Wang *et al.*, 2006] Yue Wang, Soon Tee Teoh, Kwan-Liu Ma. Evaluating the effectiveness of tree visualization systems for knowledge discovery. In: *Proc. Eurographics/IEEE-VGTC Symposium on Visualization, Eurovis'06*, 2006, 67-74.

[Witten and Frank, 2000] Ian H. Witten, Eibe Frank. *Data Mining: Practical Machine Learning Tools With Java Implementations*. Morgan Kaufmann, 2000.

[Zhang *et al.*, 2004] Ming Zhang, Hong Zhang, Donny Tjandra, Stephen T. C. Wong. DBMap: a space-conscious data visualization and knowledge discovery framework for biomedical data warehouse. *IEEE Transactions on Information Technology in Biomedicine*, 8(3), 2004, 343-53.

TCR – Textual Coverage Rate

Kerstin Bach & Alexandre Hanft

University of Hildesheim
Institute of Computer Science
Intelligent Information Systems Lab
D-31141, Hildesheim, Germany
kerstin.bach|alexandre.hanft@uni-hildesheim.de

Abstract

In this paper we will introduce a measure of saturation for unstructured texts of unknown domains. Therefore we will present the Textual Coverage Rate (TCR), a method to determine the IE coverage of unstructured texts using a given vocabulary. We advance efficiency while building vocabulary repositories tailored for given problems and ensure a certain quality of representation. Our approach, which will be evaluated using a large case base, concentrates on the development of the TCR and will motivate its application for textual Case-Based Reasoning.

1 Introduction

In the past many documents were human readable written, but it is still challenging to capture the given information for machines and apply them to new challenges. Before techniques like Case-Based Reasoning (CBR) can be applied for knowledge management, the given data has to be analyzed and prepared. CBR helps to solve problems based on previous experiences and therefore problems and their solutions (cases) have to be analyzed. Furthermore, Textual Case-Based Reasoning (TCBR) offers the opportunity to work with free text documents as well as making previous knowledge and information available. According to [Wilson and Bradshaw, 1999] texts in CBR can be divided in two kinds: fully structured cases and fully textual cases. In this paper we will focus on the fully textual cases which can be separated in text sections of different lengths.

There are many approaches to comprehend textual documents [Asiimwe *et al.*, 2007; Massie *et al.*, 2007] which focus on the formalization and the retrieval of textual documents. In this paper we will present an approach which shows whether the given terms, vocabulary, etc. is comprehensive enough to cover a given case base. We will concentrate on preparing the case bases for TCBR and we will show how this work can be done more efficiently. Also we will show how terms can be extracted and provided to support TCBR.

Before setting up a TCBR-System, for example which is based on Case Retrieval Nets (CRN) [Lenz, 1999], one has to define which kind of data is given in the case base and how it can be accessed. As a result of this preparation a case format is defined, which will contain those information and ensure that the data can be imported and processed adequately. But knowing how to access data does not ensure that the given information can be represented in a proper way. There is a huge amount of information stored in unstructured textual documents which can hardly be processed because it is written in natural language [Quasthoff, 1997] and might contain unknown words. We will show how to minimize the number of unknown words while dealing with data of new, unknown domains and how to cope with texts in natural language. We will give an approach which can be used for small and huge case bases as well.

In the first part of the paper we will describe how a vocabulary repository of terms can be created using heterogeneous data sources. In section 3 we will introduce and explain the elements of the Textual Coverage Rate (TCR) followed of an example which illustrates its calculation, followed by an evaluation of the TCR. The paper will close up with an outline of our future work advancing the TCR.

2 Repositories and Application Data

Before we can introduce the TCR, we have to create a vocabulary repository of terms which can be used to cover texts. Since our application data will be in German we will concentrate on the German language. Also, the procedure can be done for other languages as well. Comparing the English language with the German language there are huge differences of the syntax of inflections. In German, the base form usually differs while building inflections and for that reason we cannot use a stemming algorithm. Instead we decided to build vocabularies containing terms and their inflections which can be used to build CRNs [Lenz *et al.*, 1998].

According to [Lenz, 1999] Information Entities (IE) are terms which are used to build a CRN and each text section is represented of a set of IEs. Therefore the text is divided in text sections regarding to its structure, and the terms contained in a text section which match the given IEs (e.g. of an IE vocabulary) are marked in the CRN. In comparison to full text analysis IEs are easier and faster to determine as long as an vocabulary is available. Hence, the usage of IEs also provides similarity arcs between terms which offers the search for similar terms and the extension of the query (activation of IEs in the CRNs).

2.1 Vocabulary Repository

First we will describe how to create a vocabulary integrating heterogeneous data sources and building up a repository of general terms. According to [Bach, 2007] we have used both, *GermaNet*[1] and a web service of the *Projekt Deutscher Wortschatz*[2] to collect data.

GermaNet is a lexical-semantic net, similar to *WordNet* of the Princeton University, developed at the University of

[1] http://www.sfs.uni-tuebingen.de/lsd/

[2] http://wortschatz.uni-leipzig.de/Webservices/

Tübingen [Lemnitzer and Kunze, 2002]. We have used *GermaNet* to enhance our vocabulary to be able to cover a new domain.

To use the terms and their synonyms in a given vocabulary to build up a case base the *GermaNet* entries have to be integrated in the vocabulary. The terms themselves are used to represent IEs and the semantic relations between terms can be used to assign the similarity arcs. After integrating *GermaNet* we are able to find synonyms on given terms, although *GermaNet* only provides the base forms it covers most of general language terms used in German.

The terms described in *GermaNet* contain no inflections which are important to recognize in natural language texts. Especially in the German language the inflections of a term can differ from its base form. To recognize base terms and inflections the web service provided by the *Projekt Deutscher Wortschatz* can be used, because it is the most comprehensive collection of German words. For each base form the web service returns its inflections which can be stored as terms in the repository and related to its base form.

2.2 Corpus

To evaluate the TCR we will use an application domain of insurance claims consisting of several passages of free text. Each case can be separated in several text sections of different lengths and the vocabulary used to describe the insurance claims is a mixture of general terms and specific term. Therefore we need a vocabulary which covers both and the usage of *GermaNet* gave us a huge amount of general terms so we expect the unknown words to be domain specific terms. The case base contains more than 9.500 cases with 2.2 million words.

A typical case consists of 12 attributes and 8 of them contain unstructured text. In [Bach, 2007] the given 9 attributes are described as retrieval attributes and we consider 6 of them as retrieval attributes free text. So each case we are processing will contain 9 sections and 6 of them will be text sections. How a case is structured and what kind of data it contains can be seen in Table 1, which shows one case and the IEs which match with the dictionary and represent the text section. The table only consists retrieval attributes and the TCR will be only applied for the text sections. Furthermore the column "IE" consists terms which are included in the given text section and can be found in the vocabulary. The IEs found will be used to calculate the TCR as it can be seen in 3.2.

3 Textual Coverage Rate

Most of the domain models of TCBR systems are hand written or adapted from previous applications [Minor, 2006a; Lenz *et al.*, 1998; Hanft and Minor, 2005]. Evaluating whether the existing domain model covers enough terms to represent the given text adequately is challenging. Especially when dealing with a large amount of text an automatic evaluation is needed. Therefore we will introduce a measure of the coverage a given dictionary provides to represent an unknown text. Its name is TCR, Textual Coverage Rate, and it aims at ensuring a higher quality of unstructured text representation.

In [Bach and Hanft, 2007] an approach has been introduced where text sections are analyzed to figure out which words are not described while using a given dictionary representing an unknown text. In the first step all stop words from a given corpus are eliminated because they have no useful information content. As a second step all words which are contained in the available vocabulary are removed and as a result we get a list of words the system does not know is displayed. The knowledge engineer has to model those unknown words to assure a satisfying representation of the text.

section name	content	IEs
Id	1612	
Ursache	Verdacht auf Flüssigkeitsschaden.VU bittet um Info über Schadensursache und vielleicht über Reparaturmöglichkeiten	Verdacht, Info, Schadensursache, Reparaturmöglichkeiten
Bemerkung	Gerät kann beim Anspruchsteller besichtigt werden.	Gerät
Kurzbeschreibung	HIFI-Stereoanlage	Stereoanlage
Anschaffungswert	200	200
Zeitwert	50	50
Objekt	HIFI- Verstärker Pioneer, Model A- 204R	Verstärker, Model
Zustand	gebraucht, leichte Kratzer am Verstärkergehäuse. An der Front fehlen vier Einstell-Drehknöpfe.	gebraucht, Kratzer, Front, fehlen
Gerätealter	9	9
Schäden	Der Verstärker weist keine spannungsspezifische Funktion auf. Auf der Haupt- und Endstufenplatine sind Flüssigkeitsspuren zu finden. In diesem Bereich sind die Leiterbahnen und Bauteile wie z.B. Widerstände, Kondensatoren und IC-Kontaktbeine korrodiert.	Verstärker, keine, Funktion, Haupt, finden, Bereich, Bauteile, Widerstände

Table 1: An exemplary case (retrieval attributes) of the databased used to evaluate the TCR.

Handling large databases the amount of unknown words increases, for this purpose it is important to know which words have to be modeled in first place. In the approach mentioned above lists ordered by frequency have been applied. The impact of modeling words with a high frequency in the source data in the beginning is comprehensive. Like it is shown in Figure 1 the number of unknown words decreased constantly and after looking over the result it has been noticed that words which would help to cover the unknown text were not always been modeled.

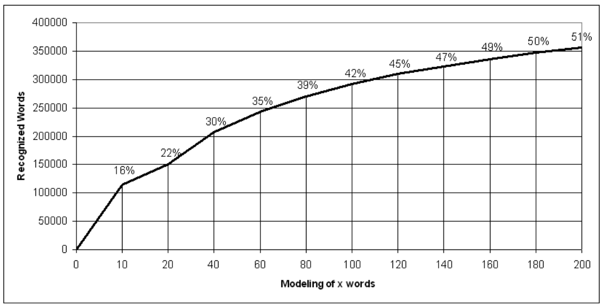


Figure 1: Impact on modeling the 200 most frequent unknown words

3.1 Motivation

Instead of using the frequency of occurrences to prioritize words which have to be modeled first, we will introduce an approach which regards each text section and assigns its coverage with IEs. We aim to indicate words which have to be modeled to ensure each text section is represented satisfactorily. Therefore we focus on both, the expected number of IEs in a given section and the frequency of unknown words.

Preparing an unknown corpus for TCBR requires an analysis if the given dictionary holds suitable terms. We will introduce the Textual Coverage Rate (*TCR*) to describe the potential representation of the source text using the existing dictionary. Therefore we measure the IE coverage of each text section to determine whether it contains a minimum number of terms given in the dictionary or not.

3.2 Calculation

Following [Minor, 2006a] a case c with k text sections can be described as $c = [S^1, S^2, \ldots, S^k]$. Each text section is represented by a set of IEs, called S^i. In addition, we will use T^i which describes the expected number of IEs in each considered text section. In comparison to [Bach and Hanft, 2007] we do not have a global T, because we figured out, that T should depend on the local attribute.

(1) and (2) calculate the number of text sections which contain less IEs than given by T^i.D_{cov} describes the coverage rate of one text section. It is 0 if there are less than T IEs in the tested section S^i. For example a text section is represented by two IEs ($|S^i| = 2$) and three IEs are expected ($T = 3$) this section is less covered and D_{cov} for the considered section will be 0.

$$D_{cov}(S^i, T^i) = \begin{cases} 0 & , \quad |S^i| < T^i \\ 1 & , \quad \text{else.} \end{cases} \tag{1}$$

To calculate the *TCR* the number of appropriate covered text sections has to be summed up and the ratio between this sum and the total number of sections gives the *TCR*:

$$TCR(c, T^i) = \frac{\sum\limits_{i=1}^{k} D_{cov}(S^i, T^i)}{k}. \tag{2}$$

The *TCR* shown above describes the percentage of text sections represented by at least T IEs. If every text section is adequately covered (for each text section $|S^i| \geq T$ is true) the *TCR* will be 1. Otherwise the knowledge engineer should model more terms to increase the coverage of the given dictionary. To figure out which words should be added to the dictionary the approach described in [Bach and Hanft, 2007] can be used.

Furthermore, if the *TCR* is 1, the percentage of text sections which contain more than T IEs should be calculated. For that reason the ratio of excess coverage can be examined as shown in (3) and (4). In opposite to (1) and (2) only text sections represented by more than T IEs are factored.

$$D_{excess}(S^i, T) = \begin{cases} 1 & , \quad |S^i| > T \\ 0 & , \quad \text{else.} \end{cases} \tag{3}$$

$$C_{excess}(c, T) = \frac{\sum\limits_{i=1}^{k} D_{excess}(S^i, T)}{k}. \tag{4}$$

A high excess coverage ratio C_{excess} (more than 0.8) points out that more than the expected T IEs represent a text section and the knowledge engineer can consider increasing T. After increasing T the *TCR* has to be updated and the recalculated C_{excess} helps to decide whether T is chosen correct or still too low. If necessary this step has to be repeated until a C_{excess} of 0.5 or less occurs.

The *TCR* can be used to explain to the knowledge engineer how many words have to be modeled to achieve a certain quality (given by T) covering the corpus. In addition the C_{excess} can increase the quality of coverage, because it shows how many words have to be modeled to increase T.

3.3 Calculation Example

According to the given data in Table 1, Table 2 shows in detail how the TCR is calculated. S^i counts the given sections which will be referenced during the calculation. The column $|S^i|$ contains the number of IEs found in the text section and according to section 3.2 D_{cov} is calculated. T is chosen according to the average number of IEs found in each section. k contains the total number of text sections and the TCR in the given example is 0.5. To consider increasing T, C_{excess} is calculated based on the given data. The result 0.3 shows, that most of the considered text sections to calculate C_{excess} are represented of T IEs.

	S^i	$\lvert S^i \rvert$	T^i	D_{cov}	D_{excess}
Id					
Ursache	1	4	1	1	1
Bemerkung	2	1	1	1	0
Kurzbeschreibung	4	1	2	0	0
Anschaffungswert					
Zeitwert					
Objekt	8	2	7	0	0
Zustand	9	4	1	1	1
Gerätealter					
Schäden	11	8	10	0	0
Σ				3	
	$k = 6$		$TCR = 0.5$		$C_{excess} = 0.3$

Table 2: Calculation of the TCR based on the exemplary case of Table 1.

As a result the solutions show that the knowledge engineer should first model terms contained in sections which were rated 0 while calculating D_{cov}. Hence, T was chosen adequate for each attribute to represent the given text sections. As described before we chose T according to the average number of IEs in each section.

4 Evaluation

After introducing the TCR now we will present the evaluation using the previously described corpus (see section 2.2). For the evaluation we use a corpus of 9640 cases and each case looks like the example given in Table 1. In a first step we calculated the average number of IEs occurring in each section. The results are given in in Table 3.

	S^i	min	max	avg	T^i
Ursache	1	0	20	1.16	1
Bemerkung	2	0	20	0.71	1
Kurzbeschreibung	4	0	12	2.20	2
Objekt	8	0	53	7.18	7
Zustand	9	0	32	1.50	1
Schäden	11	0	71	9.98	10

Table 3: Summary of IE distribution

Table 3 shows that depending on the text section the number of IEs found can vary. To assign an appropriate T

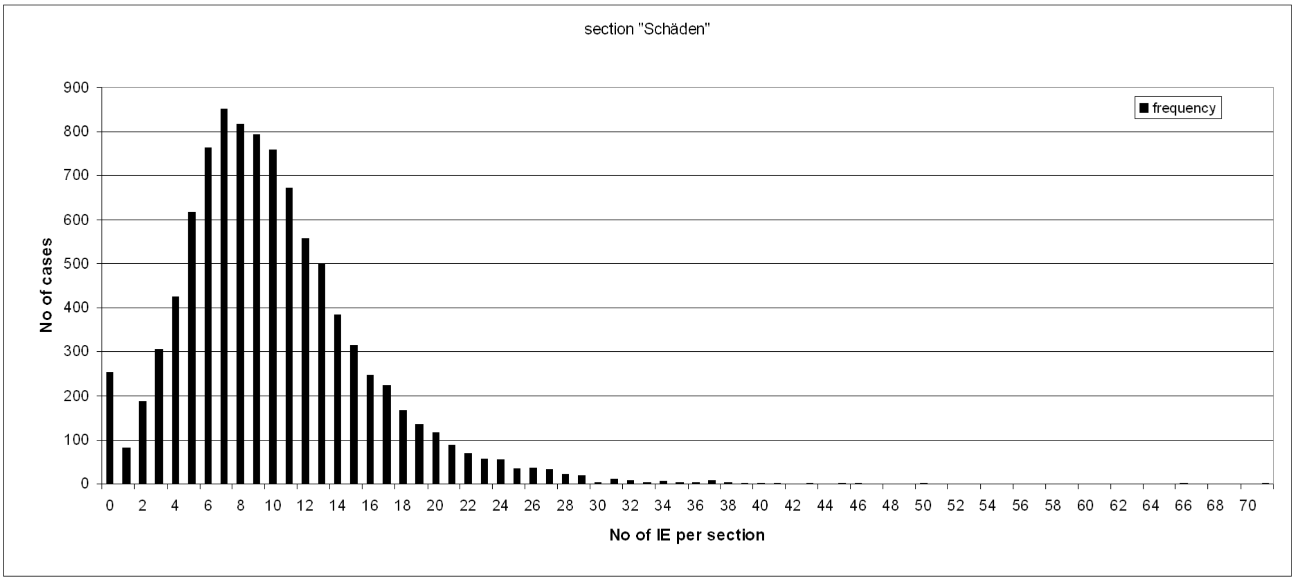


Figure 2: Histogram of found IEs per section

we first choose the average number of IEs per section which can later be refined. As an advancement in comparison to [Bach and Hanft, 2007] we do not have a global T, because during evaluation we figured out, that the variance can be large. For example in text section 4, "Kurzbeschreibung", and text section 11, "Schäden", we found a minimum of 0 IEs. "Kurzbeschreibung" has a maximum of 12 IEs unlike "Schäden" contains a maximum 71 IEs. This difference would make it hard to assign a global TCR while we are aiming for an adequate representation of each text section.

In a first step we recommend to choose the average number of IEs to figure out how the text can be represented. Depending on the given data the T might have to be increased. For the actual case base we get the following results (which can be seen in Figure 2) for the attribute "Schäden" after choosing the average number of IEs per text section.

Figure 2 shows the frequency of the IEs found in each text section of "Schäden". Although the average number of IEs is 10, more than 50% of the text sections contain less IEs which can be seen as positive skew in Figure 2. The skew for the distribution of the section "Schäden" is 1.30. We computed the frequency distribution of the other five sections as well and all of them show a similar positive skew.

Obviously the much higher the value is chosen for T, the number of IEs which have to be modeled increase, because the TCR of more cases is insufficient. Now the knowledge engineer has to decide whether the given coverage is exhausted or if a modeling is necessary. In our example, for text section "Schäden", it is required to remodel terms, because the text section contains a lot of information and the retrieval is going to need this data to work properly.

Changing T influences the saturation of the text sections of IEs, which assures a higher probability of retrieving correct documents. But aiming at a high T means that many terms have to be modeled which might be time-consuming. Using the TCR can tell the knowledge engineer how many terms have to be modeled and he will be able to decide if it is worth it. After evaluating our data using different numbers of cases, we suggest to take the median as value for T.

TCR	No of cases
0	8
0.167	272
0.333	1676
0.5	3339
0.667	2938
0.833	1258
1	149

Table 4: Number of cases with the same TCR

For our test case base, we calculate the TCR for each case and get an average of 0.563. Having a closer look at the distribution and summing up the cases with the same TCR, which can be seen in 4, one third of the cases have a TCR of 0.5. About 45% have a TCR of 0.6 or higher, but although 8 cases have no sufficiently filled sections as well as 1956 cases have more than an half insufficient text sections.

In our example more modeling is required because the considered text section contains a lot of information and the retrieval is going to need this data to work properly.

5 Conclusion & Outlook

This paper devotes furthering the performance of TCBR applications concentrating on the preparation of data and case base. We described the usage of *GermaNET* and *Projekt Deutscher Wortschatz* to create a vocabulary of terms which will be used represent unknown texts. Creating vocabularies from scratch is challenging, so we based our vocabulary on the ExperienceBook II[3] vocabulary, which was developed at the Humboldt University of Berlin [Hanft and Minor, 2005; Minor, 2006a; 2006b]. Nevertheless, the presented approach using *GermaNET* and *Projekt Deutscher Wortschatz* can be used to build a new vocabulary as well. Furthermore, the English complement of *GermaNET*, *WordNET*[4] is also available and can be used just like the German version.

[3] https://roy.informatik.hu-berlin.de/ExpBookII/

[4] http://wordnet.princeton.edu/

The Textual Coverage Rate (TCR) presented in this paper measures the coverage of IE of unstructured text section using a given vocabulary and facilitate a deep insight in the considered text corpus. This empowers the knowledge engineer to decide which parts of the corpus should be modeled first and how much should be done to achieve a certain quality of modeling which means coverage of unstructured text through IEs. The TCR was evaluated using a corpus with over 9500 cases.

Similar work is done at Robert Gordon University. [Massie *et al.*, 2007] describes the extraction features from text in anomaly reports to map them to structured cases. As in our approach, [Wiratunga *et al.*, 2005] also motivate the pre-processing of data to extract features, but this work uses rules to extract features, like the Propositional Semantic Indexing (PSI). In comparison to our approach, PSI depends on the domain and relies on word-class cooccurrences. Parts of the calculation of TCR are like the vector space model [Salton *et al.*, 1975], especially the term frequency, but TCR avoids the problem of small similarity values by long documents and, as usually in CBR, the requirement of an exact matching of the key words in query and case is not necessary.

In future work we will use TCR to facilitate automatic maintenance of knowledge in a Case Factory, which has been described in [Althoff *et al.*, 2006]. Furthermore we will concentrate on developing the TCR aiming at its application in systems based on CoMES [Althoff *et al.*, 2007], because those applications will for example process contributions on community platforms.

Currently, our vocabulary repository contains terms and their synonyms, but we are aiming at enhancing the vocabulary with similarities between terms, so CRNs can demonstrate their strength dealing with fully textual cases.

Another challenge for the future will be coping with homonyms which can possibly be done using a semantic classification of terms to figure out the meaning of a certain term. Therefore the classes of the terms in the text section have to be considered and according to those we might be able to assign the homonyms' meaning.

References

[Althoff *et al.*, 2006] Klaus-Dieter Althoff, Alexandre Hanft, and Martin Schaaf. Case Factory – Maintaining Experience to Learn. In Mehmet H. Göker, Thomas Roth-Berghofer, and H. Altay Güvenir, editors, *Proc. 8th European Conference on Case-Based Reasoning (ECCBR'06), Ölüdeniz/Fethiye, Turkey*, volume 4106 of *Lecture Notes in Computer Science*, pages 429–442, Berlin, 2006. Springer Verlag.

[Althoff *et al.*, 2007] Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Régis Newo, Meike Reichle, Martin Schaaf, and Karl-Heinz Weis. Collaborative Multi-Expert-Systems – Realizing Knowlegde-Product-Lines with Case Factories and Distributed Learning Systems. In Joachim Baumeister and Dietmar Seipel, editors, *Accepted for the Workshop Proceedings on the 3rd Workshop on Knowledge Engineering and Software Engineering (KESE 2007)*, Osnabrück, Germany, 2007.

[Asiimwe *et al.*, 2007] Stella Asiimwe, Susan Craw, Bruce Taylor, and Nirmalie Wiratunga. Case Authoring: from Textual Reports to Knowledge-rich Cases. In Michel Richter an Rosina Weber, editor, *ICCBR07*, volume 4626 of *Lecture Notes in Artificial Intelligence*, pages 179–193, Berlin, 2007. Springer.

[Bach and Hanft, 2007] Kerstin Bach and Alexandre Hanft. Domain Modeling in TCBR Systems: How to Understand a New Application Domain. In David C. Wilson and Deepak Khemani, editors, *Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR) 2007, Workshop on Knowledge Discovery and Similarity*, pages 95–103, Belfast, Northern Ireland, 2007.

[Bach, 2007] Kerstin Bach. Domänenmodellierung im Textuellen Fallbasierten Schließen. Master's thesis, Institute of Computer Science, University of Hildesheim, 2007.

[Hanft and Minor, 2005] Alexandre Hanft and Mirjam Minor. A Low-Effort, Collaborative Maintenance Model for Textual CBR. In Stefanie Brüninghaus, editor, *ICCBR 2005 Workshop Proceedings*, pages 138–149, 2005.

[Lemnitzer and Kunze, 2002] Lothar Lemnitzer and Claudia Kunze. GermaNet - representation, visualization, application. In Rodriguez, M.G., and C.P.S. Araujo, editors, *Proceedings Conference on Language Resources and Evaluation (LREC) 2002, main conference, Vol V.*, pages 1485–1491, 2002.

[Lenz *et al.*, 1998] Mario Lenz, André Hübner, and Mirjam Kunze. Textual CBR. In Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard, and Stefan Wess, editors, *Case-Based Reasoning Technology – From Foundations to Applications*, Lecture Notes in Artificial Intelligence, LNAI 1400, pages 115–137. Springer-Verlag, Berlin, 1998.

[Lenz, 1999] Mario Lenz. *Case Retrieval Nets as a Model for Building Flexible Information Systems*. Dissertation, Humboldt University of Berlin, Berlin, 1999.

[Massie *et al.*, 2007] Stewart Massie, Nirmalie Wiratunga, Alessandro Donati, and Emmanuel Vicari. From Anomaly Reports to Cases. In Michel Richter and Rosina Weber, editors, *ICCBR07*, Lecture Notes in Artificial Intelligence, pages 359–373. Springer, 2007.

[Minor, 2006a] Mirjam Minor. *Erfahrungsmanagement mit fallbasierten Assistenzsystemen*. Dissertation, Humboldt University of Berlin, May 2006.

[Minor, 2006b] Mirjam Minor. Experience Management with Case-Based Assistant Systems. In Thomas Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir, editors, *ECCBR 2006*, volume 4106 of *Lecture Notes in Artificial Intelligence*, pages 182–195, Berlin, 2006. Springer.

[Quasthoff, 1997] Uwe Quasthoff. Projekt Der Deutsche Wortschatz. In Gerhard Heyer and Christian Wolff, editors, *GLDV-Jahrestagung*, pages 93–99. Deutscher Universitäts-Verlag, 1997.

[Salton *et al.*, 1975] Gerard Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Communication of the ACM*, 18(11):613–620, November 1975.

[Wilson and Bradshaw, 1999] David C. Wilson and Shannon Bradshaw. CBR Textuality. In Stefanie Brüninghaus, editor, *Proceedings of the Fourth UK Case-Based Reasoning Workshop*, pages 67–80, University of Salford, 1999.

[Wiratunga *et al.*, 2005] Nirmalie Wiratunga, Robert Lothian, Sutanu Chakraborti, and Ivan Koychev. A Propositional Approach to Textual Case Indexing. In Alipio Jorge, Luis Torgo, Pavel Brazdil, Rui Camacho, and Joao Gama, editors, *PKDD*, volume 3721 of *Lecture Notes in Computer Science*, pages 380–391. Springer, 2005.

panta rhei[*]

Christine Müller, Michael Kohlhase
Computer Science, Jacobs University, Bremen, Germany
c.mueller/m.kohlhase@jacobs-university.de

Abstract

This paper introduces panta rhei, an *interactive and collaborative community reader*, which integrates narrative structures into arbitrary materials while at the same time facilitating readers to challenge, discuss, and rate the presented information. Three case study are provided to describe the functionality of the system, each of them illustrating different challenges. In particular, this paper focuses on the educational release of panta rhei, which facilitates the annotation and rating of educational content.

1 Introduction

panta rhei — *everything flows*. These two words where used by Plato to summarize the study of his colleague Heraklit, who argued that the world is a permanently *becoming and vanishing* one, in which everything constantly changes. This also applies to the area of Web2.0, which revolutionized the WWW and transformed it into a more social, user friendly, emergent, and flexible network, in which users have become media producers and web applications became more open and as social, while at the same time improving their mutual integration.

For example, thanks to their high usability in terms of content creation, Web2.0 applications, such as wikis, web annotation tools, and social bookmarking software, have acquired a mass of user-specific information that users can share among each other. However, in order to e.g. find, explore, and track information, users have to be able to filter and structure web content. The Web2.0 technique of sharing and tagging bookmarks offers a user-friendly interface for this. It allows to match bookmarks according to their topics and to improve searching. However, the user still has to dig through the tag clouds and to filter relevant and useful information.

Essentially, users lack the coherent, consistent, and well-researched structure that conventional media like books, courses, or films provide. These put the bare facts into a *narrative* context that guides the reader, facilitates understanding, and avoids information duplication. This concern has been addressed in the educational knowledge repository CONNEXIONS [CNX07], where learners can draw on the course structures of a vast educational corpus in order to more easily find and explore the material.

Analogously, the eLearning application ACTIVEMATH [MAF$^+$01] automatically generates user-adapted narrative structures. Nevertheless, neither of the approaches implements the knowledge acquisition as an collaborative and social process: CONNEXIONS supports authors to collaboratively create content and learners to explore and find content, but does not offer an interactive discussion or feedback loop for its users. In contrast, ACTIVEMATH at least asks students to fill out questionnaires about an exercise and offers to post comments on lecture notes, which are eventually considered by the authors of the courses. However, both systems only have rudimentary means to discuss course related problems and to give feedback on the lecture. Thus, students have to read and interpret the material without interacting with other students or the author. Moreover, authors do not receive feedback on whether their materials are understandable or useful.

In this paper we introduce panta rhei, an *interactive and collaborative community reader*, which integrates narrative structures into Web2.0 materials while at the same time facilitating readers to challenge, discuss, and rate the presented information. A demonstrator is available at [Mü07]. panta rhei provides an elaborated annotation of the content, allowing users to pose questions on-the-fly. By managing the users' annotation in a threaded structure, users can then commonly discuss and answer questions. Furthermore, panta rhei offers an rate view on the content which allows users to e.g. rate the relevance or helpfulness of other users' entries. Based on technologies, such as social ratings and virtual communities, panta rhei furthermore offers personalized rankings of user entries and adapted views on the content. In consequence, panta rhei supports the social and emergent nature of collaborative knowledge acquisition and exchange but does also offer narrative structures to most easily explore and find information.

[*] This work was funded by JEM-Thematic-Network ECP-038208. We would further like to thank the members of the KWARC group and the DFKI Bremen for the fruitful discussions and continuous feedback on our work.

2 `panta rhei` — Workflow

The `panta rhei` project aims at providing a general framework that combines arbitrary presentation hypermedia, its narrative structure, and various discussion platforms. The narrative structure is used to select the respective content from the presentation component, while the discussion items link to the presented content.

Figure 1 shows the general architecture of the system consisting of three main components: the *import*, `panta rhei`, and `my panta rhei`. The *import component* transforms the author's inputs, i.e. the content and narrative structure of the respective documents for `panta rhei`. `my panta rhei` includes the user-specific data such as the reader's interaction, ratings, preference settings, and his community membership. `panta rhei` includes the user interface, which presents a personalized view on content, structure, and the forum based on information such as the user's interactions as well as user and community specific ratings and preferences. Furthermore, it includes the UI adapter responsible for the adaptation of the interface as well as the community interpreter, which will be responsible for the identification of communities of practice (CoPs) [Wen05] as well as the classification of readers as community members. Please note that the community features are not part of `panta rhei v1`.

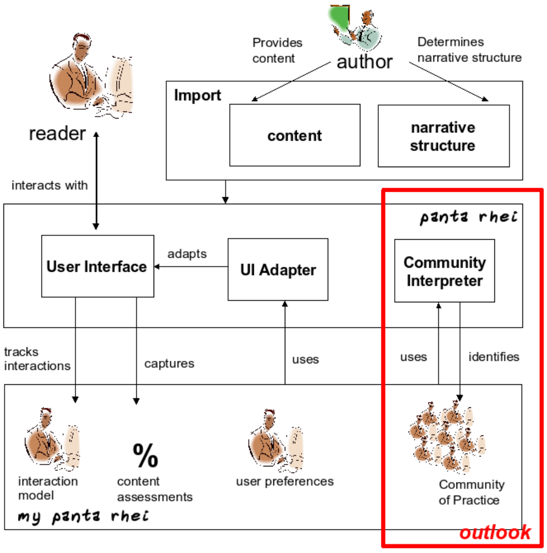


Figure 1: The `panta rhei` architecture

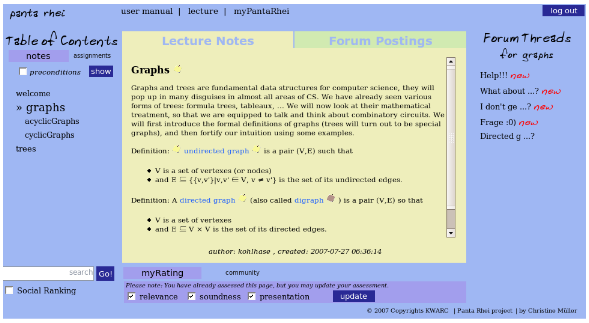
2.1 Three Case Studies

We will use three case studies in order to guide our intuition: An educational case study based on our General Computer Science lecture (GenCS), an academic case study based on the MKM conference, and finally a case study used to enable the collaborative specification of the semantic mark-up language OMDoc [Koh06].

In preparation of these case studies, we have opened a collaborative discussion and rating space of the `panta rhei` system itself, referred to as *panta agora* (cf. [Mü07]). In the time of Socrates and Plato, the *agora* was the market-place and center of public life. The Greek word "agora" comes from the verb "ageirein" meaning "to gather" and designated initially to the assembly of the whole people. The Athens agora supposedly was the place Plato spent most of his time discussing philosophy and answering questions. Gathering information in an collaborative environment which supports a bottom-up rather than a top-down communication very much relates to the intentions of social software and community tools such as `panta rhei`. According to the initial meaning of 'agora', the *panta agora* is a central space for all users to discuss the `panta rhei` system itself (centered around the documentation of the `panta rhei` system) allowing users to comment on the system, pose question, report bugs, as well as enter the suggestion for improvements and extension of `panta rhei`.

2.2 GenCS scenario

In its first version, `panta rhei` is based on the educational scenario, in which the system combines marked-up course material, the narrative course structure, and a course forum for the lecture General Computer Science (GenCS) at Jacobs University. Currently, the GenCS lecture is accompanied by a course forum that is used to distribute the course notes, homeworks, and announcement. Students use it to discuss the GenCS lecture, to ask questions, and to solve their homework. They are supported by our Teaching Assistants (TAs) who contribute to the discussions. It has been a limiting factor to the discussions that one cannot explicitly point to the section in the lecture or homework since these are only available as attachments in PDF format. In consequence, students have to describe their problem in the forum hoping that the TAs and other students will find the referred slides in the PDF and provide appropriate answers. Moreover, in order to reply to all upcoming questions and to assure that their answers did help the students, TAs have to frequently check the forum postings.



Graphs

Graphs and trees are fundamental data structures for computer science, they will pop up in many disguises in almost all areas of CS. We have already seen various forms of trees: formula trees, tableaux, ... We will now look at their mathematical treatment, so that we are equipped to talk and think about combinatory circuits. We will first introduce the formal definitions of graphs (trees will turn out to be special graphs), and then fortify our intuition using some examples.

Definition: undirected graph is a pair (V,E) such that

- V is a set of vertexes (or nodes)
- and E ⊆ {{v,v'}|v,v' ∈ V, v ≠ v'} is the set of its undirected edges.

Definition: A directed graph (also called digraph) is a pair (V,E) so that

- V is a set of vertexes
- and E ⊆ V × V is the set of its directed edges.

author: kohlhase , created: 2007-07-27 06:36:14

Help!!! new
What about ...? new
I don't ge ...? new
Frage :0) new
Directed g ...?

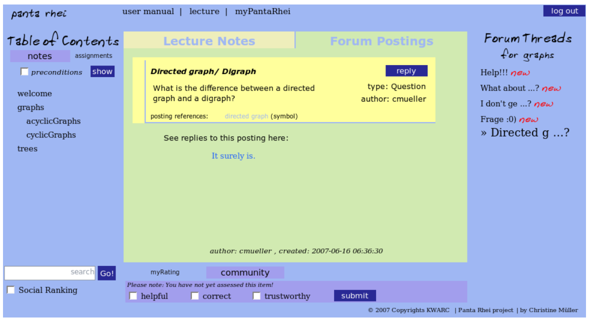
Please note: You have already assessed this page, but you may update your assessment.

☑ relevance ☑ soundness ☑ presentation

© 2007 Copyrights KWARC | Panta Rhei project | by Christine Müller

Figure 2: The `panta rhei` content

Figure 2 and Figure 3 present two screen-shots of the system: The *table of content* on the left displays the sections and subsection of GenCS, i.e. the narrative structure of the course and the list of assignments. Depending on the selected tab, the main section in the center of `panta rhei` displays the hyper-media presentation, i.e. a *course* page or assignment (Figure 2), or the discussion media, i.e. a *forum* posting (Figure 3) used to discuss the lecture and homework. The section on the right-hand side displays a list of *forum threads*



panta rhei user manual | lecture | myPantaRhei log out

Table of Contents
notes assignments
□ preconditions show
welcome
graphs
acyclicGraphs
cyclicGraphs
trees

Lecture Notes Forum Postings

Directed graph/ Digraph reply
What is the difference between a directed graph and a digraph?
type: Question
author: cmueller
posting references: directed graph (symbol)

See replies to this posting here:
It surely is.

author: cmueller , created: 2007-06-16 06:36:30

Forum Threads for graphs
Help!!! new
What about ...? new
I don't ge ...? new
Frage :0) new
» Directed g ...?

search Go!
□ Social Ranking

myRating community
Please note: You have not yet assessed this item!
□ helpful □ correct □ trustworthy submit

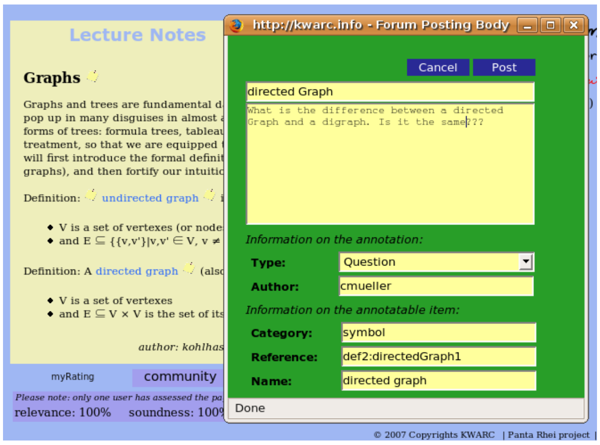
© 2007 Copyrights KWARC | Panta Rhei project | by Christine Müller

Figure 3: The `panta rhei` forum

that are linked to the course page and which `panta rhei` can rank and filter. Moreover, users can rate course pages and forum postings by using the rating form below the main section.

Features `panta rhei` supports students' activities, e.g. the *browsing of course material*, the *annotation* of course material or forum postings as well as the *search for postings and course material*, and thus facilitates the discussion of course material between students and TAs. Additionally, the system allows to *rate* of forum postings (helpful, correct, trustworthy) and course pages (relevance, soundness, and presentation). In [KK06], these ratings or *value judgments* have been introduced as essential property in order to model a community of practice.

Annotation of course materials To post a question or comment referencing an information item (e.g. a concept, its definition, or a claim), students can to click on a specific locator button in the form of a *post-it*. This displays a pop-up, given the author, the type of annotation, as well as further meta-information about the annotated item on the page. We call an information item that can be annotated in this way an **annotation item** (ANIT). After submitting the annotation, `panta rhei` creates a respective forum posting that is linked to respective ANIT. In Figure 4 the student decided to pose a question about the *concept* of a *directed graph* on the *graphs* course page. When posting his question, a posting is created, (see section 3 for details about the link `graphs#def2:directedGraph1`).



Lecture Notes

Graphs

Graphs and trees are fundamental d
pop up in many disguises in almost a
forms of trees: formula trees, tableau
treatment, so that we are equipped t
will first introduce the formal definit
graphs), and then fortify our intuitio

Definition: undirected graph i

- V is a set of vertexes (or node
- and E ⊆ {{v,v'}|v,v' ∈ V, v ≠

Definition: A directed graph (also

- V is a set of vertexes
- and E ⊆ V × V is the set of its

author: kohlhas

myRating community
Please note: only one user has assessed the pa
relevance: 100% soundness: 100

http://kwarc.info - Forum Posting Body

Cancel Post

directed Graph

```
What is the difference between a directed
Graph and a digraph. Is it the same???
```

Information on the annotation:

Type: Question
Author: cmueller

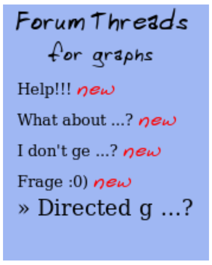
Information on the annotatable item:

Category: symbol
Reference: def2:directedGraph1
Name: directed graph

Done

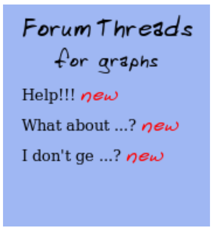
© 2007 Copyrights KWARC | Panta Rhei project |

Figure 4: Posting a question.



Forum Threads for graphs
Help!!! new
What about ...? new
I don't ge ...? new
Frage :0) new
» Directed g ...?

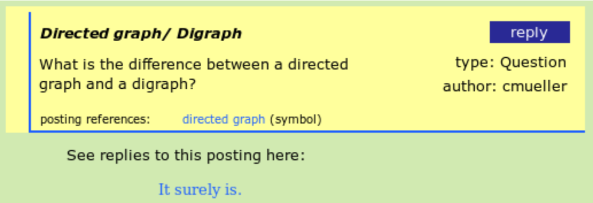
Searching for questions in the content Students can choose to solely view all postings that refer to a page or its sections. This is currently implemented by links on the page, i.e. by selecting a respective link on the page, `panta rhei` returns a list of threads that are linked to the respective section ranked by their creation timestamp (cf. figure to the left).



Forum Threads for graphs
Help!!! new
What about ...? new
I don't ge ...? new

Moreover, user's can set the display preferences of the lists. For now, user can choose to highlight all postings before a certain timestamp (i.e. 'yesterday', '3 days ago', 'a week ago', 'semester start') or to hide all postings after the timestamp. For example in the figure to the right, the list of posting has been filter: Only postings posted within the last three days are displayed. All postings that have been created since the user's last login are additionally marked as new.

Browsing the forum After selecting one of the *forum threads* in the list of postings linked to the current ANIT, the tab of the center switches to the *forum* view (cf. Figure 3). Students can now reply to the posting or browse the posting thread by clicking on the replies below the posting. Figure 5 displays a question about *directed graphs* and a link to an answer to the posting.

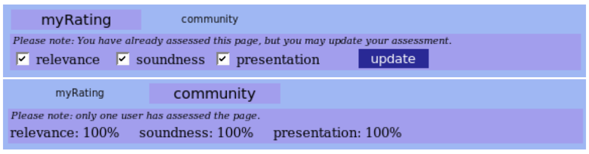


Directed graph/ Digraph reply
What is the difference between a directed graph and a digraph?
type: Question
author: cmueller
posting references: directed graph (symbol)

See replies to this posting here:
It surely is.

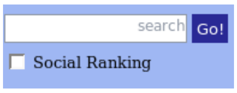
Figure 5: Viewing the directed graphs posting.

Rating content and forum postings The figure below displays a rating form for the course content allowing to rate the relevance, soundness, and presentation. In addition, users may view an community rating, i.e. an average rating of all users. Forum postings are rated in three different categories: helpful, correct, and trustworthy.



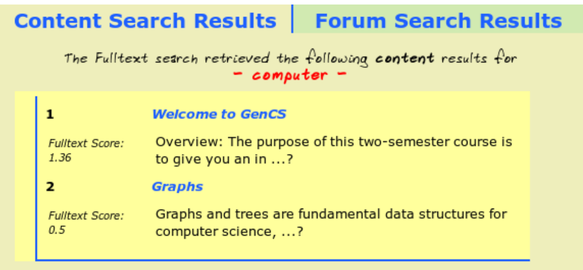
myRating community
Please note: You have already assessed this page, but you may update your assessment.
☑ relevance ☑ soundness ☑ presentation update

myRating community
Please note: only one user has assessed the page.
relevance: 100% soundness: 100% presentation: 100%



search Go!
□ Social Ranking

Searching content and forum `panta rhei` implements a simple Full text search on content and postings. Moreover, the system provides an optional social ranking of search results based on the overall rating of pages and forums. The figure above displays the results for the full text search on the course content for the query string *computer*: The similarity measure of the search is displayed on the left of the results and leads to the ordering of results.

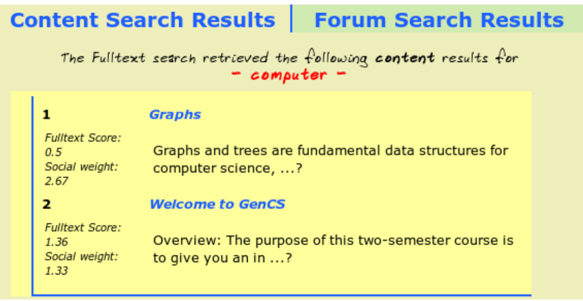

Content Search Results | Forum Search Results

The Fulltext search retrieved the following content results for
- computer -

1 Welcome to GenCS
Fulltext Score: 1.36
Overview: The purpose of this two-semester course is to give you an in ...?

2 Graphs
Fulltext Score: 0.5
Graphs and trees are fundamental data structures for computer science, ...?

However, when ranking the results based on the community rating as below, the page 'graphs' has a higher rank and is thus displayed first.

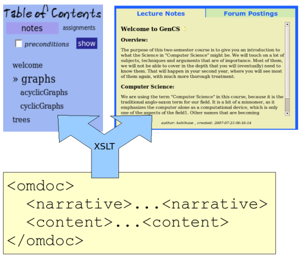

Content Search Results | Forum Search Results

The Fulltext search retrieved the following content results for
- computer -

1 Graphs
Fulltext Score: 0.5
Social weight: 2.67
Graphs and trees are fundamental data structures for computer science, ...?

2 Welcome to GenCS
Fulltext Score: 1.36
Social weight: 1.33
Overview: The purpose of this two-semester course is to give you an in ...?

2.3 OMDoc specification



Welcome to GenCS

Overview:

The purpose of this two-semester course is to give you an introduction to what the Science in "Computer Science" might be. We will touch on a lot of subjects, techniques and arguments that are of importance. Most of them, we will not be able to cover in the depth that you will (eventually) need to know them. That will happen in your second year, where you will see most of them again, with much more thorough treatment.

Computer Science:

We are using the term "Computer Science" in this course, because it is the traditional anglo-saxon term for our field. It is a bit of a misnomer, as it emphasizes the computer alone as a computational device, which is only one of the aspects of the field1. Other names that are becoming

author: kohlhase , created: 2007-07-23 06:16:14

```
<omdoc>
  <narrative>...<narrative>
  <content>...<content>
</omdoc>
```

Another possible scenario for `panta rhei` is the collaborative specification of the next version of the mathematical markup language OMDoc [Koh06]. For this we are currently transforming the OMDoc1.2 specification document from LaTeX to the new OMDoc document model proposed in [KMM07] using LaTeXML [Mil]. The figure above displays the workflow: The narrative structure of the OMDoc specification is used to build up the table of contents in `panta rhei`, while the content itself is transformed into HTML pages. In this case study, `panta rhei` will be particularly evaluated with respect utility for the collaborative extension, revision, and change of content: Collaborators are facilitated to publicly discuss, criticize, and, in particular, to collaboratively revise and to extend the specification. For example, by linking comments to a specific section on the specification, the individual argumentation can be tracked and evaluated more easily. In addition, critical or *hot* topics can be automatically identified and, eventually, facilitate the discussion on the face2face meetings for specifying OMDoc2.0.

2.4 MKM scenario

The third case study of `panta rhei` will take place within the Conference on Mathematical Knowledge Management (MKM) 2008. The conference papers will be marked-up allowing for the transformation in OMDoc and, in particular, for displaying them in `panta rhei`. After logging into `panta rhei`, the narrative structure on the right displays the names of the papers in the proceedings. However, after selecting a paper from the proceedings, the narrative structure of the paper will be displayed on the left. Both narrative structure and content of the paper can be identified based on the semantic markup according to the OMDoc document model [Koh06; KMM07] under discussion for MKM proceedings.

In contrast to the GenCS scenario, the MKM scenario requires a more elaborate feedback loop, allowing readers to post questions, enter comments, mark errors, as well as to reference other papers. In consequence, authors receive feedback from the whole community rather than a few reviewers. Moreover, enabling the collaborative review of papers after the submission meshes well with the idea of publishing post-proceedings based on the communities' rating.

Analogous to the GenCS scenario, participants of the conference can rate the helpfulness and soundness of the argumentation of other participants. In addition to the GenCS scenario, authors can rate answers to questions referring to their publication, eventually building a web-of-trust of participants they trust and whose answers they accept.

Besides offering an on-the-fly discussion platform, `panta rhei` provides statistical analysis of the users' interaction and feedback. For example, based on the number of references to a paper, the most viewed, the most discussed, or the most popular paper can be identified and are accordingly ranked in the proceedings' listing.

3 Implementation

In `panta rhei v1` we use a web interface which integrates the course material as HTML `iframe`, where the interaction model is implemented in JAVASCRIPT an PHP. Since we have access to all subsystems and materials, we can simply embed the respective function calls into the generated presentations. In later extensions where we cannot seed the subsystems with function calls we may use an extension of Annozilla [Ann07], a firefox plug in for annotating web pages or use Greasemonkey [Gre07] like approach.

Figure 6 shows gives a detailed overview over `panta rhei v1`. The arrows represent control flow and user actions, we will reference them by number in the explanations below.

Table of Contents Selecting a link in the *table of content navi* displays the respective content page in the center (1) and triggers the updating of the forum threads (2), i.e. all forum threads that reference the page or its section are displayed.

Forum Thread Selecting a link in the *forum thread navi* displays the respective posting and its direct replies in the center (3).

Content Selecting a link on a page, triggers the updating of the forum thread (4), e.g. all postings that are not linked to the item currently selected are crossed out in the thread navi. Clicking on a posting displays a popup, in which the selected item can be annotated (5).

PopUp The annotation pop up draws on the the ANIT description to display the respective meta-information (6). It is called by a JavaScript function embedded in the content page and receives the id of the item to be annotated as parameter. The annotation PopUp also stores the annotation as postings in the forum table (7).

Rating The rating component allows user's to asses content and forum postings. It stores the respective data in the rating table (8).

Search The search draws on an full text index in the content and forum table (9). Moreover, a social ranking is provided which draws on the community ratings (10).

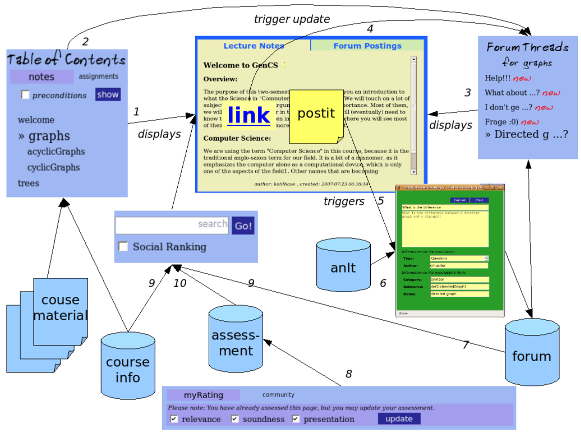


Figure 6: `panta rhei` components.

In `panta rhei`, we can make annotations at the *semantic level* by describing annotatable items (ANIT). An ANIT is described by a name (e.g. *directed graph*), its type (e.g. *symbol*), and an unique id (e.g. *lecture/graphs#def2:directedGraph1*), which consists of the folder (*lecture*), the page-name (*graphs*) and the reference of the ANIT in the page (*def2:directedGraph1*, where as *def2* is the container element). This facilitates annotation and linking on different granularity levels. Moreover, in the search for questions/answers it may be beneficial to also consider Q/A of semantically related topics (e.g. sub-elements, prerequisites or consequences). In the first prototype, the ANITs to be discussed have to be manually described and marked-up. In later extension, `panta rhei` will offer an import that will support authors with a (semi) automatic description and mark-up of content depending on the input format. For example, importing documents in the OMDOC format allows the system to automatically markup ANITs and to extract meta-information about an ANIT from the OMDOC element it was generated from. Moreover, drawing on the OMDOC *document ontology* [Doc] the current ANIT description can be extended allowing to provide additional services.

In order to combine content and its discussion in `panta rhei`, the discussion items, such as forum postings or chat entries, have to include a topic and a category field in order to store the respective annotation path as well as the type of the annotation. This currently seems to be the only integration restriction. In the first prototype forum postings are described by a unique id, a reference to either another posting or an ANIT as well as a category (i.e. 'Advise', 'Change', 'Comment', 'Example', 'Explanation', 'Question', 'SeeAlso'. These categories are based on the annotation types in Annozilla [Ann07]).

Adaptations such as the ranking of postings is be based on a user profile, which students can voluntarily create. For example, they can set filter preferences to restrict the displayed postings to the ones posted before a certain timestamp. Alternatively, `panta rhei` will offer to restrict the lists to postings written by a specific user or a community such as 'all TAs'. In future work, we also want to consider further user data which eventually can be used to improve the ranking as well as adaptation of the content and forum.

4 Outlook and Conclusion

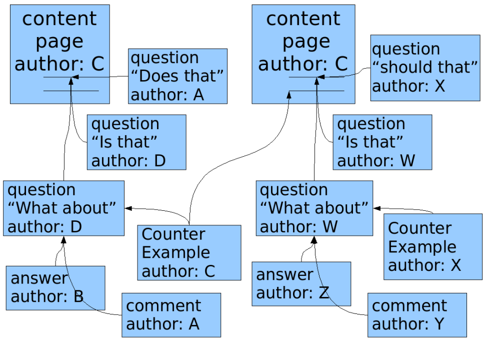
We have introduced `panta rhei`, our collaborative and interactive community reader, which facilitates the collaborative discussion, revision, and rating of knowledge while embedding it into a narrative structure to most easily explore and find information. We have discussed three case studies to describe the functionality of the system, each of them illustrating different challenges of the system. In its first version, `panta rhei` is driving the educational scenario, in which the system combines marked-up course material in HTML format, the narrative structure, and a course forum. Nevertheless, we are focusing on a general framework, in which arbitrary content can be presented. For example, one could use `panta rhei` to set up a course that solely references web pages in the WWW. As long as `panta rhei` is provided with a respective structure, either automatically extracted or manually specified, any content can be displayed without any further requirements on the presentation component. This could very well be given in the standardized data format *IMS Content Packaging* [IMS07a], which unifies the structure and storage of online course material. The data format was specified by the IMS Global Learning Consortium [IMS07b] to ensure the unique specification of learning material and the interoperability between learning systems. By being able to import IMS Packages, `panta rhei` could reuse existing learning material and, in particular, extract their narrative structures from the IMS structure.

4.1 Extending `panta rhei`

In later versions of `panta rhei`, we will also integrate our formula search engine MathWebSearch [KŞ07; Mat07] as well as an improved full text search. The MathWebSearch system is a content-based search engine for mathematical formulae. It indexes OPENMATH [BCC+04] and content MATHML [ABC+03] formulae.

Another planned extension concerns the forum functionality. All keywords in the forum that relate to the content in the presentation component will be marked up and, eventually, link to the respective content element. We are also considering the use of ontologies to formalize equivalent keywords that have the same meaning.

4.2 Identifying Communities of Practice


content page author: C
question "Does that" author: A
question "Is that" author: D
question "What about" author: D
Counter Example author: C
answer author: B
comment author: A
content page author: C
question "should that" author: X
question "Is that" author: W
question "What about" author: W
Counter Example author: X
answer author: Z
comment author: Y

`panta rhei` is particularly developed to analyze communities of practice in the area of mathematics [KK06]. In a first step, we will analyze the interaction between users in the threaded structure of references between documents, including annotations, postings, and content pages. In the figure above, two content pages as well as several interlinked postings are displayed. Analyzing these document-based interactions between authors allows for identifying two virtual communities: One consisting of users A, B, C, D and the other one including users C, X, Y, Z, W. However, the number and size of the communities strongly depends on the given parameters of the clustering algorithm, e.g. the number of interactions that classify a user to be a member of the community. Moreover, communities are emergent structures that strongly depend on the behavior of their members, who frequently change their communities, interact with other communities, and are often members of more than one communities.

References

[ABC$^+$03] Ron Ausbrooks, Stephen Buswell, David Carlisle, Stéphane Dalmas, Stan Devitt, Angel Diaz, Max Froumentin, Roger Hunter, Patrick Ion, Michael Kohlhase, Robert Miner, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, and Stephen Watt. Mathematical Markup Language (MathML) version 2.0 (second edition). W3C recommendation, World Wide Web Consortium, 2003. Available at `http://www.w3.org/TR/MathML2`.

[Ann07] Annozilla Project, seen July 2007. available at `http://annozilla.mozdev.org/`.

[BCC$^+$04] Stephen Buswell, Olga Caprotti, David P. Carlisle, Michael C. Dewar, Marc Gaetano, and Michael Kohlhase. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004. `http://www.openmath.org/standard/om20`.

[CNX07] CONNEXIONS. Project homepage at `http://www.cnx.org`, seen February 2007.

[Doc] The cnx and omdoc document ontology.

[Gre07] Greasemonkey Project, seen July 2007. project web page at `http://www.greasespot.net/`.

[IMS07a] Ims content packaging information model. Project web site at `http://www.imsglobal.org/content/packaging/index.html`, seen July 2007.

[IMS07b] Ims global learning consortium. web site at `hhttp://www.imsglobal.org/`, seen July 2007.

[KK06] Andrea Kohlhase and Michael Kohlhase. Communities of Practice in MKM: An Extensional Model. In Jon Borwein and William M. Farmer, editors, *Mathematical Knowledge Management, MKM'06*, number 4108 in LNAI, pages 179–193. Springer Verlag, 2006.

[KMM07] Michael Kohlhase, Christine Müller, and Normen Müller. Documents with flexible notation contexts as interfaces to mathematical knowledge. In Paul Libbrecht, editor, *Mathematical User Interfaces Workshop 2007*, 2007.

[Koh06] Michael Kohlhase. OMDOC – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006.

[KŞ07] Michael Kohlhase and Ioan Şucan. System description: MATHWEBSEARCH 0.3, a semantic search engine. submitted to CADE 21, 2007.

[Mö07] Christine Müller. Panta rhei. Project Home Page at `http://kwarc.info/projects/panta-rhei/`, seen August 2007.

[MAF$^+$01] E. Melis, E. Andres, A. Franke, G. Goguadse, P. Libbrecht, M. Pollet, and C. Ullrich. ACTIVEMATH System Description. In Johanna D. Moore, Carol Luckhard Redfield, and W. Lewis Johnson, editors, *Artificial Intelligence in Education*, volume 68 of *Frontiers in Artificial Intelligence and Applications*, pages 580–582. IOS Press, 2001.

[Mat07] Math WebSearch a semantic search engine. web page at `http://search.mathweb.org`, seen June 2007.

[Mil] Bruce Miller. `LaTeXML`: A LaTeX to xml converter. Web Manual at `http://dlmf.nist.gov/LaTeXML/`.

[Wen05] Etienne Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 2005.

Managing Variants in Document Content and Narrative Structures

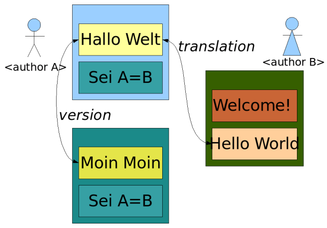
Michael Kohlhase, Achim Mahnke, Christine Müller*
Computer Science, Jacobs University Bremen, Germany
m.kohlhase/c.mueller@jacobs-university.de, achim.mahnke@dfki.de

1 Introduction

Sharing, reuse, and adaptivity are the key to efficient sustainable development, i.e. continuous long-term usability of document content. They need to be supported by tools and methods taking into account the semantic structure of a document in order to facilitate adaptivity and change management.

A particular issue in reuse and change management is the use of variants to handle consistent variations of documents, e.g. translations into different natural languages. In order to overcome the often found copy-and-paste style of reuse, the management of document variants have to be integrated into markup languages and document processing tools in order to encourage the integrated development of variants. On top of such documents, we can offer improved services like:

- Adaptation of the presentation to the reader's environment, e.g. to the kind of display, medium, or language
- Adaptation to different contexts, e.g. to the intended audience or learner's knowledge
- Checking consistency conditions, e.g. when translating a document, all constituent parts have to be present in the target language



Examples The figure to the right provides an example for the application of language variants: Author A has created a document in German. Author B, who is writing about a similar topic in English, would like to reuse $A's$ text fragments in her document. She first selects the respective fragment, *translates* it into English, and finally *relates* her translated document fragments to the original (German) one in author $A's$ document. Once the system knows about this relation, it is able to e.g. assist other authors in the *translation* of their documents. For example, if author A decides to translate his document to English, parts of the work have already been accomplished by author B. Being able to relate the German and English variants to each other, the system can help A to find and reuse the English fragment in alternative to the German one.

However, the challenge of language variants has already been addressed in other multilingual document management systems. To show more potentials of the variant concept, let us consider changes over time: Eventually, author A changes his text and, thus, creates a new *version* of one of his document fragments. This has to be reflected somehow in the document model. By modeling document versions as variants, we can also address the problem of versioning.

Variant Dimensions For the purpose of this document, we will stick to the above example scenario, but want to give an impression on what we see as candidates for document variations which can be managed by our proposed variant module:

Language-oriented In addition to natural language variants, as we have exemplified above, formal language variants such as programming or specification languages can also be suitable variant dimensions: For example, a book on teaching object-oriented programming can either have examples in JAVA and C++ depending on the concrete context it is created in.

Media-oriented The intended output medium has a great influence on how the document is written. While print-outs, e.g. a script for a lecture, are usually self-contained, the variant for reading on an electronic device would contain links to other documents or some material which cannot be printed, e.g. videos or animations.

Audience-oriented Experiences in creating lecture material for eLearning has shown that only parts of the content have to be adapted when preparing courses for different but close enough audiences, e.g. a higher education course in mathematics can essentially be the same document with variant examples tailored to the target audience, thus, the intended audience could be modeled by a variant dimension.

Obviously, this zoo of variant dimensions and relations cannot be handled with the simple methods developed for document management systems, which handle multilingual variants inside documents (or in localization files for programs) and versioning at the

* We would like to thank the members of the KWARC group and the DFKI Bremen for the fruitful discussions and continuous feedback on our work.

file-system level. We propose a general variant management infrastructure as an extension to knowledge repositories. Our approach operates across document boundaries and allows arbitrary groupings of variant objects, yielding an multi-dimensional information resource for complex presentation and variant management systems.

With our work on variants, we are building on ideas developed in the course of the MMiSS project (MultiMedia Instruction in Secure Systems) [MKB04; KBHL$^+$03], which annotates document fragments by a set of *variant dimensions*.

To each element of a structured document the MMiSS author can attach a set of attributes stating to which variant domains this element belongs (cf. listing below). All variants of a certain element have the same id and a variant selection mechanism is used to choose between these alternative presentations.

```
\begin{ assertion }[Label=1, Language=de, LevelOfDetail=low]
  Es sei ...
\end{ assertion }
```

However, the MMiSS approach lacks a clear separation of narrative and content structures.

Our work further relates to the approach Gergatsoulis et.al. [GSK01] have developed for semistructured data. However, their solution is designed for data structures, not for narrative documents and they propose an extension to XML instead of providing a proper XML application paradigm for variants.

In this paper, we will propose a solution for the management of variants using the OMDoc (Open Mathematical Documents [Koh06]) format as a concrete application paradigm and propose an OMDoc module that extends the format. Note that the methods described in this paper are independent of the mathematical aspects of OMDoc. We will only need a content/narrative document infrastructure here; other content-oriented document markup languages can be extended analogously.

Specifying the Narrative Structure of Documents The semantic markup language OMDoc1.2. provides two ways to mark up the knowledge contained in a mathematical document and its structure: *Content* OMDoc*s* are "knowledge-centered documents that contain the knowledge conveyed in a document" [Koh06]. In contrast, *narrative* OMDoc*s* are used to "reference the knowledge[-centered documents] and add the theoretical and didactic structure of a document". The *Content* OMDoc*s* are stored in a knowledge repository, which is called *content commons* according to the terminology of the educational knowledge repository ConneXions [CNX07]. The combination of the narrative structure and the (mathematical) content of a document as the formal representation of a document model, has been defined by Normen Müller as NarCons, i.e. two-dimensional graphs consisting of a **nar**rative layer and a **con**tent layer [Mül06]. The NarCon approach in OMDoc1.2 has been extended by the specification in [KMM07], which will be the starting point for the variant specification proposed in this paper: Accordingly, documents will be structured by `omdoc` elements which have two optional children to markup the content and structure of the document: a `narrative` element specifying the narrative structure of the document and a `content` caching its content and, thus, henceforth referred to as the *content cache* of the document.

The listing below presents an the OMDoc representation of author A's text according to the document model in the [KMM07]: The `narrative` element of the document includes meta information on the document, e.g. the author, as well as references to the content of the document. The `content` of the document caches the text fragments of author A's document, which are referenced in the narrative structure.

```
<omdoc>
  <narrative>
    <metadata><dc:author>A</dc:author></metadata>
    <ref xref="#t1">
  </narrative>
  <content>
    <omtext xml:id="t1">Hallo Welt</omtext>
  </content>
</omdoc>
```

2 A Variant Module for OMDoc

We introduce two new markup elements named `variant` and `vardim`. The `variant` element is used to express the fact, that an object (specified in it's `from` attribute) is a variant of another (specified in the `to` attribute). As we have seen above, there are multiple variant relations, so we need an extensible vocabulary of relation types. Relations are mathematical objects, which can be described in the OMDoc format itself, therefore we represent the type of the relation as a *symbol* in a content dictionary (cf. Listing 1), which can be referenced by the `name`, `cd`, and `cdbase` attributes. The `vardim` element categorizes an object (given in the `for` attribute) in terms of a *variant dimension* (e.g. language, version, format, formalism). Again, we use symbols in content dictionaries for the dimensions (so the `vardim` element carries the attributes `name`, `cd`, and `cdbase`). The `vardim` element represents the value of the object in this dimension, it is specified as a mathematical object. The following RelaxNG [vdV04] grammar summarizes the proposed extensions.

```
sym.att  = attribute cd{xsd:NCName},
           attribute name {xsd:NCName},
           attribute cdbase {xsd:anyURI}?
 variant = element variant {id. att ?,sym.att ,
              attribute from{xsd:anyURI*},
              attribute to{xsd:anyURI*}}
vardim   = element vardim {id. att ?,sym.att ,
              attribute for{xsd:anyURI*},
              (math|OMOBJ)}
```

To fortify our intuition, let us re-consider the languages example from the introduction. In OMDoc, A's text would now have the following form.

```
<omdoc>
  <narrative>
    <metadata><dc:author>A</dc:author></metadata>
    <ref xref="#t1">
  </narrative>
  <content>
    <omtext xml:id="t1">Hallo Welt</omtext>
    <vardim for="#t1" cd="language" name="langdim">
      <math><csymbol cd="language" name="de"/></math>
    </vardim>
  </content>
</omdoc>
```

The content part of the document contains the text itself and the specification that the text has the value “German” in the “language” dimension. Both of these concepts are defined as symbols in the content dictionary in Listing 1 below.

Note that the NARCON in the listing above is just the internal representation in the knowledge repository, or the form that systems will use to communicate to each other, not what the author actually writes. Everything except the string “Hallo Welt” can be generated by the authoring environment.

Listing 1: A Content Dictionary for the Language Variant Dimension

```
<theory name=”language”>

  <symbol name=”translation”>
    <metadata>
      <dc:description>
        This variant relation specifies that text
        fragments are translations of each other.
      </dc:description>
    </metadata>
  </symbol>

  <symbol name=”langdim”>
    <metadata>
      <dc:description>The language dimension.</dc:description>
    </metadata>
  </symbol>

  <symbol name=”de”>
    <metadata>
      <dc:description>
        This variant dimension specifies that a text
        fragment is written in German
      </dc:description>
    </metadata>
  </symbol>
  ...
  <symbol name=”en”>...</symbol>
  <symbol name=”fr”>...</symbol>
</theory>
```

Referencing the same CD, the internal representation of B’ text has the following form.

```
<omdoc>
  <narrative>
      <metadata><dc:author>B</dc:author></metadata>
      <ref xref=”#t9”>
  </narrative>
  <content>
    <omtext xml:id=”t9”>Hello World</omtext>
    <vardim for=”#t9” cd=”language” name=”langdim”>
      <math><csymbol cd=”language” name=”en”/></math>
    </vardim>
    <variant from=”#t9” to=”#t1”
        cd=”language” name=”translation”/>
  </content>
</omdoc>
```

Here B has to specify that her text is a translation of A’s, which the new OMDOC module represents by the `variant` element in the content part. Again all other information is supplied by the authoring environment which could also support B in the translation process.

Later, A updates his text (he is from northern Germany). The authoring tool will add the following two lines to the content of his document and changes the `href` pointer on the `ref` element to `#t2`.

```
<omtext xml:id=”t2”>Moin moin</omtext>
<variant from=”#t2” to=”#t1” cd=”version” name=”change”/>
```

Note that both versions of the text are now in the content part, and thus accessible. That `#t2` is a newer version of `#t1` is specified by reference to the following content dictionary, which also serves as a background reference and documentation for the version management tool, which deduces the language dimension of the new text to be “German” by the assumption that primary languages are invariant under version changes. In OMDOC, such assumptions can be encoded by `axiom` elements.

```
<theory name=”version”>
  <symbol name=”change”>...</symbol>
  <axiom>
    If x change y and vardim(lang, y) = z then vardim(lang, x) = z.
  </axiom>
</theory>
```

3 Abstract Documents

Note that the narrative parts of the documents in our examples above reference concrete objects. In particular, they fix a concrete variant (i.e. a concrete language and version). Following this approach, documents are *extensionally* described, i.e. by their specific parts. Given the concrete text fragments in a knowledge base, an author can write an “abstract” document, which does not fix specific variants. He then leaves the instantiation of the document to a presentation engine that adapts the document to the concrete situation and context of a reader. In this *intensional* approach authors solely specify the dimension of the new document instead of pre-selecting concrete variants.

The extension of an intensionally given document can change over time. For example, if the intention of an abstract document D is to be “as German as possible, else English”, and the text fragments in the content commons are translated step by step, then the extension of D changes from an all-English text to an all-German one via many mixed intermediate stages. While in the extensional approach, authors would have to re-write their concrete document whenever another German text fragment has been translated, authors of intensionally specified documents are relieved from frequently updating the German version of the text.

Of course, the presentation engine could take the document of author A (cf. section 2), follow all `variant` relations to retrieve alternative objects with the appropriate variant dimension, and assemble them to a new presentation. But it is much more intuitive to introduce an explicit *intensional* level of representation by introducing “abstract objects” that act as place holders for all the possible variant objects which can instantiate it. In our example above, we could do this by adding an abstract text fragment (which is empty, since we do not have a language-independent representation for text) together with variant specifications.

```
<omtext xml:id=”phw”/>
<variant from=”#t1” to=”#phw” cd=”mks” name=”concretization”/>
<variant from=”#t9” to=”#phw” cd=”mks” name=”concretization”/>
<variant from=”#t2” to=”#phw” cd=”mks” name=”concretization”/>
```

Then we can specify an abstract document simply as

```
<omdoc><narrative> <ref xref=”#phw”></narrative></omdoc>
```

Note that this approach is general enough to accommodate the analysis of the Mathematical Knowledge Space (MKS) ([KK05]), which suggests to distinguish the *Presentation Objects* found in formatted

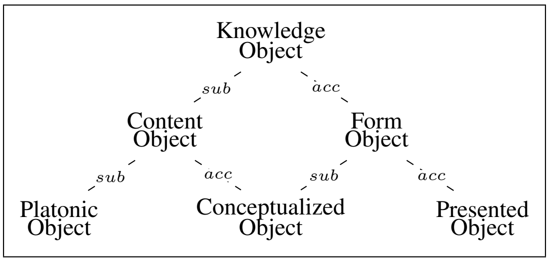


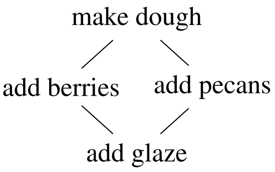
Figure 1: The MKS triangle

documents, from its substance, which is referred to as the *Platonic Object*. While creating his document, the author instantiates the Platonic Object by choosing a suitable representation (conceptualization) and presentation form for his knowledge item and finally creates the concrete Presentation Object. The same knowledge substance can therefore be represented differently (which leads to *Content Objects*) and presented in many ways resulting eventually in a set of objects which we consider as variants of the same Content or Platonic Object (see fig. Figure 1).

4 Narrative Variants

Up to now, we have restricted the variant relation to content objects, where it is supported by the notion of a *content cache*, which allows duplication and reordering.

However, it would be nice to extend the variant infrastructure to narrative structures as well, e.g. to mark up two linearizations of a knowledge structure as equivalent: For example, if two steps in a recipe for Blueberry-Pecan Muffins are independent, the choice of which to take first is arbitrary. In the recipe on the right the two narrative structures "make dough, add berries, add pecans, add glaze" and "make dough, add pecans, add berries, add glaze" are meaningful.



Unfortunately, the narrative part of an OMDoc document is presented directly to the user and does not allow duplicates like the content cache. But on the level of the knowledge base (or the World Wide Web of NarCons for that matter), all document fragments can be addressed by URIs in the `to` and `from` attributes of `variant` elements.

Listing 2: Narrative Variants

```
<narrative xml:id="pintro"/>

<narrative xml:id="p22" type="section">
  <ref xref="#example1"/>
  <ref xref="#example2"/>
  <ref xref="#elaboration3"/>
</narrative>

<narrative xml:id="p23" type="section">
  <ref xref="#csintro"/>
  <ref xref="#example2"/>
  <ref xref="#example3"/>
</narrative>
```

So, given the three narrative structures in Listing 2 and the variant information in Listing 3 our envisioned presentation engine can adapt an abstract course to the disciplinary background of the audience.

Listing 3: Variant Information

```
<vardim for="#p22" cd="area" name="audience">
  <math><csymbol cd="area" name="math"/></math>
</vardim>
<vardim for="#p23" cd="area" name="audience">
  <math><csymbol cd="area" name="cs"/></math>
</vardim>
<variant from="#p22" to="#pintro" cd="mks" name="???"/>
<variant from="#p23" to="#pintro" cd="mks" name="???"/>
```

5 Conclusion

We have presented an infrastructure for representing and managing variants in content-oriented documents: In essence, variants are accumulated in the content- and document commons, and their categories and relationships are specified by two new content elements. While our proposal integrates very well with the OMDoc format, since that provides a notion of content dictionaries and thus extensible vocabularies, there is no reason other content-commons-based formats could not be similarly extended.

The general treatment of variants as first-class citizens renders special constructs e.g. to handle multilinguality obsolete. For instance OMDoc 1.2 took great pains to cater for multiple language variants in the language itself: All top-level elements had multi-language groups of `CMP` elements (Commented Mathematical Properties) as children. This allowed multilinguality, but left the exact relation between children unspecified (though the specification hinted at a translation relation in places). Moreover, the process of adding new languages to a given element was unclear, unless the translator had access to the original document. In the new proposal, we can do away with `CMP`s altogether (and have done so in the exposition above). The relation between language variants is made explicit in a content dictionary (other relations like "rough translation" or "paraphrase" could be specified) and extensibility by translators is built into the system. Incidentally OMDoc 1.2 also allowed `FMP` (Formal Mathematical Properties) elements as siblings with the same "translation" intuition. But the "formalization" relation is an asymmetric relation (we formalize mathematical statement given in natural language in a first-order logic but not vice-versa). This highlights the fact that "translation" is an equivalence relation, which variant-aware knowledge management systems need to take into account. This can be documented in the respective content dictionaries as above.

6 Outlook

Document Adaptation based on variants As shown in Section 3, the *intensional* variant model allows to create *abstract documents*, i.e. documents that solely consists of abstract objects rather then concrete variants. In order to instantiate abstract documents, we need to explicate the variant dimension of the appropriate concrete document, henceforth referred to as the *variant context*. To do so, we introduce a `vcontext` element in OMDoc, which encapsulates a set of `vardim` elements (cf. Section 2).

The RelaxNG [vdV04] grammar of `vcontext` is given below:

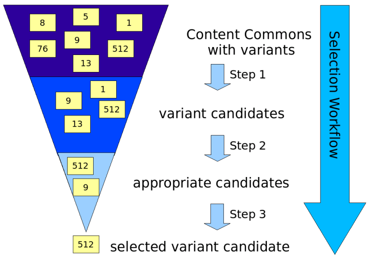
```
vcontext = element vcontext {id. att ?,
                  attribute base list {xsd:anyURI∗},vardim∗}
```

The optional `vcontext` element is a collection of zero or more `vardim` elements, which are used to specify dimensions such as *language* or *audience*. The optional `base` attribute contains a whitespace-separated list of URIs pointing to variant contexts of other NarCons, hence it can be used to inherit variant contexts. The effective variant context is computed by the variant selection algorithm, whereas the order of the `vardim` elements defines their prioritization, e.g. the first `vardim` has a greater weight than the second one.

The listing below proposes the specification of a `vcontext` in OMDoc: Three `vardim` elements are used to specify the two dimensions language and audience, where two languages (German and English) are given. The order of the `vardim` elements defines their priority: Here, the user is looking for variants for mathematicians which should preferably be German, but if the latter are not available the user would also accept English fragments.

```
<vcontext xml:id="vcontext−id" base="#inherited−vcontext−id">
  <vardim xml:id="d1" cd="area" name="audience">
    <math><csymbol cd="area" name="math"/></math>
  </vardim>
  <vardim xml:id="d2" cd="language" name="langdim">
    <math><csymbol cd="language" name="de"/></math>
  </vardim>
  <vardim xml:id="d3" cd="language" name="langdim">
    <math><csymbol cd="language" name="en"/></math>
  </vardim>
</vcontext>
```

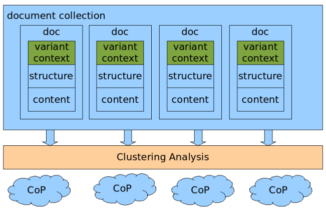
Implementation of the Variant Selection



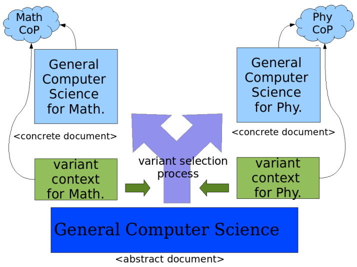
In order to tailor an abstract document to a specific *variant context*, a *variant selection process* is required, which is based on the specifications of variants by the `variant` and `vardim` elements. The figure above displays the selection workflow: The first step is to retrieve all variant candidates for the document's constituent `omdoc` elements from the content commons. In the second step the appropriate variants are selected based on the variant dimensions given in the `vcontext` element. If multiple variants are available, these are further filtered depending on the priority of the dimensions. Given the respective variants, the concrete document, including a specific narrative structure and a concrete content, can be generated: The references in the narrative structure of the document are bended to the selected variants, which are stored in the `content` cache of the document.

During the selection process various exceptions have to be handled: For example, the variant context can be too restrictive, e.g. by only accepting "German" variants, so that no variant candidate can be found if there are simply no "German" text fragments in the content commons. This could be solved in a naive way by offering no results for the respective object to be displayed, but this is obviously not a preferable solution. Instead, the users are encouraged to provide alternative values for a respective dimension, ordering them according to their preferences, e.g. they can also accept "English" texts although these are less preferred. For example, in the above `vcontext` example, the user wants to see variants which are tailored to mathematicians (first `vardim` element with id "d1") and which are preferably written in "German", but can also be "English" if no "German" variant can be retrieved from the content commons.

Identifying Communities of Practice (CoPs) based on Variants



By interpreting the selection of concrete variant dimensions and their values as *shared practice*, we want to identify *communities of practice* [LW91] (CoPs) based on similar preferred variant contexts. Vice versa, we aim at describing CoPs by their most frequently used variant dimensions for various concrete documents (cf. figure on the left).



The modeling of CoPs based on `vcontexts` will, in particular, support the previously described variant selection process: Instead of requiring a `vcontext` for each user, CoP models allow for reusing existing `vcontext` for users of the same community. For example, let us assume that students of a course are members of one CoP. A teacher could now specify the `vcontext` for two different courses, e.g. his General Computer Science (CS) lecture for Mathematicians and the CS lecture for Physicians. Depending on a student's course, the selection process can now reuse the variant information in order to instantiate abstract course material, i.e. generating appropriate (concrete) lecture notes for both communities, i.e. the mathematicians and physicians (cf. figure on the right).

References

[CNX07] CONNEXIONS. Project homepage at `http://www.cnx.org`, seen February 2007.

[GSK01] Manolis Gergatsoulis, Yannis Stavrakas, and Dimitris Karteris. Incorporating dimensions in xml and dtd. In *Database and Expert Systems Applications, 12th International Conference, DEXA 2001 Munich, Proceedings*, pages 646–656, 2001.

[KBHL$^+$03] B. Krieg-Brückner, D. Hutter, A. Lindow, C. Lüth, A. Mahnke, E. Melis, P. Meier, A. Poetzsch-Heffter, M. Roggenbach, G. Russell, J.-G. Smaus, and M. Wirsing. Multimedia instruction in safe and secure systems. In M. Wirsing and R. Hennicker D. Pattinson, editors, *Recent Trends in Algebraic Development Techniques*, volume 2755 of *Lecture Notes in Computer Science*, pages 82–117. Springer-Verlag Heidelberg, 2003.

[KK05] Andrea Kohlhase and Michael Kohlhase. An Exploration in the Space of Mathematical Knowledge. In Michael Kohlhase, editor, *Mathematical Knowledge Management, MKM'05*, number 3863 in LNAI. Springer Verlag, 2005.

[KMM07] Michael Kohlhase, Christine Müller, and Normen Müller. Documents with flexible notation contexts as interfaces to mathematical knowledge. In Paul Libbrecht, editor, *Mathematical User Interfaces Workshop 2007*, 2007.

[Koh06] Michael Kohlhase. OMDOC – *An open markup format for mathematical documents [Version 1.2].* Number 4180 in LNAI. Springer Verlag, 2006.

[LW91] Jean Lave and Etienne Wenger. *Situated Learning: Legitimate Peripheral Participation(Learning in Doing: Social, Cognitive and Computational Perspectives S.).* Cambridge University Press, 1991.

[MKB04] A. Mahnke and B. Krieg-Brückner. Literate ontology development. In Robert Meersman, Zahir Tari, and Angelo Corsaro et al., editors, *On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops*, volume 3292 of *Lecture Notes in Computer Science*, pages 753–757. Springer; Berlin; http://www.springer.de, 2004.

[Mül06] Normen Müller. An Ontology-Driven Management of Change. In *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) conference proceedings*, 2006.

[vdV04] Eric van der Vlist. *RELAXNG: A simple schema language for XML.* O'Reilly, 2nd edition, 2004.

Know the Right People?
Recommender Systems for Web 2.0

Stefan Siersdorfer
University of Sheffield, UK
Dept. of Information Studies
s.siersdorfer@sheffield.ac.uk

Sergej Sizov
University of Koblenz, Germany
Dept. of Computer Science
sizov@uni-koblenz.de

Paul Clough
University of Sheffield, UK
Dept. of Information Studies
p.d.clough@sheffield.ac.uk

Abstract

Web 2.0 applications like Flickr, YouTube, or Del.icio.us are increasingly popular online communities for creating, editing and sharing content. However, the rapid increase in size of online communities and the availability of large amounts of shared data make discovering relevant content and finding related users a difficult task. Web 2.0 applications provide a rich set of structures and annotations that can be mined for a variety of purposes. In this paper we propose a formal model to characterize users, items, and annotations in Web 2.0 environments. Based on this model we propose recommendation mechanisms using methods from social network analysis, collaborative filtering, and machine learning. Our objective is to construct collaborative recommender systems that predict the utility of items, users or groups based on the multi-dimensional social environment of a given user.

1 Introduction

1.1 Motivation

We have entered the "knowledge age" where economic power or wealth is based on ownership of knowledge [15] and the ability to utilize knowledge for the improvement of services or products. "Knowledge workers" already outnumber any other kind of worker in highly developed economic systems. Their work is characterized by complex human-centric processes [20] describing information intensive and complex tasks comprising of the creation, retrieval, digestion, filtering, and sharing of (large amounts of) knowledge.

Basically, the paradigm of a flexible environment that supports the user in producing, organizing, and browsing the knowledge is not new. It originates in the early 1940s, long time before the first personal computers and new communication tools like the internet became available. The conceptual design of Vannevar Bush's Memex [10] (an acronym for Memory Extender) is probably the most cited (e.g. [14]) and criticized (e.g. [8]) representative of the early conceptual work. In his article, Bush describes the integrated work environment that was electronically linked to a repository of microfilms and able to display stored content and automatically follow references from one document to another. A number of visionary ideas from this early conceptual work can be recognized in state-of-the art information systems (cross-references between documents, browsing, keyword-based annotation of documents using the personal "codebook", automatic generation of associative trails for content summarization, etc.).

However, an important difference attracts attention if we consider the aspect of collaboration. In fact, knowledge workers of today are never working isolated. The internet and the World Wide Web (WWW) provide an infrastructure on top of which a variety of communication channels and collaborative environments have been established. Collaboration is one of the major tasks of the knowledge worker as it denotes the exchange of information and transfer of knowledge. It is vital for any collaborative human work, e.g. for coordinating activities, reporting on work progress, discussing solutions and problems, or disseminating new information. Efficient collaboration infrastructure is probably one of the key differences of modern work environments in contrast to isolated Memex-style solutions from the past.

For this reason, it is not surprising that knowledge and data sharing in Web 2.0 applications has rapidly gained popularity. Despite disagreement on the exact definition of Web 2.0, it is common to find community and collaboration as key concepts in this latest online phenomenon. Increasingly, online content is being created, edited and shared by whole communities of users, demonstrated by the popularity of applications such as Flickr[1], YouTube[2] and Del.icio.us[3]. Web 2.0 applications provide a rich set of structures and annotations that can be mined for a variety of purposes. For example, Flickr postings are accompanied with a variety of descriptive metadata, such as creator (and/or owner), a textual description, thematic tags, temporal and geographic information and comments by other Flickr users on specific regions of uploaded pictures. Using these structures, a variety of relationships between users, tags, pictures, and groups can be explored.

Collaboration in Web 2.0 environments induces new problems and challenges. In many cases, the size of online communities has increased rapidly over the last decades and large amounts of shared data are now available. This makes the discovery of relevant content and finding users with shared interests a difficult enterprise. Ideally, the Web 2.0 platform should provide the user with adaptive browsing mechanisms and recommendations for potentially relevant content, users, or annotations. Unfortunately, recent platforms are rather limited in supporting self-organization. The vast majority just offers explicitly defined groups/topics of interest along with suitable subscription, dissemination, and communication mechanisms.

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

[2] http://www.youtube.com

[3] http://del.icio.us/

Static groups, however, cannot properly reflect and support the evolution of user interests and one-day thematical hotspots, such as “Valentine’s day” or “recent Oscar nominations”. Therefore, it is necessary to have views that put together users and content by identifying their topics of interest and current demands “on the fly”.

In this article, we address the following questions: What is the generalized model for recommender scenarios in Web 2.0 applications? How can we represent and utilize available meta information? How can we apply automatic techniques for making recommendations? In doing so, we introduce a formal notion of Web 2.0 relationships, formalize the recommender problem, discuss possible solutions and evaluation methodologies, and identify promising open questions and research directions for upcoming research.

1.2 Analysis of Requirements

Many algorithms and methods from the Web IR domain can be adapted (and have been adapted in the last decades) for applications that address Web 2.0 problems. However, some important differences of the collaborative environments should be taken into account:

- Analogously to many other collaborative environments, Web 2.0 applications exhibit the behavior of a social network. However, the underlying evolution behavior may substantially differ from existing models for the Web scenario (e.g. preferential attachment [4] and random rewiring [32]). Therefore, these models do not necessarily provide the best fit for Web 2.0 social network characteristics, such as network diameter, characteristic path length, or clustering coefficient. Moreover, there are multiple levels of system growth with potentially different evolution patterns, e.g. micro-behavior (particular users) and macro-behavior (user groups).
- The browsing/recommendation scenario substantially differs from more common IR methods used for Web retrieval. On one hand, there are multiple dimensions (users, user groups, tags, comments, personal favorite lists) that can be used for object characterization. On the other hand, particular dimensions (e.g. annotations of the given item picture or video) tend to be extremely sparse. A suitable hybrid model for multimedia, text, and user mining may be required for constructing integrated recommender applications.
- Due to a large number of available resources, recommender applications should provide a highly efficient infrastructure for data representation and personalization. Typically, the user is not interested in obtaining a vast number of recommendations; however, the acceptance of the recommender system is crucially dependent on the quality of best recommendations in the ranked list, and on the required response time for getting these recommendations.
- Almost all Web 2.0 applications are highly dynamic environments with frequent rates of change for both content and user interests. Themes of high interest (e.g. pictures associated with some event such as natural disasters, trips, or conferences) may have a short lifespan.

In order to address these custom properties of collaborative Web 2.0 applications, the target recommender system should satisfy the following top-level requirements:

Flickr Query	Cardinality of the unordered result set
www	3472
www 2007	2297
www banff	47
www conference	34
www 2007 banff	29
www conference 2007	19
www conference banff	8
www 2007 conference banff	8

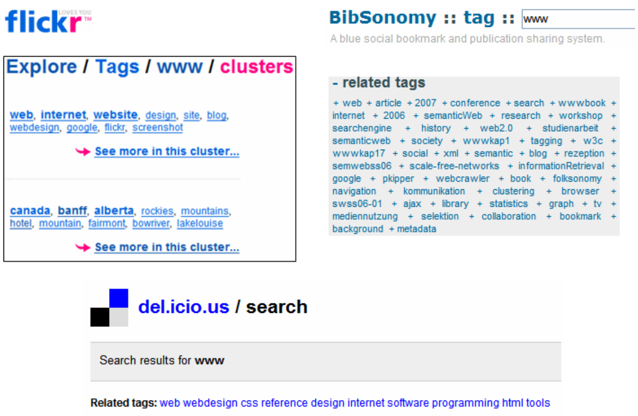
Figure 1: Samples of Querying Flickr

1. It should take into account a specialized model of dependencies between users, items, and annotations that provides a good fit for observed properties of the folksonomy.
2. The multi-dimensional model should capture various aspects of the folksonomy, including application-specific ones (e.g. favorites lists, comments, user groups, etc.) and correlations between them.
3. The system should provide at least a limited number of high-quality results with short response times, e.g. using appropriate $top - k$ retrieval algorithms in the background.
4. The system should provide mechanisms for trend detection in folksonomies and support trend-based recommendation of new content to the custom user.
5. The system should provide mechanisms of personalization, e.g. by using a multi-dimensional user-specific context that captures the “small world” of the user’s annotations and items, other related users, potentially relevant items and discussion threads with the user’s participation.

1.3 Example Scenario

Let us consider an example scenario in which a user is looking for pictures about the WWW conference 2007 in Banff, Canada. By querying Flickr for “WWW” or “WWW Banff”, he would obtain an unordered results list that contains between 2,000 and 3,000 matches (Figure 1). Since the system cannot provide a meaningful ranking for these results, finding relevant items becomes a hard needle-in-a-haystack search problem. Of course, the user could refine the conjunctive query by experimenting with more specific query formulations, e.g. by incrementally adding more and more search keywords (and thus increasing restrictivity). However, in practice, this leads to a rapid decrease in recall: the query “WWW Conference Banff 2007” returns only a few potentially relevant matches. In both described cases, finding a suitable number of potentially interesting pictures remains a difficult task.

It is clear that a suitable recommender system should aim to provide a better ratio between topic restriction and cardinality of the results set. For example, by analyzing a user’s personal profile (e.g. favorites list, participation in groups and comments) the system could suggest a community of other users with similar interests or professional background (e.g. other computer scientists that participated in the WWW 2007 conference). In the next step, the query “WWW 2007” could be executed with restricted scope on annotated items provided by these related users. Alternatively, the query could be expanded by search terms often used by the community and highly correlated with the



Explore / Tags / www / clusters

web, internet, website, design, site, blog, webdesign, google, flickr, screenshot

See more in this cluster...

canada, banff, alberta, rockies, mountains, hotel, mountain, fairmont, bowriver, lakelouise

See more in this cluster...

BibSonomy :: tag :: www

A blue social bookmark and publication sharing system.

- related tags

+ web + article + 2007 + conference + search + wwwbook + internet + 2006 + semanticWeb + research + workshop + searchengine + history + web2.0 + studienarbeit + semanticweb + society + wwwkap1 + tagging + w3c + wwwkap17 + social + xml + semantic + blog + rezeption + semwebss06 + scale-free-networks + informationRetrieval + google + pkipper + webcrawler + book + folksonomy + navigation + kommunikation + clustering + browser + swss06-01 + ajax + library + statistics + graph + tv + mediennutzung + selektion + collaboration + bookmark + background + metadata

del.icio.us / search

Search results for **www**

Related tags: web webdesign css reference design internet software programming html tools

Figure 2: Clusters of Related Keywords in Folksonomies

user's search keywords (e.g. by identifying a high correlation between the tags "WWW" and "WorldWideWeb"). To a certain extent, this functionality is supported by systems like Flickr, Bibsonomy and or del.icio.us; however, clusters of related keywords are estimated in a global setting and do not capture the personal context of a particular user (Figure 2).

Furthermore, by analyzing evolving interests of the community, the system should be able to gather items from related events (e.g. pictures from the follow-up conference WSDM 2008, announced on the WWW 2007 website), and present corresponding matches as new recommendations to the user. In the optimal case, the recommender algorithms should run as background processes without the need for human intervention or relevance feedback.

1.4 Related Work

Schmitz et al. have formalized folksonomies and discuss the use of association rule mining for analyzing and structuring them in [28]. The recent work on folksonomy-based web collaboration systems includes [12], [16], and [23] which provide good overviews of social bookmarking tools with special emphasis on folksonomies, and [**?**] which discusses strengths and limitations of folksonomies in a more general setting. In [**?**], a model of semantic-social networks for extracting lightweight ontologies from del.icio.us is defined.

The analysis of topological properties is well-known in the areas of complex networks [26; 25; 2; 11; 6] and social network analysis (SNA). Typical examples of such measures are the clustering coefficient and the characteristic path length in the tripartite undirected hypergraph: $G = (V, E)$, where $V = U \dot\cup T \dot\cup R$ is the set of nodes, and $E = \{\{u, t, r\} \mid (u, t, r) \in Y\}$ is the set of hyperedges. In other words, the hypergraph captures relationships between users, annotations, and items. An equivalent common view on folksonomy data is a quadruple $\mathbb{F} := (U, T, R, Y)$. This structure is known in Formal Concept Analysis [33; 13] as a *triadic context* [22; 29].

There are several systems working on top of del.icio.us to explore the underlying folksonomy. CollaborativeRank[4] provides ranked search results on top of del.icio.us bookmarks. The ranking takes into account how early someone bookmarked an URL and how many people followed them. Other systems show popular sites (Populicious[5]) or focus on graphical representations (Cloudalicious[6], Grafolicious[7]) of statistics about del.icio.us.

In many cases, suitable recommendations can be obtained by analyzing link-based authority measures of the folksonomy. Site ranking algorithms, for instance the PageRank algorithm [7], use topological information embedded in a directed network to infer the relative importance of nodes. Analogously, a node ranking procedure for folksonomies, the FolkRank algorithm, has been introduced in [18]. In contrast with PageRank, FolkRank also provides useful information in the case of undirected networks. Taking on a different perspective, community detection algorithms are able to detect relation similarities at a higher level. A yet different procedure is the Markov Clustering algorithm (MCL) in which a renormalization-like scheme is used in order to detect communities of nodes in weighted networks [31].

Detection of global trends in the community is an additional valuable source of information for constructing recommendations. In [3], the evolution of the relationship graphs over time is analyzed. The application of the proposed method lies in the improved detection of current real-life trends in search engines. In the future, new search methods for folksonomies should support adaptation of [3] to the Web 2.0 scenario.

Kleinberg [19] summarizes several different approaches to analyze online information streams over time. He distinguishes between three methods to detect trends: using normalized absolute change, relative change and a probabilistic model. The popularity gradient is related to the second approach, but differs insofar as it allows for the discovery of *topic-specific* trends and honours steep rises more if they occur higher in the ranking. The text mining scenario described in [19] requires focusing on words that are neither too frequent nor too infrequent.

Common recommender systems are usually used in one of two contexts: (1) to help users locate items of interest they have not previously encountered, and (2) to judge the degree of interest a user will have in item they have not yet rated. With the growing popularity of on-line shopping, E-commerce recommender systems have matured into a fundamental technology to support the dissemination of goods and services [27]. Much research has been undertaken to classify different recommendation strategies [9; 17], but here we divide them broadly into two categories: *Collaborative* and *Content-based* recommendations.

Collaborative recommendation is probably the most widely used and extensively studied technique that is founded on one simple premise: if user A is interested in items w, x, and y, and user B is interested in items w, x, y, and z, then it is likely that user A will also be interested in item z. In a collaborative recommender system, the ratings a user assigns to items is used to measure their commonality with other users who have also rated the same items. The degree of interest for an unseen item can be deduced for a particular user by examining the ratings of their neighbors. It has been recognized that a user's interest may change over time, so time-based discounting methods have been developed [5] to reflect changing interests.

Content-based recommendation represents the culmination of efforts by the information retrieval and knowledge

[4] http://collabrank.org/

[5] http://populicio.us/

[6] http://cloudalicio.us/

[7] http://www.neuroticweb.com/recursos/del.icio.us-graphs/

representation communities. A set of attributes for the items in a system is determined, such as terms and their frequencies for documents in a repository, so the system can build a profile for each user based on the attributes present in the items that a user has rated highly. The interest a user will have in an unrated item can then be deduced by calculating its similarity to their profile based on the attributes assigned to the item.

Such systems are not without their deficiencies, the most prominent of which arise when new items and new users are added to the system - commonly referred to as the *ramp-up* problem [21]. Since both content-based and collaborative recommender systems rely on ratings to build a user's profile of interest, new users with no ratings have neutral profiles. When new items are added to a collaborative recommender system, they will not be recommended until some users have rated them. Collaborative systems also depend on the overlap in ratings across users and perform badly when ratings are sparse (i.e. few users have rated the same items) because it is hard to find similar neighbors.

Hybrid recommender systems, i.e. those which make use of collaborative and content based approaches, have been developed to overcome some of these problems. For example, collaborative recommender systems do not perform well with respect to items that have not been rated, but content-based methods can be used to understand their relationship to other items. Hence, a mixture of the two approaches can be used to provide more robust systems. More recent recommender systems have also investigated the use of ontologies to represent user profiles [24]. Benefits of this approach are a more intuitive profile visualization and the discovery of interests through inferencing mechanisms.

To the best of our knowledge, this paper is the first to describe the application of recommender systems on Web 2.0 structures.

2 Recommender Systems for Web 2.0

2.1 Objects and their Relationships

The structure of a collaborative sharing framework can be considered as a tripartite network with ternary relations (assigned tags) between users $u \in U$, resources (e.g. images, media files) $r \in R$ and associated tags (in our case arbitrary text labels) $t \in T$. The set of all relations of the framework is therefore $Y \subseteq U \times T \times R$ [28]. An equivalent representation of the framework is the hypergraph $G := (V, E)$ with vertices $V = U \cup T \cup R$ and hyperedges $E \subseteq Y$.

2.2 Information Clouds

The natural way to characterize elements from U and R in a sharing framework is a collection of element-specific tags $t_i \in T, i = 1..k$. We generally define an information cloud as the tuple

$$\mathcal{T} := (Y^*, f) \tag{1}$$

where $Y^* \subseteq Y$ is a context-dependent subset of all tag relations and

$$f(t) : Y^* \rightarrow [0..1] \tag{2}$$

a *tag-rank function* that computes a score for each $t \in T^* \subseteq Y^*$.

The score can be interpreted e.g. as a relevance measure of the given tag with respect to the characterized subject (e.g. 0 = least relevant, 1 = most relevant).

The introduced notion of clouds can be directly used to characterize elements from U and R. For example, the *user-centric* cloud $\mathcal{T}_u$ (i.e. user-specific collection of tags for the user $u \in U$) and a *resource-centric* cloud $\mathcal{T}_r$ (i.e. all associated tags of a single resource $r \in R$) can be expressed as

$$\mathcal{T}_u := (Y_u, f), \quad Y_u \subseteq u \times T \times R, \tag{3}$$

$$\mathcal{T}_r := (Y_r, f), \quad Y_u \subseteq U \times T \times r, \tag{4}$$

Analogously, more complex *community-centric* clouds can be used to summarize all tags used by a group of users $U^* \subseteq U$

$$\mathcal{T}_{U^*} := (Y_{U^*}, f), \quad Y_{U^*} \subseteq U^* \times T \times R, \tag{5}$$

or for a collection of resources $R^* \subseteq R$

$$\mathcal{T}_{R^*} := (Y_{R^*}, f), \quad Y_{R^*} \subseteq U \times T \times R^*, \tag{6}$$

Finally, a combination of (5) and (6) results in a community cloud about users and resources

$$\mathcal{T}_{U^*R^*} := (Y_{U^*R^*}, f), \quad Y_{U^*R^*} \subseteq U^* \times T \times R^*, \tag{7}$$

2.3 Information Tensor

The cloud notion can be used to capture pairwise dependencies between Web 2.0 attributes like users and tags. In a more general setting, the dependencies between users, items, tags, and further interesting entities (e.g. relationships by comments, users participating in groups or personal favorites lists) can be represented as a multidimensional tensor. Formally, we write an Mth order tensor as $\chi \in \mathbb{R}^{N_1 \times .. \times N_M}$, where $N_i, i = 1..M$ is the dimensionality of ith aspect. We notice that matrix unfolding of χ (i.e. vectors in $\mathbb{R}^{N_d}$ obtained by keeping index d fixed and varying the other indices) correspond to the information clouds introduced in the previous section.

For dynamic evolution scenarios, a sequence of information tensors $\chi_1..\chi_n$ can be considered. Each tensor χ_i corresponds to the snapshot of the Web 2.0 framework at a different timepoint (e.g. by monitoring system activities over time). If n is an integer that increases with time, the sequence is called a tensor stream [30]. Both sequences and streams allow for analysis of dynamic patterns in the corresponding Web 2.0 community. For instance, methods like DTA [30] have been especially designed for finding highly correlated dimensions and monitoring them, detection of anomalies, and clustering.

By choosing suitable projections of χ, we can easily restrict the scope to particular dimensions and map the generalized model notion onto existing problem definitions from the area of recommender systems that will be discussed in the next section.

3 Proposed Recommender System Design

In this section, we will show how concepts from recommender systems can be applied to Web 2.0 applications. We will concentrate on Flickr as a prominent example; however, the proposed techniques could also be applied to any other Web 2.0 scenario. Given a large data set, the objective of a recommender system is to propose a subset of items from this data set to a user that are potentially relevant or 'interesting'. For example, in Flickr these items can be photos, groups, or other users. This leads to recommendations such as:

- Given a user, recommend photos which may be of interest.
- Given a user, recommend users they may like to contact.
- Given a user, recommend groups they may like to join.

In the remainder of this section we will first provide a formal notion of recommender systems and show how this can be applied to a scenario such as Flickr. We then discuss three approaches to tackle the recommender problem: content-based methods, collaborative methods, and hybrid methods. We will use notions based on a recent survey on recommender systems [1].

3.1 Formalizing the Problem

Consider a utility function

$$ut : U \times S \rightarrow R \tag{8}$$

, where U is a set of users, S a set of items, and R a set of relevance values (e.g. real values in $[0, 1]$).

The objective of a recommender system is to choose for a user $u \in U$ an item $s'_u \in S$ that maximizes the user's utility:

$$s'_u = argmax_{s \in S} ut(u, s) \tag{9}$$

Usually, the utility function is defined over a subspace of $U \times S$. This requires that ut must be extrapolated to the whole space $U \times S$.

In the simplest case for Flickr, U corresponds to the set of Flickr users. There are extensions and generalizations possible: U could alternatively consist of tuples $(user, photo)$, meaning a user viewing (or commenting on) a photo is provided with a list of other related photos. Since the result of this recommendation depends upon the photos, as well as the user's profile, this is an example of "personalization".

The set of items S could correspond to the set of photos (likely the most obvious option), the set of other Flickr users, the set of s groups, or tags in Flickr.

The most interesting problem is the computation of relevance values R. For classical recommender systems, relevance directly assigned by the users is available, typically in the form of a star-rating. For example, in the movie application MovieLens.org, users assign 0 stars as the lowest rating and 5 stars as the highest. In Flickr, and many other Web 2.0 applications, a direct rating is not available[8]. However, annotations supplied by users can be considered as *implicit ratings*.

For a photo the following information can be used:

- The photo belongs to the user. In this simple case we might assume that the user is interested in the photos that he has uploaded. To obtain a more fine-grained measure, the length of the textual description of the photo and the number of tags could be taken into account (the intuition behind this is that users will put more effort into the annotation of photos that are interesting to them).
- The user has marked the photo as a favorite. This is probably the most direct positive relevance assignment possible in Flickr.
- The user writes one or more comments about the photo. This implies that for the user, it was worth the effort of making a statement about the photo (whether positive or negative). More enhanced methods could take the length and date of the comment into account, and use sentiment classification to categorize the comment as positive or negative.

For assigning relevance to other users, the following clues can be considered:

- A user is on the contact list of another user. In this case, it is likely that both users share similar interests.
- A user has saved photos from another user as their favorites.
- A user has written comments on another user's photos.

Similar relevance clues can be distinguished for other items such as groups or tags.

An overall relevance function can combine these annotations (e.g. by using a weighted linear combination of evidences). It should be noted that in the way previously described, we obtain just relevance values for a subset of items already known to the respective users. In the subsequent paragraphs, we will show how we can extrapolate this and other information to recommend new items to the user.

3.2 Content-based methods

For content-based methods, the user will be recommended items similar to those preferred in the past. The simplest, and most direct approach, is to estimate the utility $ut(u, s)$ of item s for user u based on the utilities $ut(u, s_i)$ assigned by user u to items s_i that are 'similar' to s. Formally, given a content representation $Content(s)$ and a content-based profile $ContentBasedProfile(u)$ of a user u, the utility function is usually defined as:

$$ut(u, s) = score(ContentBasedProfile(u), Content(s)) \tag{10}$$

where the $score$ function should produce high relevance values if $ContentBasedProfile(u)$ is related to $Content(s)$.

In Flickr, one approach is to represent both content and user profile as feature vectors (e.g. using a classical IR vector-space model). In this approach, the features can be weighted to vary their "importance". A common term-weighting strategy in IR is known as $tf * idf$ where tf denotes the frequency of a term and idf the inverse document frequency (i.e. the number of documents a term appears in). The intution behind this weighting is that a term is considered more important (i.e. more discriminative) if it occurs more frequently in fewer documents.

Using this approach, in Flickr photos can be represented as:

- Computing a $tf * idf$ vector from the set of tags associated with the photo.
- Computing a $tf * idf$ vector from the textual description of the photo.
- Computing a group vector with each dimension corresponding to the group the photo belongs to.

Possible representations of a user-profile can be obtained by:

- Computing a $tf * idf$ vector of the user description in his profile

[8] For YouTube a star-rating is available for the videos but not for other items such as users, groups and tags

- Computing the average of the user's photos vectors
- Computing a vector representing the groups of the user
- Computing the average of the photos from other users assigned as favorites or commented by this user

The first two options for describing user content can be used for the content-based profile as well as for the content of the user (if itemset S consists of users).

Given a vector representation $\vec{u}$ of $ContentBasedProfile(u)$ and $\vec{s}$ of $Content(s)$, the cosine measure can be used as a *scoring* function (or similarity measure) to obtain:

$$ut(u,s) = cos(\vec{u},\vec{s}) = \frac{\vec{u} \cdot \vec{s}}{||\vec{u}|| \cdot ||\vec{s}||} \tag{11}$$

Machine Learning Approach Alternatively, relevance assignment can be formulated as a machine learning problem: given a set of items S_{pos} (represented as feature vectors as described above) relevant to the user, and S_{neg} that are not relevant to the user, train a binary classifier (with the two classes "relevant for the user" and "not relevant for the user") on these instances. Based on the trained model, it is then possible to estimate the relevance of new items. For Flickr, S_{pos} can be obtained using the user annotations (favorites, comments, contacts, etc.) as described in Section 3.1.

For example, linear support vector machines (SVMs) construct a hyperplane $\vec{w} \cdot \vec{x} + b = 0$ separating the set of positive training examples from a set of negative examples with maximum margin δ. For a new previously unseen, item $\vec{d}$, the SVM simply tests whether the item lays on the "positive" side or the "negative" side of the separating hyperplane. In addition, the distances of the test items from the hyperplane can be interpreted as classification confidences.

3.3 Collaborative Methods

In *collaborative recommender systems*, also coined *collaborative filtering systems*, the user is recommended items that people with similar preferences have liked in the past. Formally, the utility $ut(u,s)$ of item s and user u is estimated based on the utilities $ut(u_j,s)$ assigned to item s by those users $u_j \in U$ who are similar to user u. The value of an unknown rating $ut(u,s)$ is usually computed as an aggregate of the ratings of other users (e.g. the N most similar) for item s:

$$ut(u,s) = aggr_{u' \in U'} ut(u',s) \tag{12}$$

, where U' is the set of N users most similar to u. Examples for aggregations given in [1] are average sum or weighted sum (weighted by the user similarities).

In section 3.2 we described several ways to obtain vector respresentations $\vec{u}$ of users u. Using a similarity measure such as the cosine measure for pairs of users, we can compute the N most similar users. The relevance assignment $ut(u',s)$ can be obtained using implicit ratings of other users described in Section 3.1.

Alternatively, we can take information from the social networks implicitly contained in Flickr into account to find similar or related users. Formally, these networks can be described, for instance, by the following graphs:

- *Contact graph* $G_{contact}(U,V)$ with $(u_1,u_2) \in V$ iff user u_2 is in the contact list of user u_1.
- *Comment graph* $G_{comment}(U,V)$ with $(u_1,u_2) \in V$ iff user u_1 has written a comment on a photo of user u_2.
- *Favorites graph* $G_{favorites}(U,V)$ with $(u_1,u_2) \in V$ iff user u_1 has assigned a photo of user u_2 as favorite.
- *Group graph* $G_{group}(U,V)$ with $(u_1,u_2) \in V$ iff user u_1 and user u_2 are are members of the same group.

Possible extensions are weighted graphs, taking e.g. the number of comments or favorites in $G_{comment}$ or $G_{favorites}$ into account or normalizing the weights in $G_{contact}$ by the overall number of contacts. Furthermore, we can consider the combination of graphs, computing, e.g. the union of edge-sets of distinct graphs.

We can find related users by traversing the social network graphs. For a user u we can, e.g., consider all users that are connected by a path of length $\leq k$, where k is parameter to be determined. How to tune the parameters, which graphs to choose for the search and how to combine the results are open research questions.

3.4 Hybrid Methods

For recommender systems, the sparsity of the data can be a serious problem [1]. For instance, in the collaborative approach described in the previous Section 3.3, for many items, there might be no implicit user-feedback available from the set of similar users. To overcome this problem, hybrid methods can be applied by incorporating content-based relevance assignments to obtain utility values $ut(u',s)$. Alternatively, collaborative and content-based methods can be first run separately, and then their predictions combined. For a more exhaustive review of hybrid methods see [1].

4 Evaluation Strategies

In the previous section, we have proposed various methods for representing objects in folksonomies, using annotations and implicit information, and recommender system design. To show the viability of our approaches, besides a deeper theoretical analysis and a statistical analysis of the data, a thorough experimental evaluation of quality of recommendations is necessary.

Evaluating recommendations in Web 2.0 applications is a difficult task for several reasons. First, the absence of established reference datasets with large amounts of manually verified and labeled recommendations may require comprehensive user studies with relevance feedback. This makes reliable and reproducible large-scale evaluation very hard and time-consuming. Second, there is a significant challenge in deciding what combination of measures should better characterize the recommender quality in a comparative evaluation. Ideally, the evaluation should be objective in reflecting the quality of recommendations with respect to realistic user needs (and be orthogonal to the functionality of the underlying method). For instance, in our previously introduced application scenario, we may measure the ability of the algorithm to reconstruct "hidden" annotations of pictures (i.e. existing annotations that have been removed before applying query expansion). In this case, we would confirm the basic functionality of the proposed method, which however cannot be considered as objective proof of recommender quality.

In this section we describe two possible evaluation approaches: evaluation based on user studies and evaluation using implicit additional user information that can be directly inferred from Web 2.0 sources.

4.1 Manual Evaluation and User studies

High recommendation accuracy alone does not necessarily provide users with an effective and satisfying experience. A good recommender framework should also provide good usefulness [17]. A system that always recommends only highly popular items is probably not helpful in finding interesting "hidden" dependencies between communities, especially for a Web 2.0 environment. For example, a recommender system might tend to suggest conference pictures from other participants of to a WWW 2007 participant. Basically, this recommendation is highly accurate but rather obvious; recommended matches can be accessed by the user directly (e.g. by visiting the "WWW 2007" user group) without recommender assistance. Much more valuable would be a recommendation of images from press releases, photos from associated events like PhD workshops, or pictures of the Banff city center that are rather weakly related to the core WWW 2007 conference. Therefore, we ideally need new dimensions for analyzing recommender systems that consider both the "correctness" and the "non-obviousness" of the recommendations. Suitable dimensions are novelty and serendipity, which have been previously addressed in IR literature.

The subjective evaluation of "correctness" and "novelty" can be achieved through systematic user studies with aposteriori verification. In this evaluation scenario, recommendations of new items can be presented to multiple real users with different profiles of interest. The top-k part of the recommendation list (e.g. $k = 10$) is fully evaluated by each user by assigning scores for aspects such as "correct" and "interesting" (e.g. integer values between $1..5$).

Conceptually, manual inspection of the result set would provide the best evidence for evaluation. However, this requires comprehensive human experiments, and thus is often not scalable for large Web 2.0 platforms like Flickr or Del.icio.us.

4.2 Using Implicit Information for Evaluation

An alternative approach to approximate IR-style quality measures is the apriori method with an (estimated) gold standard. Metrics such as accuracy can be constructed by predicting the k items for which the relevance (or irrelevance) is known. A suitable approximation could be achieved by using individual favorite lists and comments, which can be considered as an indication of relevance. The recommender method should be constructed in such a way that these dimensions remain 'invisible' for the recommendation model; in other words, these dimensions must be artificially removed from the user-specific information tensor χ. An alternative is to keep these dimensions for a training set and evaluate the recommender system on a disjoint test set. The ability of the method to reconstruct favorite/comment lists (measured by the overlap between the top-k recommended items and the user-specific collection of such references, plus normalization in order to make estimates for different users comparable) can be treated as an accuracy measure.

A drawback of this methodology, however, is the absence of *negative* test samples. In fact, by using explicitly given comments and favorites we indirectly claim *all* remaining items as irrelevant, which is not entirely correct. Moreover, in practice (and in the manual evaluation scenario explained above) we are interested in finding *additional* relevant items beyond known favorites and/or comments. In a better experimental setting, the recommender method could be provided with a set of explicitly known "positive" and "negative" samples, whereby the "negative" collection could be collected through additional user studies. The ability of the method to correctly recognize "true positives" and to prioritize them in the top-k result set may provide a better accuracy estimate.

To this end, we assume that a combination of both manual user assignments and use of implicit information gathered from the folksonomy will provide the most comprehensive quality estimate.

5 Conclusion and Future Work

In this paper, we have discussed a design methodology for recommender systems in Web 2.0 applications. We have stated specific top-level requirements for recommender systems and ways of addressing them. The core representational model of our methodology captures multi-dimensional dependencies between users, items, and annotations in form of a multi-dimensional tensor. By choosing suitable projections we restrict the scope to particular dimensions of interest and can map the tasks to existing problem definitions from the area of recommender systems.

An important building block in the system design is the evaluation methodology. In this paper, we discussed pros and contras of possible evaluation strategies (apriori/aposteriori evaluation, manual result inspection vs. automated IR-style measurements) and identified the advantages of an integrated approach.

The results presented here can be summarized as a preliminary system design for Web 2.0 recommender infrastructures that will be refined and systematically evaluated in our future work. Our long-term objective is the design of scalable and reliable assistance methods that individually guide particular users through large-scale multi-dimensional Web 2.0 frameworks towards promising search results.

Acknowledgements

This work has been partially supported by the European project ”Semiotic Dynamics in Online Social Communities” (Tagora, FP6-2005-34721) and the EU-funded project Memoir.

References

[1] G. Adomavicius and A. Tuzhilin. Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[2] R. Albert and A. Barabasi. Statistical mechanics of complex networks. *Review of Modern Physics*, 74:47–97, 2001.

[3] Einat Amitay, David Carmel, Michael Herscovici, Ronny Lempel, and Aya Soffer. Trend detection through temporal link analysis. *J. Am. Soc. Inf. Sci. Technol.*, 55(14):1270–1281, 2004.

[4] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[5] D. Billsus and M. Pazzani. User Modeling for Adaptive News Access. 10(2-3):147–180, 2000.

[6] K. Borner, Soma Sanyal, and A. Vespignani. Network science: a theoretical and practical framework. *Annual Review of Information Science and Technology*, 41:537–607, 2007.

[7] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[8] M. Buckland. Emmanuel Goldberg, Electronic Document Retrieval, and Vannevar Bush's Memex. *JASIS*, 43(4):284–294, 1992.

[9] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[10] V. Bush. As We May Think. *The Atlantic Monthly*, 176(1):101 – 108, 1945.

[11] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of Networks: From Biological Nets to the Internet and WWW (Physics)*. Oxford University Press, Inc., New York, NY, USA, 2003.

[12] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. In *Proc. 15th Int. WWW Conference*, May 2006.

[13] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical foundations*. Springer, 1999.

[14] J. Gemmell, G. Bell, and R. Lueder. MyLifeBits: a personal database for everything. *Commun. ACM*, 49(1):88–95, 2006.

[15] Stephen Haag, Maeve Cummings, and Donald J. McCubbrey. *Management information systems for the information age*. Irwin McGraw-Hill, 3 edition, 2002.

[16] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine*, 11(4), April 2005.

[17] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[18] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information Retrieval in Folksonomies: Search and Ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, June 2006. Springer.

[19] J. Kleinberg. Temporal dynamics of on-line information streams. In M. Garofalakis, J. Gehrke, and R. Rastogi, editors, *Data Stream Management: Processing High-Speed Data Streams*. Springer, 2006.

[20] S. L. Kogan and M. J. Muller. Ethnographic study of collaborative knowledge work. *IBM Systems Journal*, 45(4):759, 2006.

[21] J. A. Konstan, J. Reidl, A. Borchers, and J.L. Herlocker. Recommender systems: A grouplens perspective. In *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08)*, pages 60–64. AAAI Press, 1998.

[22] F. Lehmann and R. Wille. A triadic approach to formal concept analysis. In G. Ellis, R. Levinson, W. Rich, and J. F. Sowa, editors, *Conceptual Structures: Applications, Implementation and Theory*, volume 954 of *Lecture Notes in Computer Science*. Springer, 1995.

[23] B. Lund, T. Hammond, M. Flack, and T. Hannay. Social Bookmarking Tools (II): A Case Study - Connotea. *D-Lib Magazine*, 11(4), 2005.

[24] S. Middleton, N. Shadbolt, and D. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.

[25] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167, 2003.

[26] Romualdo Pastor-Satorras and Alessandro Vespignani. *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, New York, NY, USA, 2004.

[27] Ben J. Schafer, Joseph A. Konstan, and John Riedi. Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166, 1999.

[28] C. Schmitz, A. Hotho, R. Jaeschke, and G. Stumme. Mining Association Rules in Folksonomies. pages 261–270, 2006.

[29] Gerd Stumme. A finite state model for on-line analytical processing in triadic contexts. In *ICFCA*, pages 315–328, 2005.

[30] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Philadelphia,USA*, pages 374–383, 2006.

[31] S. van Dongen. A cluster algorithm for graphs. *National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, Technical Report INS-R0010*, 2000.

[32] D. J. Watts. *Small-worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, Princeton, NJ (USA), 1999.

[33] R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. Reidel, Dordrecht-Boston, 1982.

CheckMATE – Erfahrungsmanagement für ServiceRoboter zur Realisierung von Self-Healing in Produktionsanlagen

Markus Nick, Sören Schneickert
Fraunhofer Institute for Experimental Software Engineering (IESE),
67663 Kaiserslautern,
{markus.nick, soeren.schneickert}@iese.fraunhofer.de

Jürgen Grotepaß[1], Ingo Heine[2]
[1]Freudenberg FDS GmbH & Co KG, 69469 Weinheim
[2]Freudenberg FAW, 69514 Laudenbach
{Juergen.Grotepass, Ingo.Heine}@Freudenberg.de

Abstract

In der produzierenden Industrie – insbesondere in der Pharma-, Automobil- und Zulieferindustrie – steigen die Qualitätsanforderungen in Richtung Null-Fehler-Produktion. Als Antwort kommen zunehmend automatische Verfahren zur Fehlerbehebung und Fehlervermeidung in der Produktionskette zum Einsatz. Integrierte Qualitäts- und Prozessdaten von Sensorsystemen und Anlagensteuerungen stellen hierfür die Basis dar (Projekt BridgeIT[1]). Im Projekt CheckMATE[2] soll nunmehr Beziehungen von prozessbegleitend gewonnenen Qualitätsdaten mit Erfahrungswissen hergestellt werden, so dass kleinste Qualitätsregelkreise konfiguriert werden können. Dazu werden aus Defekttrends (Defekterfahrungen) Steuerparameter für einen lernfähigen Service-Roboter ermittelt, der Prozesslageabweichungen im Prozess automatisch behebt. Hiermit wird ein Self-Healing-Verfahren zur Kompensation von Prozessdrifts umgesetzt. Prozessdrifts sind allmähliche Prozessverschiebungen vom Zielwert und können bis zu 1.5 sigma betragen, wodurch sich die Wahrscheinlichkeit für Defekte an den produzierten Teilen erhöht. Während der Evaluierungsphase des Prototypen im industriellen Pilotprozess wurden innerhalb von 2 Monaten mehr als 80 Defekterfahrungen akquiriert, die 2007 im statistischen Dauertest innerhalb der Fertigungsumgebung verifiziert werden.

References

Nick, M.; Schneickert, S.; Grotepaß, J.; Heine, I. *CheckMATE – Erfahrungsmanagement für ServiceRoboter zur Realisierung von Self-Healing in Produktionsanlagen*. KI-Themenheft Erfahrungsmanagement, 2007

[1] Das BridgeIT-Projekt (http://www.BridgeIT.de/) wurde vom Bundesminsterium für Bildung und Forschung gefördert.

[2] Das CheckMATE Projekt (http://www.checkmate-online.de/) wird vom Bundesminsterium für Bildung und Forschung gefördert.

Experience Management

Markus Nick[1], Klaus-Dieter Althoff[2], Ralph Bergmann[3]
[1] Fraunhofer IESE, Fraunhoferplatz 1, 67663 Kaiserslautern
[2] University of Hildesheim, Intelligent Information Systems
[3] University of Trier, Department of Business Information Systems II

Extended Abstract

Experience management has a high relevance for the industry as recent studies between 2001 and 2005 show. In a study of the Fraunhofer Gesellschaft, experience management was the top item among the challenges regarding knowledge and information. "Experience base" was the top item for planned usage/installation among the IT support for knowledge management. Core technologies for realizing experience management systems that come from the field of artificial intelligence are related to decision making, knowledge acquisition and extraction tasks - e.g. case-based reasoning, ontologies, machine learning, and natural language processing.

The paper [Nick *et al.*, 2007] defines the terms experience, experience management system and Experience Management and gives an overview on contributions from AI to Experience Management and future challenges:

Experience is knowledge or practical wisdom gained from what one has observed, encountered, or undergone (Webster's Dictionary). Experience is concerned with what was true or false, correct or incorrect, good or bad, more or less useful. This means that experience has a certain validity that is bound to the contexts/situations where it occurred.

An *Experience Management System (EMS)* is a socio-technical system that is established for managing, reusing, and recording experiences among its "'users"'. Experiences can be recorded and reused using a software system, which is operated by people. Usually, these people also do not or cannot make all aspects of their experience explicit for various reasons. So, assuring that the relevant experiences are recorded and reused requires further organizational measures.

Object of investigation of the field *Experience Management (EM)* are EMSs and their integration into business processes. EM considers all relevant processes for build-up, operation, use, maintenance, evaluation, improvement, and management of EMSs. EM also includes the organizational and social measures that foster the acceptance and the continuous use of the system by its users. As a research field, EM looks at the methods and technologies that are suitable for collecting experiences from various sources (documents, data, experts, etc.), recording/packaging, reusing, adapting, and maintaining experiences - including the respective organizational and social measures. Thus, EM is a special form of knowledge management.

An *EMS software* supports a certain set of operations related to reuse, adaptation/modification, and recording of experiences and capturing feedback. For these operations, their counterparts in the business processes have to be identified during the build-up of an EMS.

Various AI fields contribute to EM: Case-based reasoning (CBR), ontologies, machine learning, etc.

Future challenges for EM are Pervasive Computing, Ambient Intelligence, and Web 2.0. We expect self-learning to be a major feature of future EMSs.

References

[Nick *et al.*, 2007] Markus Nick, Klaus-Dieter Althoff, and Ralph Bergmann. Experience management. *Künstliche Intelligenz*, 2, 2007.

Using Knowledge Wikis to Support Scientific Communities

Joachim Baumeister[1,*] **, Jochen Reutelshoefer**[1] and **Karin Nadrowsk**[2]**, and Axel Misok**[2]

1) Institute of Computer Science, University of Würzburg, Germany

2) Institute of Animal Ecology, University of Giessen, Germany

Abstract

With the success of numerous applications of the Web 2.0 the interest in a web–based support of scientific communities has also gained significant relevance. The concept of Wikis, one building block of the Web 2.0, has shown to be a reasonable infrastructure for sharing and refining any kind of knowledge. The most prominent example is the encyclopedia Wikipedia, but many smaller wiki applications have proved to be beneficial, e.g., the availability of experience management and documentation projects in open–source communities and large enterprises.

In this paper, we introduce *knowledge wikis* that extend the features of a regular wiki by the representation and use of explicit problem–solving knowledge. We motivate the general ideas and benefits of knowledge wikis, and we describe a concrete knowledge wiki implementation. The actual use of a knowledge wiki is demonstrated by two case studies: the first case study reports on the share and reuse of ecological domain knowledge in the context of the BIOLOG project; the second case study describes a knowledge formalization pattern repository taken as a demonstrator from the knowledge engineering domain.

Bibliographical Note

The paper was initially presented at the *1st Workshop on Scientific Communities of Practice (SCOOP)*, Bremen, Germany, 2007
http://www.kwarc.info/events/workshops.html#scoop

Representation and Structure-based Similarity Assessment for Agile Workflows

Mirjam Minor, Alexander Tartakovski, and Ralph Bergmann
Lehrstuhl fuer Wirtschaftsinformatik II, Universität Trier, Germany

Abstract

The increasing dynamics of today's work impacts the business processes. Agile workflow technology is a means for the automation of adaptable processes. However, the modification of workflows is a difficult task that is performed by human experts. This paper discusses the novel approach of agile workflow technology for dynamic, long-term scenarios and on change reuse. First, it introduces new concepts for a workflow modelling language, which enables an interlocked modelling and execution of workflows. Second, it provides new process-oriented methods of case-based reasoning for the reuse of change experience. The methods include the representation and index-based retrieval of past workflows in order to give authoring support for adaptation of recent workflow instances. The results from an experimental evaluation in a real-world scenario highlight the usefulness and the practical impact of this work.

Keywords: Agile Workflow, Workflow Modelling Language, Adaptation of Workflows, Case-Based Adaptation, Workflow Similarity Measure

Bibliographical Note

The original paper has been published as follows:
Mirjam Minor, Alexander Tartakovski, and Ralph Bergmann. *Representation and Structure-based Similarity Assessment for Agile Workflows.* In Rosina O. Weber, and Michael M. Richter, editors, /Case-Based Reasoning Research and Development, 7th International Conference on Case-Based Reasoning, ICCBR 2007, Belfast, Northern Ireland, UK, August 2007, Proceedings/, LNCS 4626, pages 224 - 238, 2007, Springer.

A more detailed description can be found in:
Mirjam Minor, Alexander Tartakovski, Daniel Schmalen, and Ralph Bergmann. *Agile Workflow Technology and Case-Based Change Reuse for Long-Term Processes.* International Journal of Intelligent Information Technologies, accepted for publication, 2007.

15th Workshop on Adaptivity and User Modeling in Interactive Systems

Ingo Brunkhorst
L3S Research Center
Hannover, Germany
brunkhorst@L3S.de

Daniel Krause
Leibniz University Hannover
Hannover, Germany
krause@kbs.uni-hannover.de

Wassiou Sitou
Technische Universität München
Munich, Germany
sitou@in.tum.de

1 The ABIS Workshop

ABIS - *'Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen'* - is the special interest group on Adaptivity and User Modeling in Interactive Systems of the German Computer Society.

The ABIS Workshop has been established as a highly interactive forum for discussing the state of the art in personalization and user modeling. Latest developments in industry and research are presented in plenary sessions, forums and tutorials to discuss trends and experiences.

2 'Modeling Users and Adaptive Behavior'

Nowadays adaptation is seen as the most promising approach to increase the usability of complex modern software. In the recent years, various adaptation techniques have been introduced and seem to work well for specific application contexts.

To enable re-use and comparison of the different techniques, there is a strong need, not only for domain independent adaptation techniques, but also for formal description of the involved models.

Adaptive systems cannot be considered without the user's context, as they interact with and work for the user. To enhance the quality of adaptation, every adaptive system requires a very precise knowledge of the user. This knowledge includes the behavior of the user, e.g. by observing the interactions of the user with the system, to discover the context and the interest of the user.

Therefore, all adaptive systems can benefit from a comprehensive model that describes the behavior of a user, and other relevant information about her.

3 Workshop Overview

The program commitee received submissions from research and industry within the broad area of User Modeling and Adaptive Systems. Special emphasis of this year's workshop was on submissions in the following areas:

User models for adaptive systems

- Modeling behavior of users
- Modeling user's context
- Reasoning on user models
- Interpretation of user behavior and user feedback
- Machine learning for user modeling

Behavior of adaptive systems

- Context aware adaptive systems
- Adaptive navigation support
- Adaptive presentation
- Adaptive orientation support
- Evaluation of adaptive systems

4 Program Chairs

- Ingo Brunkhorst (L3S Research Center, Germany)
- Daniel Krause (Leibniz University Hannover, Germany)
- Wassiou Sitou (Technische Universität München, Germany)

5 Program Committee

- Armen Aghasaryan (Alcatel-Lucent, France)
- Mathias Bauer (DFKI GmbH, Germany)
- Susanne Boll (Die Carl von Ossietzky Universität Oldenburg, Germany)
- Manfred Broy (Technische Universität München, Germany)
- Michael Fahrmair (DoCoMo Euro-Labs, Germany)
- Sabine Graf (TU Wien, Austria)
- Dominik Heckmann (DFKI GmbH, Germany)
- Nicola Henze (Leibniz University Hannover, Germany)
- Eelco Herder (L3S Research Center, Germany)
- Hagen Höpfner (International University in Bruchsal, Germany)
- Bhaskar Mehta (L3S Research Center, Germany)
- Stephan Weibelzahl (National College of Ireland, Ireland)

Taking the Teacher's Perspective for User Modeling in Complex Domains

Christian P. Janssen & Hedderik van Rijn
Department of Artificial Intelligence, University of Groningen
Grote Kruisstraat 2/1, 9712 TS Groningen, The Netherlands
cjanssen@ai.rug.nl, D.H.van.Rijn@rug.nl

Abstract

Serious games that should adapt training to the individual might benefit from methods that are developed for intelligent tutoring systems. One method, model tracing, might be used for domains with a strict hierarchy, while complex domains that lack such a hierarchy might use the method we introduce in this paper as teacher modeling. It takes the perspective of a teacher, who does not always know what leads a student to an answer, but who does know when the answer is (in-) correct and who can assess the capacities of the student over time. The assessment of this tutoring system is based on the amount of training and the amount of positive and negative completed exercises of identified training categories. Principles from a cognitive architecture are used to keep close to aspects of human learning, namely frequency and recency of usage. Simulation runs demonstrate a successful adaptation of training: exercises of categories with which students have most difficulties are presented most.

1 Introduction

Recently it has been advocated that games might be a good training environment, for example in [Aldrich, 2005; Gee, 2003; Michael and Chen, 2005]. These games are referred to as serious games. Serious games are developed for several domains, ranging from high school mathematics to social skills. Over two hundred examples can be found on the website of social impact games [Prensky, 2007]. The new research field of serious games might benefit from techniques that are developed in other fields, and focus on information personalization and digital learning: user modeling and intelligent tutoring systems.

This paper will give an introduction to intelligent tutoring systems, in the context of serious games. Two methods will be explained, that might be used for different domains. If the training domain has a hierarchical structure, the intelligent tutoring method of model tracing might be used. On the contrast, if such a hierarchical structure is missing, other methods have to be found. We will therefore introduce an alternative method that we call *teacher modeling*. We will motivate why such a method might be beneficial for games, and then outline the mechanism and the role that principles from a cognitive architecture play in the method. Our discussion will end with a set of simulations that illustrate that teacher modeling successfully adapts its trainings to the performance of the individual: the more difficulties a student has with completing exercises of a category correct, the more exercises of this category are selected for future training.

In this paper we will often refer to students and teachers, but we do not want to restrict our discussion to them. Therefore one can consider "student" as referring to any person who is learning something, for example a pupil, trainee, or a patient. Additionally, "teacher" can be considered as referring to any person who trains the student, for example an expert, a trainer, or a therapist.

2 Tutoring systems

Intelligent tutoring systems can be defined as computational instruction systems that *adapt* instruction dynamically to the learner, ideally both in form of instruction and in content [Ohlsson, 1986]. Being adaptive has as an advantage that the student can be presented with exercises that best fit individual challenges. Or, to state it differently, the student does not have to spend time on topics that he or she already has mastered.

Several steps can be distinguished in intelligent tutoring systems. In Figure 1, we illustrate a skeleton model of intelligent tutoring systems, to which additional components (e.g., a data-base with exercises, or a natural language generator) could be added for specific systems. A basic tutoring system, as illustrated, starts with a student working on an exercise. The behavior parser continually tracks the behavior of the student, and this information is used to update the view that the system has about a student's skills and performance. For example, eye gaze could be used to derive what item is currently attended, and what items are ignored [Anderson and Gluck, 2001].

The knowledge from the behavior parser can be added to the user model. This is an abstract model that keeps track of some specific (mental) aspects of the user [Johnson and Taatgen, 2005]. The model might contain general characteristics of the student, for example "likes math, but not linguistics", but might also contain more detailed knowledge, for example "makes mistakes in math additions where one of the addends is greater than ten".

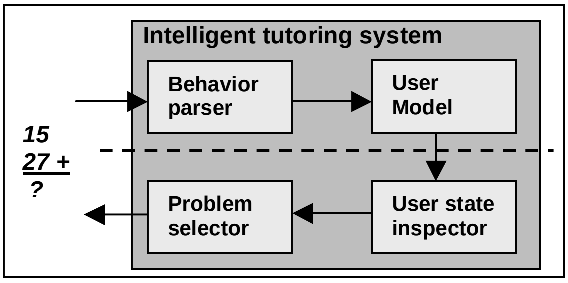


Figure 1: A skeleton model of intelligent tutoring systems.

Note that the information that the user model contains about the student might be more than the student herself is explicitly aware of, as this information might be implicitly and indirectly derived as an *interpretation* of her behavior.

The user model can be observed by the user state inspector to get a general impression about the *type* of exercises the training should focus on. Given this impression, a specific exercise can be selected by the problem selector and given to the student. Observations of the student's performance on this exercise can then be used to update the user model and to keep on selecting new exercises, until the course objectives are completed.

3 Model tracing

3.1 Outline of model tracing

One method for assessing the capabilities of users of intelligent tutoring systems is model tracing. This method tries to assess the mental processes that might have lead to a certain reaction of the student in a detailed manner. To be able to do this, a large set of possible mental processes that can be used in the task at hand are incorporated. These involve both correct rules and facts (e.g., the fact "3+4=7") and incorrect rules and facts (e.g., "3+4=8"). Based on the student's answer to an exercise, the *model* can be *traced* to identify mental processes that might have lead to that answer [Anderson and Gluck, 2001].

Model tracing requires a theory of cognition and learning, and a detailed understanding of the training domain at hand. This domain should be decomposable into a set of components, and the overall learning should be explainable by the learning of those individual components [Anderson and Gluck, 2001]. Because of this hierarchical structure, training can also be hierarchical structured: first the basic levels are trained, and once these are learned correct, more advanced levels can be trained.

An example domain for model tracing is mathematics [Anderson and Gluck, 2001]. If a student has to solve 315 + 216 and answers 521, the user model can be traced to find the possible mistake. In this case, the carry rule that the ten from 5 + 6 = 10 + 1 carries over to the addition of 10 + 10 is probably not used. A tutoring system might thus decide to start focusing on this specific rule.

In order to use model tracing, the decomposition of the domain, and the coupling to possible mental states should be possible and uncontroversial. Incorrect decomposition might lead to incorrect models and therefore to incorrect tracing of performance, and incorrect training adaptation.

3.2 Challenges of model tracing in serious games

Implementing model tracing in serious games has challenges, as games are rich environments in which a player often has a broad set of interaction possibilities. It might be difficult to map each specific action to a specific mental process. For example, in a role-playing game a player might talk to a non-player character because he wants specific information from that character, but he might also talk to the character merely to gather general intelligence.

To restrict the challenge of correctly mapping all actions to mental processes, assumptions and simplifications can be made on the mapping, with the risk of making incorrect assumptions that lead to errors in the interpretation of a player's behavior. Alternatively, the amount of interaction possibilities that the player has in the game could also be restricted. But this is at odds with both the intended flexibility of intelligent tutoring systems to adapt to an individual's needs [Ohlsson, 1986], and with principles of game design that a player should be able to play a game in several ways, thereby developing their own strategies and game-play [see discussion of the multiple routes principle in Gee, 2003].

Despite this challenge, we should not give up on games as training mechanisms, as they might have several training advantages, depending on the exact implementation of the game. Gee [2003] mentions amongst others: (1) young people are used to playing games, which makes it a natural medium for them to use, (2) the fun might encourage longer learning, and (3) players can learn by doing, instead of learning theory outside of its context.

We will therefore now introduce a method that can be used in a game setting and in which the training can be adapted to the individual without using model tracing. As this method does not require an exact mapping between single game actions and mental states, it can be applied in domains that do not have such a detailed theory, for example in depression prevention [Janssen *et al.*, 2007].

4 Teacher modeling

4.1 Motivation for teacher modeling

The method that we will outline is based on a general assessment of the capacities of the student. Educational researchers such as Shepard [2000] have identified and motivated the critical role assessments can play in normal (classroom) learning settings: a good assessment can give an indication how instruction should be adapted to the level of the individual. This conclusion can also be extended to the domains of e-learning and serious games, and therefore assessments play a central role in the method that we outline here. The method can be considered as looking at the student through the eyes of a teacher, as we try to model aspects of the assessments of the teacher: although a teacher does not always explicitly know *what* a student is thinking in order to derive an answer, the teacher does know *when* a student is making a mistake, and can make an *assessment over time* that states what skills the student often performs correct or incorrect. Moreover, the teacher continuously needs *observations of the performance* of a student, to prevent forgetting at what level the student performs. Because of these similarities with real teachers, we call our method teacher modeling.

So what information do we need to develop a system with teacher modeling? First of all, the designer of the system and a group of domain experts should define a set of training objectives. As our method is to be applied in cases were more fine-grained methods such as model tracing are inapplicable, these categories will often be very broad. Examples are "has said no in an assertive way" [in case of a depression prevention tool, Janssen *et al.*, 2007], and "has correctly answered a question about the content of chapter 10" (in a textbook tutor). We will call these categories training dimensions from now on.

For each of these training dimensions, several exercises should be available. Exercises should have multiple outcomes, leaving the student with a choice, but can take several forms, for example dialogues or multiple-choice questions. Experts should be able to indicate for each outcome, and each dimension whether the student's choice (leading to the outcome) was positive (or correct), negative (or incorrect), or not at all focusing on an aspect of that dimension (thus, training aspects of other dimensions).

This information can be stored in the behavior parser (see Figure 1). Using this information about how to interpret students' answers, the user state inspector can update its assessment of the student, this way acting as a teacher. The assessment is based on three types of information for each dimension: the amount of completed exercises, the amount of positive completed exercises and the amount of negative completed exercises. The next section will explain how this information is stored and how the assessment of the virtual teacher is made.

4.2 Using principles from a cognitive architecture

Although the information of a student's behavior could perhaps be stored in different formats (for example, an absolute counter), we use principles from the cognitive architecture ACT-R [Anderson *et al.*, 2004; Anderson and Lebiere, 1998]. Using principles from a cognitive architecture is advantageous, as this keeps us close to aspects of human cognition and learning.

We use chunks to store information on performance for each dimension. Chunks are the building blocks of ACT-R and maintain an activation level, that represents its usefulness in the past, and that is updated using the base-level learning equation [Anderson *et al.*, 2004; Anderson and Lebiere, 1998]:

$$B_i = \ln\left(\sum_{j=1}^{n} t^{-d}\right) \qquad \text{(Equation 1)}$$

In this equation t_j indicates the time since the *j*th practice of an item, and *d* indicates a time-based decay parameter (set to 0.5 by default). The formula has been shown to be good at modeling effects of recency and frequency of usage [Anderson and Schooler, 1991], and, in a slightly modified version, at predicting correct spacing of training [Pavlik and Anderson, 2005]. Being able to incorporate such aspects of learning in the system is beneficial, as our system is used in an environment where people learn.

Each dimension is modeled using three chunks. The "amount of training chunk" reflects the total amount of training, and is updated each time a player has completed an exercise of the dimension of that chunk. The second and third chunk are the "positive chunk" and the "negative chunk" which are updated each time that the student completes an exercise in respectively a positive (or correct) or a negative (or incorrect) way for that dimension.

The score that is indicated by the behavior parser after completion of an exercise is used to decide *if* the activation value of the chunks should update. However, as can be seen in Figure 2, the size with which the activation value increases depends on the amount of previous updates: the more recent and frequent chunks were updated, the smaller the increase of activation value at each update. If chunks are not updated due to completing an exercise of that dimension at a time stamp, their activation value decreases, as it is subject to decay. In our metaphor of the observing teacher that estimates the success of a student, decay can be interpreted as the teacher becoming less confident that a student performs good or bad in a certain dimension, if no new training observations are made.

Interpreting an activation score as a probability, as we did informally for our teacher metaphor, can be formally expressed using the ACT-R retrieval probability equation [Anderson *et al.*, 2004; Anderson and Lebiere, 1998]:

$$P_i = \frac{1}{1+e^{-(A_i-\tau)/s}} \qquad \text{(Equation 2)}$$

In which A_i is the activation value of chunk *i* (calculated using Equation 1), τ is the threshold value after which a chunk is selected and *s* controls the noise (set to about 0.4 by default, or set to 1 if no noise should be included). The function transforms the activation value in a sigmoidal manner [Anderson and Lebiere, 1998]. Therefore, a small increase or decrease in activation value for a chunk that already has a very high or very low activation value will lead to only a small difference in probability value, while in other cases, it can lead to a relatively big difference.

In the ideal situation, where a student has fully mastered a dimension, the probability of the positive chunk of that dimension will be high, and the probability of the negative chunk will be low (due to decay). Situations that differ from these ideal standards should be trained. However, for an application, a more strict definition for selecting training dimensions is necessary.

In the simulations of the next section, we used the following equation that reflects the certainty of our teacher that a player is good at performing a skill of a dimension:

$$Total_j = P_{pos(j)} + \left(1 - P_{neg(j)}\right) + P_{train(j)} \qquad \text{(Equation 3)}$$

Where $Total_j$ is the total score for dimension j, ranging between 0 and 3, and the probability values are the values of the retrieval probability equation (Equation 2) for the three chunks of dimension *j*: the positive chunk, the negative chunk and the amount of training chunk. The function increases if a student has a high probability value for the positive chunk, and a low value for the negative chunk. This reflects that an observing teacher is more certain that a student is good at the skill associated with that dimension if the student completes exercises correctly and *not* incorrectly. Thus, lower values of the total score indicate

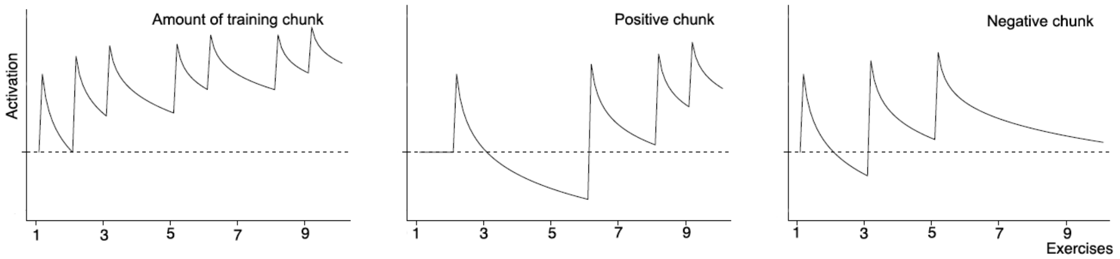


Figure 2: Simulation of the activation values for three chunks of a training dimension. There are positive outcomes for exercises 2, 6, 8 and 9 and negative outcomes for exercises 1, 3 and 5. Notice that the activation of the amount of training chunk increases with both positive and negative outcomes, and that exercises 4 and 7 do not train this dimension.

that the corresponding dimension needs training.

A teacher can only make a good assessment of a student's skills in a certain dimension, when the student has received enough exercises. To stimulate that enough observations of each dimension are made, the probability value of the amount of training chunk is included in Equation 3. With all other values being equal, a dimension with a lower probability of the amount of training chunk will have a lower total score. Thus, in these cases one will first observe a student's behavior in the dimension of which one is least certain about the performance, due to a lack of observations of that dimension. If a student completes the exercise of this dimension, the teacher will become more confident about its observation. If the exercise is completed in a positive way, the probability of the positive chunk of that dimension will increase and therefore, exercises of this dimension will become *less* likely to be presented to the student. However, if the exercise is completed in a negative way, the probability of the negative chunk will increase, and consequently the total score according to Equation 3 will decrease, compared to positive completed situations. Therefore, exercises of this dimension will now become *more* likely to be trained.

4.3 Example simulations

To test if the tutoring mechanism selected the appropriate type of training (i.e., those that a student has most difficulties with), we ran a set of simulations. In these simulations four dimensions were trained in two hundred exercises. Each exercise reflected an interaction that dealt with one dimension, and had two possible outcomes, either positive or negative for that dimension. The exercises were abstract representations for the different ways one can complete an exercise of a dimension. The time between exercises was set to a constant value. Several runs, not discussed here, have shown that results stay fairly similar for different values for this time interval.

The dimension of which an exercise should be presented was dynamically selected using the scores of the dimension according to Equation 3. However, each time we excluded the dimension that had the highest value on the amount of training chunk. Most of the time this was the dimension of which a student had made an exercise last. This guaranteed that the program would not present exercises of just one dimension constantly. For the parameters of the equations we used the following values: d = 0.5, $\tau = 0$, and s = 1 (no noise).

We tested the mechanism for four different types of students. Each character had a different set of likelihoods for completing an exercise of a dimension in a positive way. The *novice* never completed exercises positive, reflecting inexperience with the skill. The *intermediate character* completed 50% of the exercises of all dimensions positive, reflecting that part of the skills is known, but none has been mastered fully. The *partial expert* answered 25% of the exercises on two dimensions correct (A and B in Figure 3), 75% on another dimension (C) and 100% on the fourth dimension (D). This reflects that the partial expert already performs (almost) perfect on some of the skills, being an expert on these sub domains, but still has difficulties with other skills. The fourth character is the *expert*, who completed 100% of the exercises correctly for three dimensions (B, C and D), while answering 75% correct on another dimension (A). Thus the expert has almost completely mastered all skills.

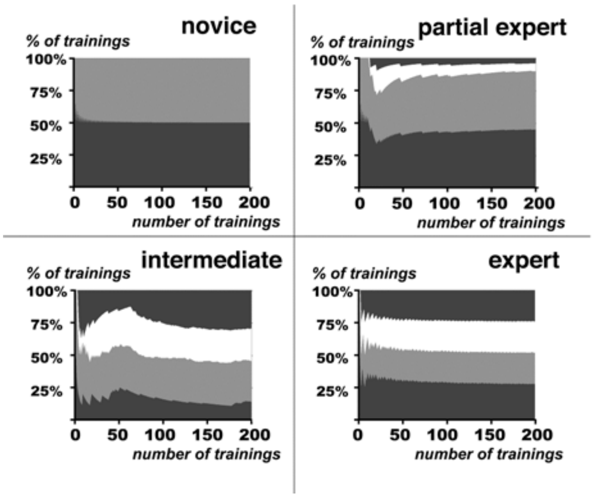


Figure 3: Division of the presented exercises of four dimensions as a function of the total number of presented trainings. The dimensions are: A (lower black area), B (grey area), C (white area) and D (upper black area). Each character (novice, intermediate, partial expert and expert) has different levels of skill for the different dimensions, and therefore a different training. Exercises of dimensions in which they perform worst are presented most.

In Figure 3 the percentage of the exercises that is devoted to each dimension is plotted for the four characters as a function of the amount of presented exercises. At the beginning of the training this pattern fluctuates a lot, as the characters are presented with exercises from different dimensions to assess what dimensions they are not good at. A more stable pattern develops as more exercises are presented. Each of the characters has a different distribution of the dimensions that are trained, which reflects that the user model adapts to individual differences. We will now discuss how the individual patterns can be related to each of the characters of our simulations.

The novice character is only presented with exercises of two dimensions, as the novice fails to complete any of them successfully. This way, the negative chunks of dimensions of which exercises are presented at least once get a high probability value, making the overall performance score for these dimensions low. This way, exercises of these dimensions will be more likely to be picked next time. As the mechanism places a penalty on the dimension with the highest activation value for the amount of training chunk, no two exercises of the same dimension can directly succeed each other. Therefore, the mechanism alternates between exercises of different, but of only *two* dimensions, as the novice already performs very badly on exercises of these dimensions.

The intermediate character has equal skills for each dimension, completing half of the exercises of each dimension correct. The positive reactions temporarily give the teacher an impression that the student is good at the skill of the corresponding dimension. Thus, the mechanism will then select an exercise of another dimension, which makes sure that the student is presented with exercises of *all* dimensions at least once. Over time, the system will become less certain that a student is good at a skill of a previously positively completed dimension, due to decay, and it will once more present an exercise of that dimension. The combination of different responses (both positive and negative) to exercises and decay of activation values lets the mechanism continuously alternate between exercises of different dimensions. As the intermediate student has the same level of skills for each dimension, all dimensions get about the same amount of training. However, as this character answers random (answering half of the exercises correct and the other half incorrect), some dimensions are slightly more trained than others.

The partial expert and the expert are also presented with exercises of all dimensions, as there is not one dimension where they continuously fail. In contrast to the other characters, the partial expert and expert have different levels of skill for different dimensions: exercises of some dimensions are always completed positive, in others they make some mistakes. The partial expert almost always positively completes exercises of dimension C (75% of the exercises) and D (100% of the exercises), but almost never (in 25% of the cases) completes exercises of dimensions A and B correct. Therefore, the system most of the time presents exercises of dimensions A and B.

As with the other characters, the expert is presented with most training on the dimension that it has least developed skills in: dimension A. However, the likelihood of completing dimension A positive (75% of the exercises) does not differ that much from that of the other dimensions (100% correct). Therefore, the amount of presented exercises of dimension A is only slightly higher than that of the other dimensions.

Figure 3 illustrates that the mechanism adapts to the skills of different types of students. However, if the training exercises are good, students will improve their skills while playing. Due to this improvement, they will answer exercises more often correctly. The training selection mechanism has to react to these changes, by offering exercises of other dimensions. To test this situation, a new set of simulations was run using the same initial characters as used for Figure 3. However, this time they learned during training. Learning was simulated by increasing the likelihood of completing an exercise positive with one percent each time that the characters were presented with an exercise of that dimension. For example, if the expert has been presented with ten exercises of dimension A, the likelihood of answering an exercise of that dimension positive has increased with ten percent from 75% to 85%.

The distribution of trainings is illustrated in Figure 4, and has a different shape than in Figure 3. During training, performance on exercises in each of the dimensions increases. The system still mostly selects exercises of dimensions in which a student performs worst, and in time students will become equally good (that is: perfect) in performance on all dimensions. Therefore the shape of the distribution goes towards that of the expert in Figure 4: all dimensions get an equal amount of training. As the novice gets better at some dimensions (and sometimes answers exercises positive) already in the beginning, but is at that moment worse at other dimensions, training is now focused on all four dimensions instead of on only two.

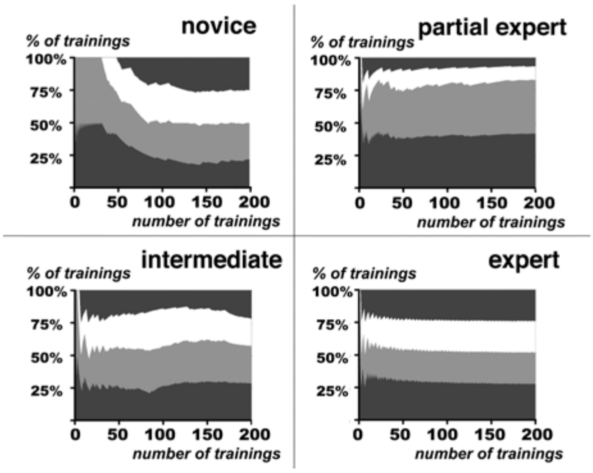


Figure 4: Division of the presented exercises of four dimensions when the characters learn during the training. The better the characters start performing on all dimensions, the more evenly spread the distribution of trainings becomes.

The higher the initial difference is between performances on the different dimensions for a character, the longer it will take before this scenario of continuous equal distributed training on all dimensions will take. Therefore, the partial expert mostly trains two dimensions, while the intermediate and novice are presented with an almost equal amount of exercises of all dimensions.

5 Discussion

In this paper a method for intelligent tutoring in complex contexts such as serious games was introduced as an alternative for the method of model tracing. An alternative was necessary, because model tracing requires a strict decomposable domain [Anderson and Gluck, 2001], which might be unavailable for some training domains. Moreover, it might be difficult to trace the mental states for all of the actions that are possible in games.

The model we introduced requires neither a strict hierarchical domain, nor a detailed understanding of the mental processes that lead a user to his or her actions. Rather, it mimics aspects of an observer, or teacher, who can assess when a student is doing something correct or incorrect, and over time can develop an impression about the student's strengths and challenges. This is not to say that the mechanism acts *exactly* like a teacher. This would be a difficult objective, as games are different settings than classrooms and therefore require at least slightly different approaches. However, the introduced method does include a continuous skill assessment that is used for training selection. This is an important aspect for good teaching [Shepard, 2000], hence the name teacher modeling.

Due to space constraints, we compared teacher modeling only with model tracing, but of course there are other methods for intelligent tutoring. Constraint-based modeling [Mitrovic and Martin, 2007], dynamic bayesian networks [see Manske and Conati, 2005, for an implementation in a game], and predictive learning curves [Baker *et al.*, 2007] are alternatives. However, these methods can only be applied for hierarchical domains, while teacher modeling can be applied in non-hierarchical domains.

Teacher modeling requires a set of broad training dimensions, and a set of exercises with multiple outcomes, each of which has been labeled by experts as being completed good or bad for a specific dimension, or as not making use of skills from that dimension. In the approach we used principles from a cognitive architecture to keep close to aspects of human cognition and learning. In our case, the mechanisms that we used are claimed to be good at modeling effects of recency and frequency of occurrence [Anderson and Schooler, 1991]. Therefore, the type of exercises that the system will select depends not only on the success of completion, but also on the frequency and recency at which the exercises have been presented. As the system can adapt training to the individual, it is preferred over simpler strategies such as level based games, in which students possibly also have to complete exercises that they are already good at and that therefore might be boring.

However, adaptive training is not a guarantee for a motivating training. As can be seen with the novice in Figure 3, the training mechanism successfully offers training on only two dimensions if a user shows repeated failure on corresponding exercises. However, training only aspects in which you fail to improve might be frustrating, and can indicate that the training is not appropriate for the player. These cases can be detected by looking at the dimension scores: these will be low and have a small variance. If detected, the system should respond to them, for example by calling in an expert.

These advanced features were not included in the simulations that we discussed, which have shown how the introduced mechanism adaptively selects training exercises. Mostly the dimensions in which students have difficulties were selected for training, also in cases where students improve their performance during game play. The effect of the training mechanism still has to be tested in real life, and each specific implementation will require a user study for its effect. The success of the test will crucially depend on the quality of the exercises that are developed, and on the interrater-reliability for the labels that experts assigned to the outcomes of the exercises: the higher the reliability, the more certain one can be that this label is correct for the behavior of students who chose this outcome, thereby decreasing the risk of an incorrect assessment.

If more exercises need to be added to the game, this is relatively easy with the teacher modeling approach. This only requires a set of exercises of which the outcomes have been labeled in broad terms; it does not require a strict decomposition of all interactions and mental states as is required in model tracing. Still it can be challenging to develop exercises, as they also have to fit game constraints. The number of game constraints and the amount of impact they might have on the difficulty for developing exercises will depend on the type of game at hand. For example, in role-playing games where a student is free to interact with virtual characters, every exercise has to be developed for a virtual character, and every virtual character must have enough exercises available for the intelligent tutoring system to make a good selection.

Although we introduced our method in the context of intelligent tutoring systems, it might also be considered as a more general method for recommender systems, because it can be used to produce individual recommendations as output [Burke, 2002]. In the context of serious games these individual recommendations can be exercises.

To further test the mechanism of teacher modeling, it is currently being implemented in a serious game. The goal of this game is to train a player's assertive skills. More on this specific project can be read in [Janssen *et al.*, 2007].

Acknowledgments

This work is funded by the Regional programme of innovative actions in Groningen (project UBAssertiv).

References

[Aldrich, 2005] Aldrich, C. *Learning by doing: A comprehensive guide to simulations, computer games, and pedagogy in e-learning and other educational experiences*. Pfeiffer, San Francisco, 2005.

[Anderson *et al.*, 2004] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. An integrated theory of the mind. *Psychological Review*, 111(4): 1036-1060, 2004.

[Anderson and Gluck, 2001] Anderson, J. R., and Gluck, K. What role do cognitive architectures play in intelligent tutoring systems? In S. M. Carver and D. Klahr (Eds.), *Cognition & instruction: Twenty-five years of progress*, pages 227-262, Erlbaum, Mahwah, NJ, 2001.

[Anderson and Lebiere, 1998] Anderson, J. R., and Lebiere, C. J. *The atomic components of thought*. Erlbaum, Mahwah, NJ, 1998.

[Anderson and Schooler, 1991] Anderson, J. R., and Schooler, L. J. Reflections of the environment in memory. *Psychological Science*, 2(6): 396-408, 1991.

[Baker *et al.*, 2007] Baker, R. S. J. de, Habgood, M. P. J., Ainsworth, S. E., and Corbett, A. T. Modeling the acquisition of fluent skill in educational action games. In *Proceedings of the 11th International Conference on User Modeling*, pages 17-26, Corfu, Greece.

[Burke, 2002] Burke, R. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4): 331-370, 2002.

[Gee, 2003] Gee, J. P. *What video games have to teach us about learning and literacy*. Palgrave Macmillan, New York, 2003.

[Janssen *et al.*, 2007] Janssen, C. P., Van Rijn, H., Van Liempd, G., and Van der Pompe, G. User modeling for training recommendation in a depression prevention game. In *Proceedings of the first NSVKI student conference*, pages 29-35, Nijmegen, The Netherlands.

[Johnson and Taatgen, 2005] Johnson, A., and Taatgen, N. A. User modeling. In R. W. Proctor and K. L. Vu (Eds.), *The handbook of human factors in web design*, pages 424-438, Erlbaum, Mahwah, NJ, 2005.

[Manske and Conati, 2005] Manske, M., and Conati, C. Modelling learning in an educational game. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2005

[Michael and Chen, 2005] Michael, D. R., and Chen, S. L. *Serious games: Games that educate, train, and inform*. Thomson course technology, Boston, 2005.

[Mitrovic and Martin, 2007] Mitrovic, A., and Martin, B. Evaluating the effect of open student models on self-assessment. *International Journal of Artificial Intelligence in Education*, 17(2): 121-144, 2007.

[Ohlsson, 1986] Ohlsson, S. Some principles of intelligent tutoring. *Instructional Science*, 14(3): 293-326, 1986.

[Pavlik and Anderson, 2005] Pavlik, P. I., and Anderson, J. R. Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, 29: 559-586, 2005.

[Prensky, 2007] Prensky, M. Social impact games website (www.socialimpactgames.com). Retrieved 19-01-2007

[Shepard, 2000] Shepard, L. A. The role of assessment in a learning culture. *Educational Researcher*, 29(7): 4-14, 2000.

Prediction Algorithms for User Actions

Melanie Hartmann and Daniel Schreiber
Telecooperation Group, Darmstadt University of Technology
D-64289 Darmstadt, Germany
{melanie,schreiber}@tk.informatik.tu-darmstadt.de

Abstract

Proactive User Interfaces (PUIs) aim at facilitating the interaction with a user interface, e.g., by highlighting fields or adapting the interface. For that purpose, they need to be able to predict the next user action from the interaction history. In this paper, we give an overview of sequence prediction algorithms (SPAs) that are applied in this domain, and build upon them to develop two new algorithms that base on combining different order Markov models. We identify the special requirements that PUIs pose on these algorithms, and evaluate the performance of the SPAs in this regard. For that purpose, we use three datasets with real usage-data and synthesize further data with specific characteristics. Our relatively simple yet efficient algorithm FxL performs extremely well in the domain of SPAs which make it a prime candidate for integration in a PUI. To facilitate further research in this field, we provide a Perl library that contains all presented algorithms and tools for the evaluation.

1 Introduction

Nowadays applications gain more and more functionality, mostly leading to a decreased usability. An intelligent interface can counter this effect by explaining the user how to work with an application, by performing repetitive actions on his behalf, by adapting the interface to his needs, and by suggesting content for input fields. Proactive user interfaces (PUI) aim at combining all these features in an augmentation of traditional user interfaces. We state that the main features of a PUI are:

- **Support Mechanisms**: provide online help that adapts to the user and his current working context
- **Interface Adaptation**: adapt the provided options and content to the user's needs and preferences
- **Content Prediction**: suggest data to be entered that is inferred from previous interactions and context information
- **Task Automation**: recognize usage patterns to allow automation of repetitive tasks.

Thereby, the interface adaptation can range from highlighting the functions the user really needs in his current context to minimizing the interface to these functions. There has been a debate for years whether automatic adaptation of the user interface helps or confuses the user [Findlater and McGrenere, 2004; Sears and Shneiderman, 1994; Mitchell and Shneiderman, 1989]. Adaptation that is not explicitly asked for by the user is often seen as a flaw that introduces unpredictable behavior and that therefore should be avoided. However, [Gajos *et al.*, 2006] shows that carefully planned adaptation can indeed improve usability and avoid confusion. We assume that interface adaptation in form of highlighting interface elements does not confuse the user and can reduce his cognitive load by helping him to focus on the important parts of the interface. Furthermore, there are also scenarios, in which a complete reordering of the interface is needed. For example, mobile users often have only a small display to their disposal, which makes it essential that the shown functions are limited to those the user needs most. For these tasks, we need an algorithm that can predict the next user actions, a so called-sequence prediction algorithm (SPA). However, knowing the next user action is not only of importance for the interface adaptation, but also for supporting the user by telling him what to do next, for suggesting content for the next input field and for automating tasks.

In this paper, we examine statistical methods for sequence prediction that can be applied without any prior task knowledge. We explore the limits of such an approach to define the baseline for further enhancements. Moreover, we compare the best performing algorithms with respect to the required memory and processing time, as those factors are of great importance when used on small mobile devices. In future research, we aim at integrating knowledge about the task and the context in the SPAs. This knowledge can be provided by the application developer, learned by the PUI, and/or modified by the user. This should then also enable us to predict not only the next action, but also content for input fields what can hardly be achieved by pure statistical approaches.

There already are some papers on SPAs, but they mostly compare just one algorithm to a baseline and moreover lack a formal description of the presented algorithms. Further, they do not describe a general method for the evaluation that is applicable to different application domains with varying requirements on the SPA. In this paper, we try to overcome these limitations first by giving a broad overview of existing algorithms with their formal descriptions and second by showing how SPAs can be systematically evaluated. To facilitate further research in this field, we provide a Perl library [Hartmann and Schreiber, 2007] that contains all presented algorithms and some tools for the evaluation.

The structure of this paper is as follows. In Section 2, we provide a short definition of sequence prediction along with some specific requirements for the problem of user action prediction in PUIs. We give an overview of existing algorithms in Section 3 and present two enhanced algorithms

FxL and Adaptive FxL in Section 4. Section 5 describes a systematic approach to compare the performance of these algorithms. The findings of our experiments are reported in Section 6. The paper concludes with an outline of some issues for further research.

2 Sequence Prediction

Algorithms for sequence prediction were mainly developed in the area of data compression. Intelligent user interfaces also make use of such predictions for facilitating interaction (e.g., in an intelligent home environment [Gopalratnam and Cook, 2007]) or for assisting the user by giving explanations (e.g., adaptive tutoring system [Künzer *et al.*, 2004]).

The problem of predicting the next symbol (representing a user action) in an input sequence can be formally defined as follows: Let Σ be the set of possible input symbols and let $A = a_1...a_n$ with $a_j \in \Sigma$ be a sequence of input symbols of which the first i symbols, that is $a_1 \dots a_i$, have already been observed. A SPA decides at first whether it is able to make a prediction and if so returns the probability for each symbol $x \in \Sigma$ that x is the next element in the input sequence. These values define a conditional probability distribution P over Σ, where $P(x|a_1 \dots a_i)$ is the probability for the singleton subset of Σ containing x. Most algorithms consider only the last k elements of the input sequence for the computation (i.e., $P(x|a_1 \dots a_i) = P(x|a_{i+1-k} \dots a_i)$).

In principle, there are two ways of calculating the probabilities: on-demand or live. Algorithms using the former method maintain a data structure to compute the probabilities. They update the data structure after each symbol in the input sequence, whereas the live algorithms update the probability distributions itself. We do not explicitly address the fact that Σ may be unknown to the algorithm and assume that the probability of an unseen element is 0.

For applying a SPA in a PUI, it has to satisfy certain requirements. At first it has to meet the requirements defined for an Ideal Online Learning Algorithm in [Davison and Hirsh, 1998]. Further, the algorithm needs a high applicability. For example, if we want to reduce the user interface to the most relevant functions for mobile usage, we always need a prediction what might be used next. Due to the small screens of mobile devices, it is simply not possible to display all available functions. Moreover, if we look at adapting the interface by highlighting operations, the costs that arise from false highlighting are minimal as the user will be at most distracted for a short moment. Another requirement that PUIs pose is that the algorithm should be able to return good predictions even with a small amount of training data, as the PUI should already be able to assist the user after few usages. Further, we assume that we have to deal with a high amount of long repetitive sequences. This is motivated by the fact that we aim to equip mostly form based applications that are likely to be processed in a specific manner.

3 Existing Algorithms

In this section, we describe the main ideas of the four most prominent existing SPAs. Their implementations are included in the Perl library [Hartmann and Schreiber, 2007]. Most existing algorithms use Markov models to predict the next action. They often combine the results of several different order Markov models to obtain optimal performance.

IPAM

One of the first algorithms for predicting the next user action is IPAM [Davison and Hirsh, 1998]. It employs a first order Markov model, i.e., it bases its predictions only on the last seen symbol of the input sequence. IPAM maintains a list of unconditional probabilities $P_{ipam}(x),\ x \in \Sigma$ and a table of conditional probabilities $P_{ipam}(x|y),\ x, y \in \Sigma$ with the latter being directly used as probability distribution for predicting the next symbol, i.e., $P(x|a_1 \dots a_i) = P_{ipam}(x|a_i)$. This live algorithm can be described in a recursive formula, where the new probabilities $P'_{ipam}(x|y)$ are computed for all $x \in \Sigma$ from $P_{ipam}(x|y)$ using a weight α after observing symbol a_{i+1} in the sequence A according to following equation:

$$P'_{ipam}(x|a_i) = \begin{cases} \alpha\ P_{ipam}(x|a_i) + (1-\alpha) & \text{if } x = a_{i+1} \\ \alpha\ P_{ipam}(x|a_i) & \text{otherwise} \end{cases}$$

The probabilities $P_{ipam}(x|y)$ for $y \neq a_i$ are not updated and the same equation holds for the unconditional probabilities. If a symbol y was observed for the first time, its conditional probability distribution is initialized with $P_{ipam}(x|y) = P_{ipam}(x)\ \forall x \in \Sigma$ and its unconditional probability is initialized with 0 before updating the probabilities as above.

According to some empirical results, Davison and Hirsh recommend a value of 0.8 for α. Thus, the more recent actions have a greater impact on the predictions than older actions. The basic idea of IPAM is also used in other prediction algorithms. Thereby, the table of conditional probabilities is often extended so that the conditional probabilities depend not only on the last seen item, but also on the last two items etc.

ONISI

Gorniak and Poole argue that the last action does not provide enough information to predict the next action [Gorniak and Poole, 2000]. Hence they build an on-demand prediction model called ONISI that employs a k-nearest neighbors scheme. Thereby, they consider not only the actions performed by the user, but also the corresponding user interface states. They compute a probability distribution according to the k longest sequences in the interaction history that match the immediate history. Gorniak and Poole found that $k = 5$ was sufficient to gain optimal results in their sample application (a web application for learning AI concepts). However, some actions are strongly correlated to the state in which they are performed but do not belong to long sequences. To account for this fact, Gorniak and Poole weigh off the probability distributions determined by the current state and by the longest sequences. For that purpose they use a weighting factor α that they found to perform best with a value of 0.9.

$$P_{onisi}(x|(s_1, a_1) \dots (s_i, a_i)s_{i+1}) =$$
$$\alpha \frac{l(s_{i+1}, x)}{\sum_y l(s_{i+1}, y)} + (1-\alpha)\frac{fr(s_{i+1}, x)}{\sum_y fr(s_{i+1}, y)}$$

Thereby, $l(s, a)$ returns how often the state action pair (s, a) occurred after the longest k sequences that match the recent interaction history. In contrast, $fr(s, a)$ reflects how often action a occurred in the interface state s.

For our experiments, we have to limit the algorithm to a single interface state, as the real datasets used in the experiments have no interface states and no restriction for the order of actions.

Jacobs Blockeel

Jacobs and Blockeel [Jacobs and Blockeel, 2002] claim that the longest match in the history for the immediate history is not always the best choice for determining the probabilities of the next symbol and that the ideal length can not be known in advance. Their live algorithm builds upon IPAM but allows longer premises in the table of conditional probabilities. For that purpose, they add a step that is performed only after a correct prediction was made. If the algorithm made a correct prediction for a_{i+1} after observing the sequence $A = a_1 \dots a_i$, new entries for every C that is suffix of A and where $P_{jb}(x|C) > 0$ are added to the probability table. Let L be the longest suffix of the concatenation $C \circ a_{i+1}$ for which already an entry $P_{jb}(x|L) > 0$ exists. This probability distribution is taken as the best estimation of the new probabilities. Let P_{jb} be the probabilities that result after the IPAM update step. Next the probabilities for all new premises $C \circ a_{i+1}$ are computed as

$$P'_{jb}(x|C \circ a_{i+1}) = P_{jb}(x|L).$$

So the algorithm does not rely on a fixed order Markov model, but uses a mixed order approach to compute the probability distribution of the next element. Using the above equations, the sum of the probabilities over Σ is not always 1 and a normalization has to be performed.

ActiveLeZi

Another on-demand algorithm that considers several Markov models is ActiveLeZi [Gopalratnam and Cook, 2007]. It stores the frequency of input patterns in a trie according to the compression algorithm LZ78. To overcome some of the problems arising with LZ78, a variable length window of previously-seen symbols is used. The size of the window grows with the number of different subsequences seen in the input sequence. Let suf_l be the suffix of length $l + 1$ of the immediate interaction history A, that is $a_{i-l} \dots a_i$. The probabilities are recursively defined as follows:

$$P^0_{alz}(x|A) = \frac{fr(x)}{\sum_{y \in \Sigma} fr(x \circ y)}$$

$$P^l_{alz}(x|A) =$$

$$\frac{fr(suf_l \circ x)}{fr(suf_l)} + \frac{fr(suf_l) - \sum_{y \in \Sigma} fr(suf_l \circ y)}{fr(suf_l)} P^{l-1}_{alz}(x|A)$$

Thereby, $fr(x)$ returns the frequency of the input pattern x as stored in the trie. The probability distribution that is finally returned by ActiveLeZi is P^k_{alz} where k is the current size of the window.

4 FxL and Adaptive FxL

We agree with most papers that knowing only the last action is not sufficient for making good predictions. Therefore, we tested two further on-demand approaches for combining the results from different order Markov models. The presented algorithms build on the KO algorithm [Künzer *et al.*, 2004] but vary in the weighting functions used. Further, the KO algorithm does not achieve high applicability values with the proposed parameters as it takes only frequencies of subsequences into account that have a minimal support in the input sequence.

The algorithm builds upon an n-gram trie containing the frequencies of different input subsequences. The function $fr(a_1 ... a_i)$ returns how often the sequence $a_1 ... a_i$ has already been seen. To reduce the amount of data that needs to be stored, only n-grams of a length up to a specified value k are taken into account. The n-gram models are then used to assign a score to every symbol denoting the probability of a symbol to occur next in the input sequence. As the scores for the symbols can sum up to a value greater than 1, they have to be normalized. Thus

$$P(x|a_1 ... a_i) = \frac{score(x)}{\sum_{y \in \Sigma} score(y)}$$

where $score(x)$ is calculated by adding the absolute frequencies of x succeeding any suffix (up to length $k - 1$) of $a_1 ... a_i$. As the longer suffixes yield more reliable results than the shorter ones, the frequencies are assigned a weight $w(j)$ depending on the length j of the suffix that is considered. Thus, the score is computed as follows:

$$score(x) = \sum_{j=1}^{k-1} w(j) fr(a_{i+1-j} ... a_i \circ x)$$

For $w(j)$ we tested two different weight measures. At first we tried a very simple approach that uses the suffix-length as weight, $w(j) = j$. We call this approach **FxL** as the score for a symbol is calculated by simply multiplying the frequency (F) of the symbols with the length (L) of the suffix they succeed.

However, such an approach does not adapt to the specific features of a dataset (e.g., that many shorter sequences occur in the dataset and thus need a greater weight assigned). Therefore, we developed a new algorithm called **Adaptive FxL** that considers the predictive quality of the different order models. We define the predictive quality q_i of the i-th order model as the percentage of how often it was able to make a correct prediction. For that purpose, we store for each model how often it was able to make a prediction at all (applicability ap) and how often this prediction was correct (c). The predictive quality of a model then results in $q = c/ap$. As we found that assigning greater weights to longer suffixes leads to better results, we did not use the predictive quality alone as weight, but also considered the suffix-length and an additional factor f_j: $w(j) = j\ q_j f_j$. Thereby, the factor f_j reflects the probabilities that all higher order models make a wrong prediction. Thus, the factor f_j is 1 for the highest order model that is able to make a prediction for the current sequence, and is reduced for the lower models with $f_{j-1} = (1 - q_j) f_j$.

5 Evaluating SPAs

For evaluating the performance of SPAs the following metrics are used in the literature: prediction accuracy pr_{ac}, prediction probability pr_p, and applicability ap. The prediction accuracy and probability are computed by assigning a score to every prediction made by the algorithm and averaging over the number of predictions made by the algorithm. Thus we can compute the prediction accuracy pr_{ac} and probability pr_p over the whole input sequence $a_1 ... a_n$ using following equation:

$$pr_x(a_1 ... a_n) = \frac{1}{m} \sum_{i=0}^{n-1} eval_x(a_1 ... a_i, a_{i+1})$$

where m ($\leq n$) is the number of predictions made and $eval_x$ the evaluation function that returns the value for a single prediction (it returns 0 if no prediction was made by the algorithm). The $eval_x$ function thereby differs for the two metrics:

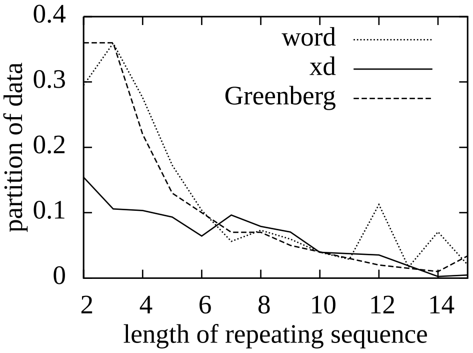


Figure 1: Distribution of repetitive sequences for three real datasets used for the evaluation

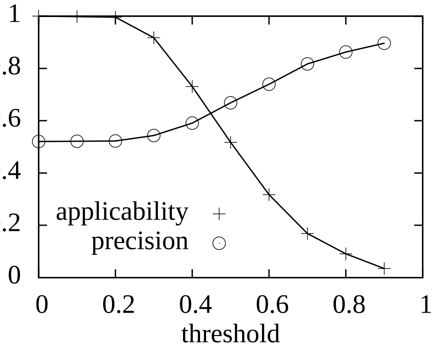


Figure 2: Applicability and precision for different confidence thresholds

The prediction accuracy $eval_{ac}$ considers only the symbols which are predicted with maximal probability. We define the set $\hat{A}$ as the set of all these symbols: $\hat{A}_{i+1} = \{x \in \Sigma | \forall y \in \Sigma . P(x|a_1 \dots a_i) \geq P(y|a_1 \dots a_i)\}$. The $eval_{ac}$ function for the prediction accuracy then returns a value reflecting whether the actual next symbol a_{i+1} is among these values $\hat{A}_{i+1}$. Thus, $eval_{ac}$ is 0 if $a_{i+1} \notin \hat{A_{i+1}}$ otherwise it is computed as $eval_{ac}(a_1...a_i, a_{i+1}) = \frac{1}{|\hat{A}_{i+1}|}$.

In contrast, the $eval_p$ function for the prediction probability rates with which probability the correct symbol was suggested, no matter if it has been assigned a maximal probability or not: $eval_p(a_1...a_i, a_{i+1}) = P(a_{i+1}|a_1...a_i)$.

Albrecht [Albrecht *et al.*, 1998] states that the prediction probability provides finer-grained information about the performance of an algorithm than the prediction accuracy, but that both measures produce generally consistent assessments.

Finally, the third metric applicability ap is defined as the ratio of input symbols the algorithm was able to make a prediction for: $ap = m/n$.

As the algorithms are often evaluated on datasets containing the input sequences of several users, the results have to be averaged over all users. This can be done by computing the average over the results of all users weighing every user equally, independent of the length of the corresponding sequence (macroaverage), or by averaging over all data (microaverage), emphasizing frequent users.

In order to judge which algorithm best suits the requirements of an application, we need to evaluate the algorithms depending on several parameters influencing their performance, mostly characteristics of the input sequences. The most important parameters are: (1) dataset size available for training, (2) distribution of repetitive sequences and (3) noise in the repetitive sequences.

We call a sequence repetitive if it is not part of a longer repetitive sequence and occurs at least k times in the corresponding dataset. The parameter k can be a constant or defined relatively to the dataset size. For our experiments we set $k = 5$. Figure 1 shows the sequence distribution for the real datasets Word, XD, and Greenberg used in the experiment. Noise is introduced in a repetitive sequence if the user sometimes alters the sequence of actions. However, to measure noise the repetitive sequences have to be known in advance.

To reduce the target metrics that have to be evaluated for judging the performance of an algorithm, the applicability can be turned into the fourth parameter. This also enables a better comparability of the algorithms, as they can vary heavily in their applicability values. For measuring the prediction accuracy and prediction probability for a given applicability, several (ap, pr) pairs have to be computed to infer the pr value corresponding to the specified ap value from them. The (ap, pr) pairs can be obtained by evaluating the algorithm using several thresholds for the reported predictions. This means that only predictions are taken into account whose probability is over this threshold. We assumed (and also found) that a higher threshold (up to 80 or 90%) leads to a decrease in applicability and to an increase in the prediction accuracy of the algorithm (see Figure 2). If the threshold is relatively high (over 80 or 90%) the algorithm is often only able to make very few predictions for the whole input sequence which makes the prediction value unreliable. However, this does not corrupt the results, as such low applicability values are hardly ever of interest. The required pr value is finally computed by linear interpolation of the (ap, pr) values.

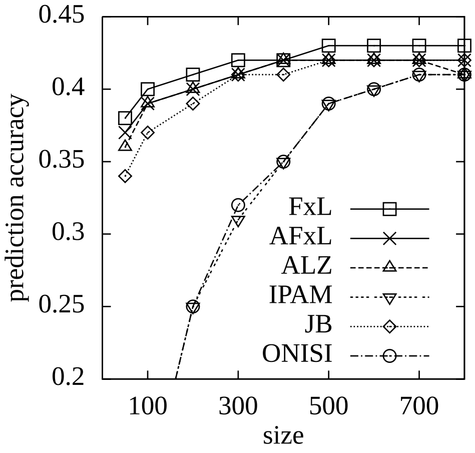


Figure 3: Performance on the Greenberg dataset regarding dataset size

In order to facilitate the evaluation, we provide a Perl library [Hartmann and Schreiber, 2007] containing the implementation of all algorithms presented in this paper. It can easily be extended with new algorithms and contains special support for facilitating the implementation of n-gram based approaches. Further, this library contains methods that compute the prediction probability or accuracy values for these algorithms.

6 Experiment

For the evaluation of the SPAs we used three real usage datasets (Greenberg, XD and Word) as well as synthesized data. The Greenberg dataset [Greenberg, 1988] is widely

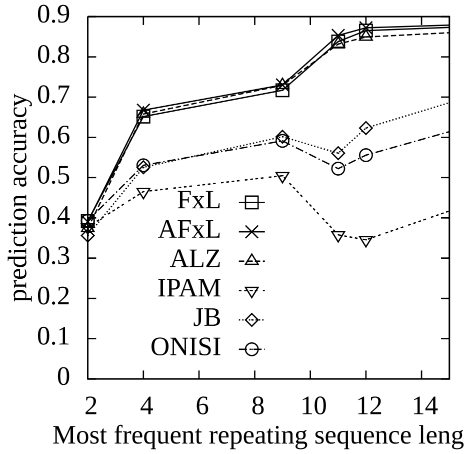


Figure 4: Performance regarding sequence length

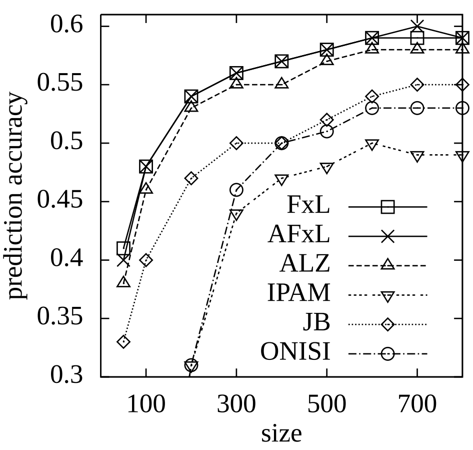


Figure 6: Performance on the XD dataset regarding dataset size

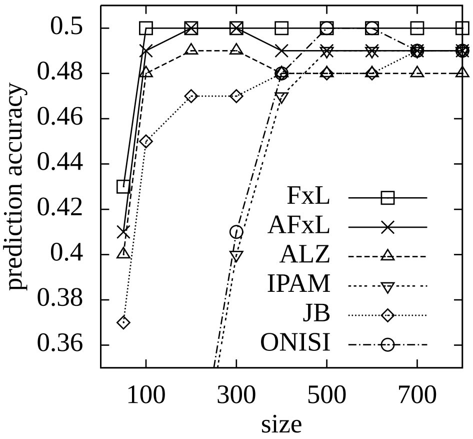


Figure 5: Performance on the Word dataset regarding dataset size

used in the literature and contains over 225 000 UNIX commands from 168 users assigned to four groups depending on their computer experience. The CrossDesktop[1] dataset (abbreviated XD) contains log data from a web application for managing files and emails. It contains about 200 000 requests from 37 users whereat the usage varies heavily between the different users. The third dataset (Word) contains logs of MS Word usage[2] described in [Linton *et al.*, 2000]. The synthesized datasets were generated to evaluate the algorithms with respect to the parameters "distribution of repetitive sequences" and "noise in repetitive sequences" that can not be varied in the real datasets. For all datasets we used the first 20% for training the algorithms, where the predictions were not included in the performance evaluation, if not explicitly stated otherwise. We chose to macroaverage the results for the different users because it is more important to optimize the results for the average than for the frequent user in our application domain . Further, we used only prediction accuracy as evaluation metric for presented results because we are more interested in the prediction with the highest probability than in its actual probability value. However, we yield consistent results if we use prediction probability or if we microaverage the results. As discussed in Section 5, we analyze the performance of the algorithms regarding the four parameters training size, distribution of repetitive sequences, noise, and applicability. Due to space limitations, we do not go into detail regarding the applicability of the algorithms, we just normalize the applicabilities as described in Section 5 to ensure the comparability of the algorithms. In the presented results, we used applicability levels of 90%.

At first we calculated the prediction accuracies for the different algorithms depending on the sequence length. Thereby we included all predictions in the performance evaluation (no training data). As the results in Figure 5-3 show, FxL and Adaptive FxL (AFxL) perform best on all datasets followed by ActiveLeZi (ALZ).

In Section 5, we already pointed out that the three datasets differ in the underlying sequence distributions. To analyze the algorithms regarding this parameter, we created random datasets that varied in this parameter (using sequence lengths of 500 symbols). Figure 4 shows the average prediction accuracies over 80 randomly generated datasets. Thereby the x-coordinate represent the sequence lengths that were repeated most often in the dataset. The result is similar to the previous ones: The algorithm FxL, Adaptive FxL and ActiveLeZi perform better than the other ones.

At last, we have to consider the parameter noise for the evaluation. Thus, we also synthesized datasets that varied in their noise level. We used three operators to insert noise: An element from the original sequence could be left out, repeated up to 5 times or swapped with the following element. The versions of the dataset differed with which a noise operator was applied. The results in Figure 7 show the decay in the algorithms' performances with rising noise levels and confirm our former results.

To sum it up, we evaluated all algorithms varying the four parameters applicability, dataset size, sequence distribution and noise level using real and synthetic datasets. The overall result shows that the algorithms FxL, Adaptive FxL and ActiveLeZi perform best regarding all tested parameters.

For choosing the optimal candidate for the integration in a PUI, we also have to consider the storage and computational costs. These factors play a significant role in PUIs, as they are also used on mobile devices with memory and power restrictions. The storage costs are limited for the FxL and AFxL algorithms by the specified k and the amount of possible user actions ($|\Sigma|$), whereas the storage costs of ActiveLeZi grow with the dataset size. For our

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

[2] http://www.cs.rutgers.edu/ml4um/datasets/

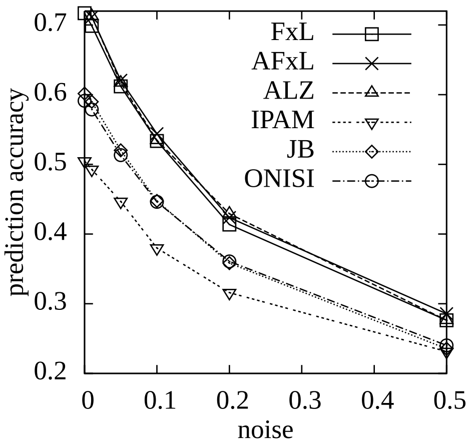


Figure 7: Performance of the algorithms regarding different noise levels

		unix	Word	XD
prediction accuracy	FxL	44.1%	57.8%	50.4%
	AFxL	43.9%	58.2%	50.1%
	ALZ	42.7%	56.0%	48.5%
stored keys	FxL:3	258	227	111
	FxL:6	2061	1599	949
	ALZ	2706	2743	1955
comput. time [s]	FxL	0.57	0.54	0.32
	AFxL	1.08	1.08	0.68
	ALZ	1.47	1.46	0.89

Figure 8: Average performance values for the three real datasets

tests we used AFxl and FxL with $k = 5$, but we found that the algorithms already reach their optimum with $k = 3$ or $k = 4$ and stay at that level for larger k. Figure 8 lists the average costs and prediction accuracies of the three prime candidates for the three real datasets. Thereby, “stored keys” reflects the keys contained in the main data structure used by the algorithm. As FxL and AFxL rely on the same data structure and vary with the specified k, we list values for the FxL algorithms with $k = 3$ and $k = 6$. The results show that the FxL algorithm clearly outperforms the other two, and is thus the best candidate for applying it in PUIs.

7 Conclusion and Future Work

In this paper, we presented an overview of existing SPAs which can be used for predicting the next user action. We analyzed the parameters that have to be considered for the evaluation, especially focusing on the demands of a PUI. We found that the simple FxL algorithm performs best and is thus the prime candidate for the incorporation in PUIs. However, we saw that the prediction accuracy achievable with pure statistical approaches is limited to 40-60%. We assume that integrating task and context knowledge can improve these results. This leads to the problem how this task knowledge should be acquired. To solve this problem, we aim at combining modeled and learned knowledge. The PUI learns the taskmodel from observation, e.g., which actions are available and which context information corresponds to them. As the knowledge learnable by observation is limited, we enable the user to modify and extend the taskmodel, and to enrich it with constraints and interrelations between task model and context information. In the ideal case, such an enhanced taskmodel is provided by the application itself.

Acknowledgments

We would like to thank SAP Research Darmstadt for supporting our research in the AUGUR project.

References

[Albrecht *et al.*, 1998] David W. Albrecht, Ingrid Zukerman, and Anne E. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User Modeling and User-Adapted Interaction*, 8(1-2):5–47, 1998.

[Davison and Hirsh, 1998] Brian D. Davison and Haym Hirsh. Predicting Sequences of User Actions. In *Proceedings of AAAI-98/ICML-98 Workshop Predicting the Future*, pages 5–12. AAAI Press, 1998.

[Findlater and McGrenere, 2004] Leah Findlater and Joanna McGrenere. A comparison of static, adaptive, and adaptable menus. In *Proceedings of SIGCHI*, 2004.

[Gajos *et al.*, 2006] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of AVI ’06*. ACM Press, 2006.

[Gopalratnam and Cook, 2007] Karthik Gopalratnam and Diane J. Cook. Online sequential prediction via incremental parsing: The active lezi algorithm. volume 22, pages 52–58, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[Gorniak and Poole, 2000] Peter Gorniak and David Poole. Predicting future user actions by observing unmodified applications. In *AAAI/IAAI*, 2000.

[Greenberg, 1988] Saul Greenberg. Using unix: collected traces of 168 users. Research report 88/333/45, 1988.

[Hartmann and Schreiber, 2007] Melanie Hartmann and Daniel Schreiber. A perl library of sequence prediction algorithms. "http://www.tk.informatik.tu-darmstadt.de/index.php?id=augur", 2007.

[Jacobs and Blockeel, 2002] N. Jacobs and H. Blockeel. Sequence prediction with mixed order Markov chains. In *Proceedings of the Belgian/Dutch Conference on Artificial Intelligence*, 2002.

[Künzer *et al.*, 2004] A. Künzer, F. Ohmann, and L. Schmidt. Antizipative modellierung des benutzerverhaltens mit hilfe von Aktionsvorhersage-Algorithmen. *MMI-Interaktiv*, (7), 2004.

[Linton *et al.*, 2000] Frank Linton, Deborah Joy, Hans-Peter Schaefer, and Andrew Charron. Owl: A recommender system for organization-wide learning. *Educational Technology & Society*, 3(1), 2000.

[Mitchell and Shneiderman, 1989] J. Mitchell and B. Shneiderman. Dynamic versus static menus: an exploratory comparison. *SIGCHI Bull.*, 20(4), 1989.

[Sears and Shneiderman, 1994] Andrew Sears and Ben Shneiderman. Split menus: effectively using selection frequency to organize menus. *ACM Trans. Comput.-Hum. Interact.*, 1(1):27–51, 1994.

State of the Art of Adaptivity in E-Learning Platforms

David Hauger and Mirjam Köck
Institute for Information Processing and Microprocessor Technology
Johannes Kepler University, Linz

Abstract

Adaptivity has been an important research topic during the past two decades, especially in the field of e-learning. This paper deals with the question of whether and to what extent adaptivity is actually being used in e-learning systems. It describes the state of the art of adaptivity features and gives an overview on the most frequently used learning management systems (LMSs) as well as on a number of research projects and systems providing adaptivity.

1 Introduction

Adaptive Hypermedia (AH) has been explored and researched for several years now. In 1996 Brusilovsky claimed that [Brusilovsky, 1996b, p. 1]

> "AH systems can be useful in any application area where the system is expected to be used by people with different goals and knowledge and where the hyperspace is reasonably big. Users with different goals and knowledge may be interested in different pieces of information presented on a hypermedia page and may use different links for navigation."

In the same paper he pointed out that adaptivity can be especially helpful in education and listed some first approaches to educational AH systems. There has been a lot of work and research in adaptive educational hypermedia in the intervening years, evidenced by the number of publications and dedicated events.

This paper examines the extent to which adaptivity is employed today in widely used e-learning systems. The rest of the paper is structured as follows: Section 2 deals with the question whether and how adaptivity can enhance e-learning. Moreover, it introduces several adaptation techniques. Section 3 presents a list of popular LMSs as well as an overview on some systems already providing adaptivity features. The paper is concluded with a summary and discussion of the findings.

2 Adaptivity in E-Learning

This section deals with adaptivity in e-learning systems and explains why and how adaptivity is able to improve the quality of e-learning environments.

2.1 Why Can Adaptivity Enhance E-Learning?

According to [Brusilovsky, 1996a] adaptivity is of particular importance in the field of e-learning for two main reasons. First, a learning system might be used by learners differing in their goals, learning styles, preferences, knowledge and background. Moreover, the profile of a single learner changes (e.g. the knowledge increases as an effect of learning). Second, the system can help the learner to navigate through a course by providing user-specific (not necessarily linear) paths.

Taking care of these differences, the system is able to provide personalized access to the content (fitting the individual user's needs). The fact that the decisions on what is presented are based on the user's profile (e.g. goals, knowledge) allows taking care of a single user. This compensates for one significant problem of common e-learning systems that provide the same view of the information for all learners.

2.2 How Can Adaptivity Enhance E-Learning?

There are several different ways to categorize adaptivity features. [Beaumont and Brusilovsky, 1995] distinguish between adaptation on content level (adaptive presentation support) and on link level (adaptive navigation support).

Adaptive presentation support Adaptive presentation support describes the presented content as an assembly of fragments. Depending on how these fragments are put together [Beaumont and Brusilovsky, 1995] divide adaptive presentation support into "conditional presentation" ([Fischer *et al.*, 1990]), the "stretchtext" technique ([Boyle and Encarnacion, 1994], [Kobsa *et al.*, 1994]) and the "frame-based" technique (used in HYPADAPTER [Böcker *et al.*, 1990] and EPIAIM [De Rosis *et al.*, 1993]).

[Henze, 2000] adds page or page fragment variants as another adaptation technique which is – although similar to the frame-based technique – a more general approach.

[Brusilovsky, 1996b] differentiates between adaptation techniques (implementation level) and adaptation methods (conceptual level). [Brusilovsky, 1996b] lists the following methods for adaptive presentation support:

- Additional explanations: Displays the parts of a document matching the user's knowledge or goal (used in MetaDoc [Boyle and Encarnacion, 1994], KN-AHS [Kobsa *et al.*, 1994], ITEM/IP [Brusilovsky, 1992], EPIAIM and ANATOM-TUTOR [Beaumont, 1994]).
- Prerequisite explanations: If prerequisites for a concept are not sufficiently known, the corresponding information is inserted by the system (used in Lisp-Critic [Fischer *et al.*, 1990] and C-book [Kay and Kummerfeld, 1994]).

- Comparative explanations: Emphasizes similarities between the currently displayed concept and known ones (used in (ITEM/IP, Lisp-Critic and C-book).
- Explanation variants: In some cases displaying or hiding parts of information is not sufficient which leads to creating different variants of a piece of information and presenting the best fitting one (used in ANATOM-TUTOR, Lisp-Critic, HYPADAPTER, ORIMUHS [Encarnação, 1995], SYPROS [Gonschorek and Herzog, 1995] and WING-MIT [Kim, 1995]).
- Sorting: The fragments of information are sorted according to their relevance for the user (used in HYPADAPTER and EPIAIM).

Adaptive navigation support Adaptive navigation support deals with all the possibilities of modifying visual links enabling navigation (e.g. by reordering, hiding or annotation).

As for adaptive presentation support [Henze, 2000] defines various methods for adaptive navigation support (based on [Brusilovsky, 1996b]):

- Direct guidance: The user is provided a sequential path through the system, either using the "next best" strategy (guidance with a "next"-button) or "page sequencing or trails", where reading sequences through (parts of) the system are generated.
- Adaptive sorting: The links of a document are sorted according to their assumed relevance (based on previous knowledge or similarity to the current document).
- Adaptive hiding: Links are hidden or disabled if the system assumes that they are not relevant and/or distracting.
- Link annotation: Links are annotated by text, colouring, an icon, or dimming in order to give some extra information to the learner.
- Map annotation: The discussed annotation methods are used for adapting graphical overviews and/or maps.

Criteria for adaptation An adaptive system may be either concept-based or not bound to a specific concept ([Aroyo *et al.*, 2006]). Concept-based systems use a model of the content (the "domain model" or "content model") to structure the information. If the structure of the content is relatively straightforward or the content is of small size, it may not be necessary to develop a specific model.

Especially in the area of adaptive learning systems concept-based architectures are more commonly used.

According to [Aroyo *et al.*, 2006] the adaptation itself is based on a user's preferences (e.g. learning and cognitive styles, language) as well as on assumptions about the current user's (knowledge) state. [Kareal and Kléma, 2006] state that the presented information should adapt to the learners' prior knowledge and skills, learning capabilities, learning preferences or styles, performance level and knowledge state, interests, personal circumstances (location, tempo, etc.) and motivation.

3 Overview of E-Learning Systems

The first section of this section gives an overview on popular e-learning systems. The second section of this section provides a list of systems using the adaptivity features mentioned in section 2.2.

3.1 Popular E-Learning Systems

The number of e-learning systems has constantly been increasing during the past years as a lot of companies, faculties, universities and other institutions developed systems for common or personal use. Therefore, it is practically impossible to set up a complete list of e-learning systems. The following list includes some of the systems most frequently used in e-learning (mainly Learning Management Systems (LMSs)).

- .LRN [.LRN] is an open source e-learning and community building software originally developed at MIT. Today it is supported by a worldwide consortium of educational institutions, non-profit organisations, some industry partners and open source developers. .LRN is built on the top of OpenACS (Open Architecture Community System) [OpenACS] which is a toolkit for building scalable, community-oriented web applications.
- ATutor [ATutor] is an open source system supporting learning and content management and specifically considering accessibility and adaptability issues. It was first released in 2002 after two studies conducted that evaluated the accessibility of learning platforms to people with disabilities. Several features are planned for the near future, including a barrier free authoring tool and a streaming media server.
- Blackboard [Blackboard] was founded in 1997 and provides course and content management systems, collaboration tools and a number of other services combined in the "Academic Suite" and the "Business Suite". It is one of the most popular and successful commercial e-learning systems. It can be extended according to own needs.
- Bodington [Bodington] is an open source LMS specialized on higher and further education developed by the University of Leeds. Bodington uses the metaphor of "buildings", "floors", and "rooms" to structure the Virtual Learning Environment (VLE). The main target is to be pedagogically flexible. In September 2006 the University of Oxford, the University of Cambridge, the UHI Millennium Institute and the University of Hull announced the "Tetra Collaboration" between Sakai and Bodington.
- BSCW [BSCW] (Basic Support for Cooperative Work) is a commercial shared workspace system mainly supporting advanced document management. Additionally it offers group and time management facilities as well as communication features like discussion boards, annotations and surveys. The project was initiated in 1995 and is still developed by FIT (Fraunhofer Institute of Technology) and OrbiTeam.
- CLIX [CLIX] is a commercial LMS developed by the imc (information multimedia communication) AG. It is available in different releases especially suitable for several different application scenarios.Additionally there are a couple of auxiliary features that can be added to the basic application in order to fit the individual needs of a scenario or project.
- Dokeos [Dokeos] is a quite complex e-learning and CM system and evolved out of the LMS "Claroline".

Most parts of the software can be downloaded for free, whereas others are offered on a commercial basis by the like-named company. In terms of adaptivity Dokeos provides progress-based learning paths (teachers may define prerequisites for items).

- Ilias [Ilias] is a service-oriented open source LMS, whose first prototype was developed within the VIRTUS project in 1997/1998 at the University of Cologne. In 2000 Ilias became an open source software. Currently, it is being developed by a collaboration network of several universities and companies.
- InterWise [InterWise] is a commercial conferencing and collaboration tool. It provides mainly synchronous possibilities of interaction including audio and video conferencing, desktop sharing, instant messaging, whiteboard, etc. Although it is no traditional learning platform, but more a conferencing tool, its main focus lies on e-learning (primarily in companies). InterWise provides virtual classrooms with possibilities going further than those of usual conferencing systems, e.g. by implementing different roles and the possibility to pose questions and receive statistics on the answers.
- Moodle [Moodle] is a very popular free Course Management System (CMS) that has its origins in the 1990ies. In 2003 the company moodle.com was launched to provide commercial support, managed hosting, consulting and other services. Since 2005 there is a fixed team of lead developers employed by Moodle, in addition to a large community of developers and supporting organisations contributing source code, ideas, etc. to the project. The general design tries to consider pedagogical principles and learning theories. The lesson module of Moodle also provides different learning paths. As the user's possible answers on a question can be used as starting points for different learning paths, some kind of "weak adaptivity" is supported (depending on the definition of adaptivity - as there is no user model).
- The OLAT [OLAT] (Online Learning And Training) project was started in 1999 at the University of Zürich. OLAT is a free LMS that is, since 2001, officially supported by the IT Department of the University of Zürich. In 2004 OLAT became open source.Today further development ist still lead by the University of Zürich, commercial support for the LMS is offered by various companies.
- OpenUSS with Freestyle Learning [OpenUSS] was developed by the University of Münster (starting in 2000). According to the website [OpenUSS] "Freestyle Learning (FSL) and Open University Support System (OpenUSS) are specifications for Learning Content System (LCS) and Learning Management System (LMS). They provide J2SE, J2ME and J2EE reference implementations on those specifications". OpenLMS is now also collaborating with OpenUSS.
- Sakai [Sakai] is a service-oriented Java-based open source LMS developed in 2004 by the universities of Michigan, Indiana, Stanford and the Massachusetts Institute of Technology. They contributed their existing LMSs to the new e-learning platform. Later other projects and partner institutions joined the Sakai community and developed Sakai tools based on their products (e.g. OSPortfolio, Samigo, Melete). Today Sakai is developed by 116 cooperating organizations and funded via a partners program.
- WebCT [WebCT] was a commercial Course Management System created in 1996 at the University of British Columbia. In 2006 WebCT was acquired by Blackboard [Blackboard], but it is still in use.

Unfortunately these systems provide no or just weak adaptivity features. Although adaptivity has been a research topic for about fifteen years, it is still used mainly in research projects rather than in the most frequently used LMSs (see table in figure 1).

3.2 Adaptive E-Learning Systems

The previous section provided a brief overview of popular, widely used e-learning systems. This section focuses on well-known historical and modern adaptive e-learning systems instead, which, while not as popular, have extensive support for adaptivity; most of these systems provide both adaptive presentation support and adaptive navigation support.

- AHA [AHA] is an open Adaptive Hypermedia Architecture providing adaptive content presentation based on fragments as well as link annotation and link hiding [De Bra and Calvi, 1998]. The current version is based on AHAM (Adaptive Hypermedia Application Model) [De Bra *et al.*, 2002]. User Model and Adaptation Engine are strictly separated.
- ALFANET [ALFANET] (Active Learning For Adaptive Internet) was developed within a European project from May 2002 to April 2005. Its architecture is service-oriented, uses multi-agent technology and is based on several standards [Santos *et al.*, 2004] (e.g. IMS-LD, IMS-QTI, IMS-CP, IEEE-LOM, IMS-LIP).
- ANATOM-TUTOR [Beaumont, 1994] is an adaptive system for teaching anatomy. It can be used in three different modes: browsing mode (without any adaptivity), question mode (using the user model extensively to find questions and to evaluate the answers) and hypermode (adaptive presentation and navigation support).
- AnnotatEd [Farzan and Brusilovsky, 2006] is an adaptive tool for annotating web pages. Based on the annotations (peer review) this tool is able to provide social navigation support. AnnotatEd can be used in combination with Knowledge Sea (see below).
- CHEOPS [Negro *et al.*, 1998] uses an internal knowledge model to provide adaptivity and is implemented as a set of CGI-BIN PERL scripts. The main information taken into account is the history of visited pages. Moreover, the system allows annotations on specific pages.
- ELM-ART [Weber and Brusilovsky, 2001] provides information as an interactive adaptive textbook and uses a combination of an overlay model and an episodic student model to provide adaptive navigation support, course sequencing, individualized diagnosis of student solutions, and example-based problem-solving support.
- EPIAIM [De Rosis *et al.*, 1993] is used for statistics in epidemiology and generates user-taylored messages.

The main focus of this adaptive system lies on the generation of natural language based on the experience of a user within a certain knowledge domain.

- HYPADAPTER [Böcker *et al.*, 1990] supports exploratory learning in the domain of Common Lisp and offers adaptive presentation support as well as adaptive navigation support (sorting, hiding, annotating).
- InterBook [InterBook], [Brusilovsky *et al.*, 1998] is an authoring tool for the development and delivery of adaptive electronic textbooks that transforms plain text to specially annotated HTML. It includes a web server for the publication of the textbooks, stores an individual model for each user and provides adaptive guidance, adaptive navigation support, and adaptive help.
- ITEM/IP [Brusilovsky, 1992] (Intelligent Tutor, Environment and Manual for Introductory Programming) supports a course on introductory programming based on the minilanguage Turingal. "The mini-language serves as a tool in mastering the main concepts of programming, programming languages' structures and skills in program design and debugging." ITEM/IP consists of several interacting components providing support for the different phases of the learning process: the pedagogical module (enhancing the choice of teaching operations), the programming laboratory (enabling students to work independently) and the information kernel (including all factual knowledge).
- iWeaver [Wolf, 2003] is a PhD project designed to provide an adaptive, flexible learning environment for the Java programming language. The system creates a learner profile by assessing learning styles with the help of a range of multiple choice questions when the user first enters. Later users receive personalized recommendations and an individual view of the available learning tools. iWeaver combines adaptive navigation and adaptive content presentation techniques.
- KN-AHS [Kobsa *et al.*, 1994] is an adaptive hypertext client for the user modeling system BGP-MS. It provides automatic adaptation of hypertext to a user's state of domain knowledge. KN-AHS draws assumptions about the user's knowledge by an initial interview and some of the hypertext actions the user performs. When a new concept is introduced, its presentation is adapted to the user's familiarity with it, e.g. by offering additional explanations after having retrieved the respective information from the related user model.
- KnowledgeSea II [Brusilovsky *et al.*, 2006] is a system for personalized information access. It offers various methods of accessing information, including two-level visualization, hypertext browsing, recommendation and social search. Personalization is provided by social navigation support which is an approach for browsing-based and recommendation-based information access. KnowledgeSea II includes an adaptive search facility combining a common vector search engine and social navigation. By that every user may benefit from the whole community's knowledge; search results are adapted to the user based on the history of activities.
- KnowledgeTree [Brusilovsky, 2004] is a distributed architecture for adaptive e-learning based on the reuse of intelligent educational activities. It combines learning content and learning support devices and presumes the existence of at least four kinds of communicating servers: activity servers, value-adding servers, learning portals (e.g. KnowledgeSea) and student modeling servers.
- MetaDoc [Boyle and Encarnacion, 1994] introduces an adaptive presentation technique based on stretchtext. It handles nodes as stretchtext pages and presents a requested page collapsing all extensions not relevant and uncollapsing all extensions relevant for the user.
- METOD [METOD] (MetaTool for Educational Platform Design) is a European Union funded project basically aiming at creating a general paradigm for educational platform development. Part of the project's results is MetaTool that allows creating METOD projects storing various kinds of content and (meta) information, e.g. topics, student types, learning styles, exercises and learning paths. The projects can then be exported to various Content Management Systems that have to support a specific METOD plugin in order to provide adaptive learning.
- NetCoach [NetCoach] is a further development of ELM-ART containing an own authoring system that allows the development of adaptive courses. Generally all material belonging to a course is organized in a tree structure and can be freely browsed by the learner. Additionally, the system offers personalization of courses by adaptive curriculum sequencing and adaptive link annotation.
- SQL-Tutor [Mitrovic and Ohlsson, 1999] combines Intelligent Tutoring and Adaptive Hypermedia in a constraint-based architecture. It provides support for university-level students learning SQL and consists of an interface, a pedagogical module and a student modeling unit analyzing students' answers.

4 Discussion and Conclusion

The previous section shows that over the past two decades a lot of effort was put into exploring and researching the benefits of adaptivity in e-learning. Therefore a large number of research projects and systems (including the ones mentioned in section 3.2) already uses adaptivity.

Unfortunately, none of these systems is already being used by a large and worldwide community outside the research area. Most of the popular e-learning platforms have not yet taken advantage of adaptivity, possibly because the expected profit does not yet justify the high effort of implementing and authoring adaptive courses. Moreover, most adaptive systems do not support e-learning standards [Paramythis and Loidl-Reisinger, 2004].

Table 1 presents an overview on features supported by some representative systems (listed in section 3).

Within this table we may identify two groups of systems. The first group – systems mentioned in section 3.1 – supports a lot of "standard features" a learning platform is expected to include, but as already mentioned they do not provide adaptivity features. The second group – systems mentioned in section 3.2 – provides these features, but as many of the "standard features" are missing, they are rather not suitable for common e-learning scenarios.

The next step in order to make the knowledge and experience gained in research projects on adaptivity available

	Bodington	Dokeos	Moodle	OLAT	OpenUSS	Sakai	AHA	Alfanet	ELM-ART	InterBook	KnowledgeSea	METOD	NetCoach
Learning													
Assignments – Choice	X	X	X	X	X	X	X	X	X	X		X	X
Assignments – Text	X	X	X	X	X	X	X	X	X	X		X	X
Assignments – Files	X	X	X	X		X	X					X	
Stored Evaluation	X	X	X	X	X	X	X	X	X	X		X	X
Glossary			X	X	X	X			X	X			X
Bookmarks	X			X									
Annotations	X			X	X								X
Search	X		X	X	X	X				X			X
Cooperation													
Forum	X	X	X	X	X	X		X					X
News	X		X	X	X	X		X					X
Email	X		X	X	X	X							X
Personal Messages			X		X			X					
File Management	X	X	X	X	X	X		X				X	X
Calendar	*)	X	X	X	X	X		X					
Polls	X	X	X	X		X							
Nickpage Authoring	X	X	X	X	X	X		X					
Collaboration													
Chat	X	X	X	X	X	X							X
Audio Conferencing		X	*)		*)	*)							
Video Conferencing		X	*)		*)	*)							
Whiteboard		X			X	*)							
Desktop Sharing					*)	*)							
Learning Management													
User Group Support	X	X	X	X		X				X	X	X	X
Creating Content Online	X	X	X	X	X	X	X	X				X	X
Importing Content	X	X	X	X	X	X		X	X	X		X	X
Stored Learner Model							X	X	X	X	X	X	X
Personalization	X	X	X	X	X			X					X
Adaptivity													
Adaptive Guiding							X	X	X	X	X	X	X
Link Annotation							X	X	X	X	X	X	X
Link Hiding							X	X	X	X	X	X	X
Adaptive Pres. Support							X	X	X	X		X	X
Standards													
IMS CP	*)	X	X	X		X		X					
IMS QTI	X	X	X	X		X		X					
SCORM	*)	X	X	X	*)	X	X[1]	X					
IEEE LOM		X	X	*)		X		X					

X...Supported *)...Planned to be supported

[1] not part of the original system, but added as an extension

Figure 1: Features of e-learning systems

to a large community of learners would be their combination with commonly required features. As research projects usually neither aim at the implementation of already widely used features nor have the capacities and resources to redevelop them, a better approach would be the transfer of achievements in the field of adaptivity into the development of the large and most frequently used systems.

Acknowledgements

The work reported in this paper has been partially funded by the Socrates Minerva “Adaptive Learning Spaces” (ALS) project (229714-CP-1-2006-1-NL-MPP). For information on the ALS project please refer to the project’s web site: http://www.als-project.org.

References

[Aroyo *et al.*, 2006] L. Aroyo, P. Dolog, G.-J. Houben, M. Kravcik, A. Naeve, M. Nilsson, and F. Wild. *Interoperability in Personalized Adaptive Learning*. In *Educational Technology & Society*, 9 (2), pages 4–18, 2006.

[Beaumont, 1994] I. Beaumont. *User modeling in the interactive anatomy tutoring system ANATOM-TUTOR*. In *User Models and User Adapted Interaction*, 4(1), pages 21–45, 1994.

[Beaumont and Brusilovsky, 1995] I. Beaumont and P. Brusilovsky. *Educational applications of adaptive hypermedia*. In *INTERACT*, pages 410–414, 1995.

[Böcker *et al.*, 1990] H.-D. Böcker, H. Hohl, and T. Schwab *Hypadapter - Individualizing Hypertext*. In D. Diaper, D. J. Gilmore, G. Cockton, and B. Shackel, editors. *Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction*, pages 931–936, Amsterdam, The Netherlands, North-Holland Publishing Co., September 2005.

[Boyle and Encarnacion, 1994] C. D. B. Boyle and A. O. Encarnacion. *Metadoc: An Adaptive Hypertext Reading System*. In *User Model. User-Adapt. Interact.*, 4(1), pages 1–19, 1994.

[De Bra and Calvi, 1998] P. De Bra and L. Calvi. *AHA! An open Adaptive Hypermedia Architecture*. In *The New Review of Hypermedia and Multimedia*, 4, pages 115–139, Taylor Graham Publishers, 1998.

[De Bra *et al.*, 2002] P. De Bra and A. Aerts and D. Smits and N. Stash. *AHA! meets AHAM*. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 381–384, Springer LNCS 2347, May 2002.

[Brusilovsky, 1992] P. Brusilovsky. *Intelligent Tutor, Environment and Manual for Introductory Programming*. In *Educational and Training Technology International*, 29(1), pages 26–34, 1992.

[Brusilovsky, 1996a] P. Brusilovsky. *Adaptive hypermedia, an attempt to analyze and generalize*. In P. Brusilovsky, P. Kommers, and N. Streitz, editors. *Multimedia, Hypermedia, and Virtual Reality. Lecture Notes in Computer Science*, 1077, pages 288–304, Springer-Verlag, Berlin, Germany, 1996.

[Brusilovsky, 1996b] P. Brusilovsky. *Methods and Techniques of Adaptive Hypermedia*. In *User Modeling and User-Adapted Interaction*, 6(2-3), pages 87–129, 1996.

[Brusilovsky *et al.*, 1998] P. Brusilovsky, J. Eklund, and E. Schwarz. *Web-Based Education for All: A Tool for Developing Adaptive Courseware*. In *Proceedings of Seventh International World Wide Web Conference, WWW98*, pages 291–300 April 1998.

[Brusilovsky, 2004] P. Brusilovsky. *KnowledgeTree: a Distributed Architecture for Adaptive E-Learning*. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 104–113, 2004.

[Brusilovsky *et al.*, 2006] P. Brusilovsky, R. Farzan, and J.-W. Ahn. *Layered Evaluation of Adaptive Search*. In *Proceedings of Workshop on Evaluating Exploratory Search Systems, at SIGIR2006*, 2006.

[Encarnação, 1995] L. M. Encarnação. *Adaptivity in graphical user interfaces: An experimental framework*. In *Computers & Graphics 19*, (6), pages 873–884, 1995.

[Farzan and Brusilovsky, 2006] R. Farzan and P. Brusilovsky. *AnnotatEd: A Social Navigation and Annotation Service for Web-based Educational Resources*. In *Proceedings of E-Learn 2006–World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 2006.

[Fischer *et al.*, 1990] G. Fischer, T. Mastaglio, B. Reeves, and J. Rieman. *Minimalist explanations in knowledge-based systems*. In *Proceedings of the 23rd annual Hawaii international conference on system sciences, HICSS-23*, pages 309–317, Kailua-Kong, Hawaii, January 1990.

[Gonschorek and Herzog, 1995] M. Gonschorek and C. Herzog. *Using hypertext for an adaptive helpsystem in an intelligent tutoring system*. In *Proceedings of the World Conference on Artificial Intelligence in Education*, pages 274–281, August 1995.

[Henze, 2000] N. Henze. *Adaptive Hyperbooks: Adaptation for Project-Based Learning Resources*. PhD thesis, University of Hannover, 2000.

[Kareal and Kléma, 2006] F. Kareal and J. Kléma. *Adaptivity in e-Learning*. In *In Current Developments in Technology-Assisted Education*, pages 260–264, Formatex, 2006.

[Kay and Kummerfeld, 1994] J. Kay and R. J. Kummerfeld. *An Individualised Course for the C Programming Language*. In *Proceedings of the Second International WWW Conference "Mosaic and the Web"*, Chicago, USA, 1994.

[Kim, 1995] D.-W. Kim. *WING-MIT: Das auf einer multimedialen und intelligenten Benutzerschnittstelle basierende tutorielle Hilfesystem*. In *WING-IIR Technical Report 69*, University of Regensburg, Germany, 1995.

[Kobsa *et al.*, 1994] A. Kobsa, D. Müller, and A. Nill. *KN-AHS: An Adaptive Hypertext Client of the User Modeling System BGP-MS*. In *Proceedings of the Fourth International Conference on User Modeling, UM1994*, pages 99–105, 1994.

[Mitrovic and Ohlsson, 1999] A. Mitrovic and S. Ohlsson. *Evaluation of a Constraint-Based Tutor for a Database Language*. In *International Journal of Artificial Intelligence in Education*, 10, pages 238–256, 1999.

[Negro *et al.*, 1998] A. Negro, V. Scarano, and R. Simari. *User Adaptivity on WWW through CHEOPS*. In *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, HYPERTEXT98*, Pittsburgh, USA, 1998.

[Paramythis and Loidl-Reisinger, 2004] A. Paramythis and S. Loidl-Reisinger. *Adaptive Learning Environments and eLearning Standards*. In *Electronic Journal of e-Learning*, 2(1), pages 181–194, 2004.

[De Rosis *et al.*, 1993] F. De Rosis, N. De Carolis, and S. Pizzutilo. *User tailored hypermedia explanations*. In *Proceedings of INTERCHI'93 (Adjunct proceedings)*, pages 169–170, April 1993.

[Santos *et al.*, 2004] O. C. Santos, J. G. Boticario and C. Barrera. *The Standards-based Architecture of the Adaptive Learning Environment aLFanet*. In *WSEAS Transactions on Computers*, 3, pages 1814–1818, December 2004.

[Weber and Brusilovsky, 2001] G. Weber and P. Brusilovsky *ELM-ART: An Adaptive Versatile System for Web-based Instruction*. In *International Journal of Artificial Intelligence in Education*, 12, pages 351–384, 2001.

[Wolf, 2003] C. Wolf. *iWeaver: Towards 'Learning Style'-based e-Learning in Computer Science Education*. In *Proceedings of the fifth Australasian conference on Computing education*, Adelaide, Australia, pages 273–279, 2003.

[.LRN] *dotLRN*. http://www.dotlrn.org. last download: 10.7.2007.

[AHA] *AHA project*. http://aha.win.tue.nl. last download: 16.7.2007.

[ALFANET] *ALFANET*. http://rtd.softwareag.es/alfanet. last download: 16.7.2007.

[ATutor] *ATutor Learning Content Management System*. http://www.atutor.ca. last download: 16.7.2007.

[Blackboard] *Blackboard - Educate. Innovate. Everywhere*. http://www.blackboard.com. last download: 16.7.2007.

[Bodington] *Bodington.org :: Home*. http://bodington.org. last download: 16.7.2007.

[BSCW] *BSCW Home Page*. http://bscw.fit.fraunhofer.de. last download: 16.7.2007.

[CLIX] *IMC - Lernplattform CLIX, Rapid Authoring LECTURNITY, LMS und eLearning Lösungen*. http://www.im-c.de. last download: 16.7.2007.

[Dokeos] *dokeos Open Source e-Learning*. http://www.dokeos.com. last download: 16.7.2007.

[Ilias] *ILIAS open source*. http://www.ilias.de. last download: 16.7.2007.

[InterBook] *InterBook*. http://www2.sis.pitt.edu/˜peterb/InterBook.html. last download: 17.7.2007.

[METOD] *METOD*. http://idec.gr/metod and http://metod.uni-mb.si. last download: 17.7.2007

[Moodle] *Moodle - A Free, Open Source Course Management System for Online Learning*. http://www.moodle.org. last download: 16.7.2007.

[InterWise] *Interwise: Unlimited Voice, Web and Video Conferencing for the Enterprise*. http://www.interwise.com. last download: 16.7.2007.

[NetCoach] *ORBIS AG: ORBIS NetCoach*. http://www.orbis.de/netcoach. last download: 16.7.2007.

[OLAT] *Open Source LMS OLAT | About OLAT*. http://www.olat.org. last download: 16.7.2007.

[OpenACS] *OpenACS Home*. http://www.openacs.org. last download: 16.7.2007.

[OpenUSS] *CampusSource - Software - OpenUSS & FSL*. http://www.campussource.de/software/openuss. last download: 16.7.2007.

[Sakai] *sakaiproject.org*. http://www.sakaiproject.org. last download: 16.7.2007.

[WebCT] *WebCT.com*. http://www.webct.com. last download: 16.7.2007.

Context-adaptation based on Ontologies and Spreading Activation

Tim Hussein, Daniel Westheide, Jürgen Ziegler
University of Duisburg-Essen
Lotharstr. 65, 47057 Duisburg, Germany
{hussein,westheide,ziegler}@interactivesystems.info

Abstract

Ontologies and spreading activation are known terms within the scope of information retrieval. In this paper we introduce SPREADR, an integrated adaptation mechanism for web applications that uses ontologies for representing the application domain as well as context information like location, user history and local time. Those context factors can be modeled in an ontology and be linked to certain domain nodes. In each session a *Spreading Activation Network* is build based on those ontologies and recognized context factors or user actions can trigger an activation flow through this network. A node's resulting activation value then represents its importance according to the current circumstances. While identically in structure, the Spreading Activation Networks are personalized by automatically modifying link weights and activation levels of nodes. As a result the system learns about the user preferences and can adjust its adaptation mechanism for future runs through implicit feedback.

1 Introduction

The increasing amount of information presented in current web based applications often makes them difficult to use. Finding the appropriate content within the flood of data can be challenging and may cause a user to reject a web application. Various approaches have been proposed to overcome these problems by adaptation, each with its particular advantages and drawbacks. Roughly, two approaches can be distinguished: Context-aware and self-adaptive systems. Context-aware systems adapt the content according to the current circumstances, but usually have no learning mechanism. Ideally a system should continuously observe the user's behavior in order to learn from his decisions and thereby improve future adaptations. This is what self-adaptation is about.

Like [Dey *et al.*, 2004] we propose an integrated view of context and domain information to contextually offer the appropriate content. For this purpose we make use of two concepts originally known within the scope of artificial intelligence and information retrieval: Ontologies and spreading activation.

In Section 2 we present existing approaches that cover context engineering and adaptation techniques. Subsequently, we illustrate our context engineering approach in Section 3 and show a method to integrate contextual information into an existing domain ontology. We propose a method for creating an integrated ontological model for each user representing the current usage context. A modified spreading activation algorithm is then used to adjust those networks according to the user interaction and to refine his profile over time. Simultaneously, self-adaptation is performed by adjusting the adaptation mechanism itself due to implicit relevance feedback. This technique is explained in Section 4, followed by an outline of the system architecture in Section 5 and a description of a prototypical implementation and its evaluation in Section 6. We conclude the article in Section 7, summarizing our conclusions and pointing out future directions for research.

2 Related work

Various authors such as [Middleton *et al.*, 2004] proposed that adaptation by continuous observation is desirable in order to learn from the user's behaviour and the circumstances under which this behaviour occurs. While user modeling and recommendation techniques have been the focus of research for a long time, there have been few attempts that take contextual information into account for purposes of personalization ([Herlocker and Konstan, 2001], [Adomavicius and Tuzhilin, 2001] and [Kovacs and Ueno, 2006]). Current systems mostly concentrate on the user's transaction history, which is of course an important factor for adaptation. Those systems are usually called *user adaptive*. Within applications that are always used in the same context, this is not a problem at all, but with increasing complexity of web applications context becomes a substantial factor in terms of usability.

It can be assumed that there is always a contextual background for the user's information and service needs. Most of the time the circumstances have a crucial impact on our decisions as we always act in different roles. Winter coats might be interesting in November, but not in July. One would only like to know about the cafeteria menu of the day on working days. So a system should automatically adapt itself to the particular context to present the user ”the right thing at the right time in the right way” [Kappel *et al.*, 2003]. Those systems are not *user adaptive* but *context adaptive*. Traditional context-aware scenarios focus on single context factors like the user's current location [Chen and Kotz, 2000], for instance for presenting tourist information [Cheverst *et al.*, 2000]. [Henricksen and Indulska, 2005] calls the field of context-aware computing ”immature”. Though this is a hard judgement we agree that context-adaptivity still has a long way to go. Especially there is a lack of approaches that take the user history into account as well as multidimensional context information.

Recent activities include the *a CAPella* system [Dey *et* *al.*, 2004] which can be "trained" by the user to automatically recognize certain events depending on the current context via multi-modal sensing: Information obtained from a microphone, a camera, RFID antennas and other devices is being used and interpreted. As a result, *a CAPella* for instance recognizes the start of a meeting and automatically presents certain documents that have been used in a similar context in the past. This interesting and promising approach is a good example for sensing and unifying context information. However it strongly focusses on real-world interaction and needs certain external equipment for context sensing, which makes it not directly applicable for the purpose of web engineering.

Collaborative Filtering techniques – well known from the field of recommender systems – have some qualities that make them a good choice for filtering semantically untagged items. This approach is very popular, because it leads to notable results especially in recommender systems and frees the web engineer from creating and maintaining complex content tagging. Yet pure collaborative filtering has some crucial disadvantages, too, like the *ramp-up-problem* often referred to as explained in [Burke, 2002]. Combining collaborative filtering techniques with content based mechanisms has been shown to be a feasible solution to the ramp-up-problem. In [Claypool *et al.*, 1999] and [Melville *et al.*, 2002] different approaches are presented. However, these solutions are rather user centered and do not adequately take the particular context into account. Nonetheless, collaborative filtering algorithms have been proven to be a simple but effective instrument that can be integrated into more sophisticated systems. We think that there is a strong need for an integrated strategy that considers as much context information as possible: The current usage context in terms of situation-awareness as well as the past interactions including the contexts for the time being.

Ontologies have been proven to be a good choice for knowledge representation. Having its roots in philosophy, ontologies have become popular for computer science since the 1990s [Neches *et al.*, 1991]. Ontologies can be used to represent manifold information in a human-understandable and machine-readable format consisting of entities, attributes, relationships and axioms [Guarino and Giaretta, 1995]. Examples for using ontologies in recommender systems can be found in [Middleton *et al.*, 2004], where *Quickstep* and *Foxtrot* are illustrated – two systems that make use of ontologies to recommend scientific research papers a particular user might be interested in. We think that in a truly adaptive system the adaptation techniques have to become part of an optimization process itself through learning mechanisms. So the challenge is to close the gap between user adaptive and context adaptive systems [Oppermann, 2005] and to provide in this sense a holistic system with appropriate learning mechanisms. In this paper we want to introduce SPREADR (Spreading Activation Driven Reasoning) - a model-driven, user- and context-adaptive solution for this problem.

3 Context engineering

Developing a web application with SPREADR implies the creation of several models. A typical scenario is presented in Section 6. The models that can be defined in SPREADR for instance include a domain and a context model. Each of these models plays a distinct role in building the content presented to the user and is an ontology represented in the Web Ontology Language (OWL) format.

Within an ontology, nodes are semantically related to each other. As an extension of this basic mechanism, we assign individual weights to the relations. The resulting networks are identical in structure for each user, but the weights of the links are individualized. Nodes in the ontology (concepts and instances) are treated in a similar way by assigning activation values to them. A fact in reality is valid for everyone, but it is more or less relevant for a certain user. If one considers an item to be important it receives a higher activation value. Thereby, we can create individual weighted networks for each user to represent his personal interests. Information regarding his current location, device, local time etc. can be handled just as well: By raising the activation of the corresponding nodes we can represent an individual usage context and by individualizing the relation weights certain factors are more or less tied to a certain concept for each user.

The domain model

Generally speaking, the purpose of the domain model is to represent knowledge relevant to the respective application as well as pieces of content or references to them, depending on the type of content that is represented [Kaltz, 2006]. To represent relevant domain knowledge, the domain model contains classes and semantic relations between them. Instances of those classes are likewise part of the domain model, linked to each other by appropriate relations. A domain model should be created by domain experts without considering the context-adaptations that are supposed to take place [Kaltz, 2006].

The context model

When building a model-driven context-aware web application, it is not only necessary to model the domain, but also the context. Here, context is "any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves" [Dey, 2001].

Thus, the quality of a context-aware web application depends on the relevance of the modeled context. In order to reduce complexity, we distinguish between five categories of context, as proposed by [Ziegler *et al.*, 2005]:

- *User and role:* individual users or groups of users that are defined according to their different roles.
- *Task:* task-oriented context, e.g. work assignments or a user's personal goals.
- *Location:* the user's physical or virtual location (e.g. internet vs. local area network).
- *Time:* e.g. the season, the weekday or the time of day.
- *Device:* The user's device, e.g a PDA, a mobile phone or a personal computer.

Each of the these context categories is modeled in an ontology containing all context factors the system is supposed to be aware of. At runtime, contextual information is recognized (Section 5) and the respective context factors are activated.

However, for the context to have an effect on the content selection, it is also necessary to model appropriate *context relations*. A context relation defines a link between a context factor and an item from the domain model as well as a relevance weight. This relevance weight is used to determine the general importance of specific domain items in a given context.

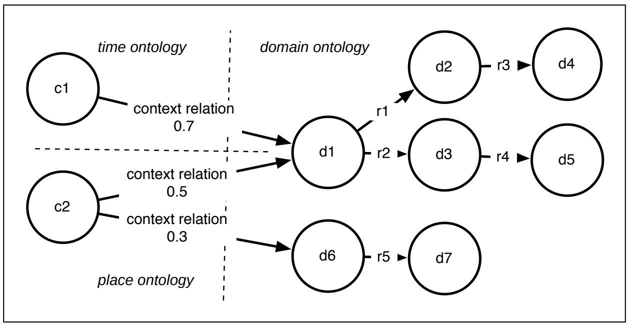


Figure 1: Connecting context factors to the domain model

Usually, when modeling a context relation, it is the context engineer's task to define a relevance weight that reflects the importance of the relation as seen by the majority of users. Therefore, profound knowledge about the target group of the web application is required so as to satisfy the needs of most of the users. Especially when building web applications for a heterogeneous user group, this approach of defining static relevance weights that are valid for all users is problematic. Therefore an adaptive system requires appropriate learning mechanisms. We approach this problem by automatically adjusting the weight of a context relation for an individual user according to his interaction with the system and thus deviate from the weight defined in the context model (see Section 4).

4 Adaptation by spreading activation

In our system, activation is spread within the domain model, starting with items that are related to currently activated context factors. This way we can exploit existing relations within the domain and reduce complexity for the context engineer. The concept of spreading activation traces back to [Collins and Loftus, 1975]. Their model of spreading activation networks was originally applied in the fields of psycho linguistics and semantic priming [Anderson, 1983]. Later, the idea was adopted by computer scientists: Spreading activation techniques have successfully been used in several research areas in computer science, most notably in information retrieval ([Cohen and Kjeldsen, 1987], [Crestani, 1997] and [Berger *et al.*, 2004]). The principles of spreading activation have also been used by [Pirolli and Card, 1995] in their information foraging theory. [Kovacs and Ueno, 2006] extend classical associative spreading activation networks with "link types" and "context nodes" to generate context-adaptive recommendations. An interesting approach for using spreading activation to explore transitive associations between users can be found in [Huang *et al.*, 2004], who use this strategy to avoid the sparsity problem in collaborative filtering.

Several algorithms have been developed to implement the concept of spreading activation. However, the general idea is the same: At the beginning, one or more nodes are activated. These are called *initial nodes*. From these initial nodes, activation is propagated through the network. Once triggered the so called *pulse* passes through adjacent nodes – thereby amplifying them – until a certain termination condition is met.

4.1 The branch-and-bound algorithm

Three algorithms often used for spreading activation are examined in [Huang *et al.*, 2004]: *Constrained leaky capacitor* (originally proposed by [Anderson, 1983]), *Hopfield nets* and *Branch-and-bound*. For performance reasons we chose the branch-and-bound algorithm and our implementation of this algorithm is as follows:

Initialization: Before the actual execution of spreading activation begins, the network must be initialized:

1. The weights for the links are set based on the user's individual context model. Moreover, in our approach, the network is not necessarily in a blank state when a spreading activation run starts. Therefore, initial activation levels for each node in the network are set. These are based on the resulting activation levels of the previous run.
2. The initial nodes are activated with a certain value. The activation received by the start nodes is added to their previous state. Optionally the new activation level is calculated by applying an activation function to this sum.
3. The initial nodes are inserted into a priority queue ordered by descending activation.

Execution: After initialization, the following steps are repeated until a defined termination condition is fulfilled or the priority queue is empty. The termination condition can be configured freely, but two pre-defined termination conditions are provided: (1) A maximum of *activated* nodes is reached, (2) a maximum of *processed* nodes is reached. A processed node is a node that has itself propagated activation to adjacent nodes.

1. The node with the highest weight is removed from the queue.
2. The activation of that node is passed on to all adjacent nodes, if this is not prevented by some restriction imposed on the spreading of activation. If a node j receives activation from an adjacent node i, a new activation level is computed for j.

   $$A_j(t+1) = A_j(t) + O_i(t) \times w_{ij} \times a$$

   where $A_j(t)$ is the previous activation of j, $O_i(t)$ is the output activation of i at the time t, w_{ij} is the weight of the relation between i and j and a is an attenuation factor. The output activation of a node is the activation it has received. An arbitrary function can be used to keep the values in a predefined range.
3. The adjacent nodes that have received activation are inserted into the priority queue unless they have already been marked as processed.
4. The node that passed on its activation to the neighboring nodes is marked as processed.

Our spreading activation mechanism provides several parameters that allow the context engineer to modify its behavior, depending on the desired results and the domain. In order to prevent activation from spreading through the whole network and eventually activating every single node, we make use of constrained spreading activation: Depending on the concept type, the outgoing edges, the path-distance between nodes the spreading activation process can be influenced. Details about those constraints can be found in [Cohen and Kjeldsen, 1987] and [Rocha *et al.*, 2004]. Additionally, the attenuation factor used in the input function can be configured and reverberation can be prevented. This means that a node j must not propagate activation to a node i if node j has itself been activated by node i before in the same run. Finally, our spreading activation mechanism allows the adjustment of relation type weights. A relation type weight is used for each relation for which no individual weight has been set in the initialization phase of the algorithm.

4.2 Context reasoning

When a new session starts, a user specific Spreading Activation network is being created from the various models. In its structure and semantics those networks are the same for each user. It is individualized by adding numerical values to it: Each node receives a specific activation that represents the importance of that domain item for the respective user. In addition to their semantics, the relations also receive weights for the individual user. If a relation r between two nodes i and j has a high value this means that the relation is very important for the user. For other users the same relation r may be completely irrelevant.

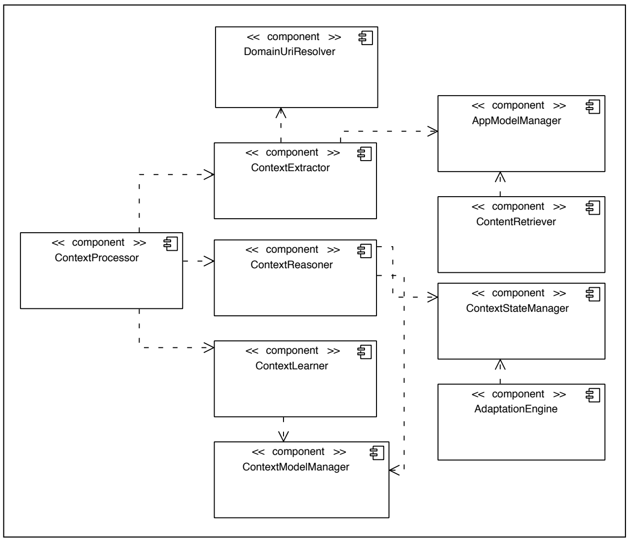
At the beginning, each node has an initial activation value of 0. Upon a request by a user, the requested content node and the sensed context factors are being used as initial nodes for the spreading activation process. As a result certain concepts and instances that seem to be important in this particular context can be used for adaptation effects. Furthermore the resulting values are being saved and can be considered for future adaptations - some of them only within the current session, some permanently. Indeed, it is neither appropriate nor does it reflect reality to let the activation rise monotonously during the entire period of usage. Because of this, we implemented a slight decay of all node weights that takes place at frequent intervals. Hence, a certain weight for a domain item can only be maintained if it is regularly activated – either directly if the user clicks on the appropriate navigation node, or indirectly by spreading activation from a related domain item.

4.3 Learning by adjusting the relation weights

As already mentioned, our system does not only manage separate node weights for each user, but also separate relation weights. A relation weight represents the importance that the relation has for an individual user. Initially, the relevance weight of a context relation is defined globally in the context model. Thus, it is the same for each user first. Adjusting the weight of a relation within the domain model is similar to the adjustment of context relation weights. When a spreading activation run is performed, each node stores the path to the initial node whose propagation led to its activation – together with information on how much activation it has received via that path. If within a certain amount of requests the user navigates to a domain item i that has previously been activated in a spreading activation run via the path p, the importance of the relations that form the path p are increased. Relation weight adjustments are stored permanently, so that the system learns what is important in a specific usage context. If the user does not "confirm" the activation path within a certain period of time by requesting the recommended node, the relation is considered to be not very important and therefore decreased in weight. This idea was inspired by Hebbian Learning [Hebb, 1949]. By doing so the spreading activation process itself becomes part of the adaptation process.

5 Architecture

Whenever the system recognizes certain context factors, activation energy is injected into the context model in order to activate relevant domain items. Based on this information, adaptations of content or navigation can be initiated. The framework focuses on context recognition, context reasoning, learning about context, and providing services for adaptation to context. The actual generation of the web pages that are sent to the browsers is not part of it, which allows for a maximum freedom concerning the technologies for generating pages. Generally, most of the main components of SPREADR can be assigned to either of two tasks: context processing or response generation. However, for some of the components, such a clear-cut classification is not possible because they are involved in both tasks. Figure 5 shows the main components as well as the dependencies between them.



<< component >> DomainUriResolver
<< component >> AppModelManager
<< component >> ContextExtractor
<< component >> ContentRetriever
<< component >> ContextProcessor
<< component >> ContextReasoner
<< component >> ContextStateManager
<< component >> ContextLearner
<< component >> AdaptationEngine
<< component >> ContextModelManager

Figure 2: Components in SPREADR

5.1 Context processing components

Context Processor

The ContextProcessor is responsible for initiating and controlling the context processing of a request. Upon each request to the server, the ContextProcessor must be provided with a HttpServletRequest object that represents the current request in order to start the context extraction, reasoning, and learning process. Therefore, it contains references to the ContextExtractor, the ContextReasoner, and the ContextLearner component (if learning of context models is desired).

Context Extractor

The first component called by the ContextProcessor upon a new request is the ContextExtractor. Its task is to extract context information from the data that is available in an HttpServletRequest object and to return the extracted context information as a collection of context factors, each of them assigned to one of the context categories described in Section 3 and has an activation level in the interval $[0, 1]$. In its default implementation, the ContextExtractor delegates its task to sub-components, each of them being responsible for extracting context factors of a single category. Once the ContextProcessor has received the extracted context factors from the ContextExtractor, it hands them over to the ContextReasoner.

Context Reasoner

This component is responsible for activating additional context factors, based on past context states and on the extracted context factors, for instance by combining context factors from several categories in order to infer additional context factors. However, although this can easily be changed by implementing an alternative ContextReasoner, the core reasoning is achieved by spreading activation in the context model. Access to the context model is provided by the ContextModelManager (see below). The spreading-activation based reasoning is performed by a corresponding sub-component. First and foremost, it is used to activate context factors representing domain items that are relevant in the current context. When the ContextReasoner has finished, it passes the context state, an artifact of its work that consists of all currently active context factors, to the ContextStateManager. Moreover, it tells the ContextModelManager to save the context model.

Context State Manager

The ContextStateManager manages past and present context states for an individual user and enables other components to access this information.

Context Learner

After context reasoning, context learning can optionally be initiated by the ContextProcessor. To do so, the latter calls the ContextLearner which accesses a context model from the ContextModelManager, modifies the model in some way and asks the ContextModelManager to save the changed model.

Context Model Manager

It is the ContextModelManager, that is responsible for managing the context model for a specific user. It provides access to the context model and, if asked to do so, persists the context model, so that potential changes that have been made to it by the Context Learner are not lost after the session has terminated. While this component is intended to provide user-specific context models, alternative implementations might provide the same model to all users. This is reasonable if no Context Learner is used.

5.2 Response generation components

While the functionality of SPREADR does not comprise the actual page generation, it provides some components that are useful for accessing the content to be rendered: the *DomainUriResolver*, the *AppModelManager*, the *ContentRetriever*, and the *AdaptationEngine*.

Domain URI Resolver

The DomainUriResolver is a utility component, that resolves the requested URI to an item or concept in the domain ontology. When using a MVC framework, a Controller can use this information to provide the view with the information that is necessary to render the requested domain item. The DomainUriResolver is also used by the ContextExtractor to extract the appropriate context factor in the context category application.

Application Model Manager

Access to the ontologies is provided by the AppModelManager.

Content Retriever

The ContentRetriever is a special component because implementations of it must be developed for the individual web application. It can then be called to get the information that is to be displayed, encapsulated in an instance of a class adhering to the JavaBean specification.

Adaptation Engine

It is the task of the AdaptationEngine to provide content and services to components that are responsible for page generation. Currently, its sole functionality is to provide items that are relevant in the current context.

6 Experiences

In order to test the effectiveness of our adaptation mechanism and to clarify our methodology, we developed an adaptive music portal. This is a typical scenario where adapting to the user and his current context is often considered to make sense. People often have a small number of favourite artists but are not aware of other artists they might like, do not notice dates of interesting concerts taking place close to their current location, or that the music they are interested in is dependent on context such as time. In our scenario, we target these problems by adapting the content of the portal to the current usage context, i.e. to the user profile enriched with activations of items by the current context. Our music portal provides album reviews, artist biographies, concert information and several kinds of additional information about events and items.

For evaluation purposes we created 4 typical usage scenarios that had to be simulated by various users. Each of the scenarios contained a part with context learning enabled (via relation weight adjustments) and a part without. The users did not know about those technical details and had to rate the quality of the adaptation effects. Those effects have been rated considerably better when context learning was enabled, because in that case they were able to find interesting items with significantly less clicks.

7 Conclusions

We introduced a novel approach to determine the most important elements of a given ontology with regard to current context. On this basis adaptation activities can automatically be performed. We call this a holistic spreading activation technique, because it relies entirely on the results of the spreading activation processes. Furthermore context relations are fully integrated into the propagation process and thereby affect the adaptation activities. Our goal was to design an adaptation mechanism for context-adaptive web applications including appropriate learning mechanisms to close the gap between context-awareness and self adaptation. [Perkowitz and Etzioni, 2000] call those web sites adaptive, which "automatically improve their organization and presentation by learning from visitor access patterns". The proposed system meets this requirements. As for future directions, our short-term goal is to cluster user profiles and thus allow cross-network-propagation of activity action, whereas the design of truly information-centered applications with context-adaptive interfaces may be a possible long-term goal.

References

[Adomavicius and Tuzhilin, 2001] Gediminas Adomavicius and Alexander Tuzhilin. Multidimensional recommender systems: A data warehousing approach. *Lecture Notes in Computer Science*, 2232, 2001.

[Anderson, 1983] John R. Anderson. A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 22:261–295, 1983.

[Berger *et al.*, 2004] Helmut Berger, Michael Dittenbach, and Dieter Merkl. *An Adaptive Information Retrieval System Based on Associative Networks*, volume 31 of *Conferences in Research and Practice in Information Technology*. ACS, Dunedin, New Zealand, 2004. First Asia-Pacific Conference on Conceptual Modelling (APCCM2004).

[Burke, 2002] Robin D. Burke. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact*, 12(4):331–370, 2002.

[Chen and Kotz, 2000] Guanling Chen and David Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dartmouth College, 2000.

[Cheverst *et al.*, 2000] Keith Cheverst, Nigel Davies, Keith Mitchell, and Adrian Friday. Experiences of developing and deploying a context-aware tourist guide: The GUIDE project. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM-00)*, pages 20–31, N. Y., August 6–11 2000. ACM Press.

[Claypool *et al.*, 1999] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper, 1999.

[Cohen and Kjeldsen, 1987] Paul R. Cohen and Rick Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Inf. Process. Manage*, 23(4):255–268, 1987.

[Collins and Loftus, 1975] Allan M. Collins and Elizabeth F. Loftus. A spreading-activation theory of semantic processing. *Psychological Review*, 82:407–425, 1975.

[Crestani, 1997] Fabio Crestani. Application of spreading activation techniques in information retrieval. *Artif. Intell. Rev*, 11(6):453–482, 1997.

[Dey *et al.*, 2004] Anind K. Dey, Raffay Hamid, Chris Beckmann, Ian Li, and Daniel Hsu. a CAPpella: programming by demonstration of context-aware applications. In Elizabeth Dykstra-Erickson and Manfred Tscheligi, editors, *Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, April 24 - 29, 2004*, pages 33–40. ACM, 2004.

[Dey, 2001] A. K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.

[Guarino and Giaretta, 1995] N. Guarino and P. Giaretta. Ontologies and knowledge bases: Towards a terminological clarification. In N. J. I. Mars, editor, *Towards Very Large Knowledge Bases*, pages 25–32. IOS Press, Amsterdam, 1995.

[Hebb, 1949] Donald O. Hebb. *The Organization of Behavior*. John Wiley Sons, 1949.

[Henricksen and Indulska, 2005] Karen Henricksen and Jadwiga Indulska. Personalising context-aware applications. In Robert Meersman et al., editor, *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, Agia Napa, Cyprus*. Springer, 2005.

[Herlocker and Konstan, 2001] Jonathan L. Herlocker and Joseph A. Konstan. Content-independent task-focused recommendation. *IEEE Internet Computing*, 5(6):40–47, 2001.

[Huang *et al.*, 2004] Zan Huang, Hsinchun Chen, and Daniel Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems*, 22(1):116–142, 2004.

[Kaltz, 2006] Joachim Wolfgang Kaltz. *An Engineering Method for Adaptive, Context-aware Web Applications*. PhD thesis, Universitaet Duisburg-Essen, Campus Duisburg, 2006.

[Kappel *et al.*, 2003] Gerti Kappel, Birgit Pröll, Werner Retschitzegger, and Wieland Schwinger. Customisation for ubiquitous web applications a comparison of approaches. *Int. J. Web Eng. Technol*, 1(1):79–111, 2003.

[Kovacs and Ueno, 2006] Alexander I. Kovacs and Haruki Ueno. Recommending in context: A spreading activation model that is independent of the type of recommender system and its contents. In Vicent Wade, Helen Ashman, and Barry Smyth, editors, *Proceedings of the AH2006*. Springer, 2006.

[Melville *et al.*, 2002] Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[Middleton *et al.*, 2004] Stuart Middleton, Nigel Shadbolt, and David De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.

[Neches *et al.*, 1991] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3):16–36, 1991.

[Oppermann, 2005] Reinhard Oppermann. From user-adaptive to context-adaptive information systems. *iCom, Zeitschrift für interaktive und kooperative Medien*, 3/2005:4–14, 2005.

[Perkowitz and Etzioni, 2000] Mike Perkowitz and Oren Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artifical Intelligence*, 118(1-2):245–275, 2000.

[Pirolli and Card, 1995] Peter Pirolli and Stuart K. Card. Information foraging in information access environments. In *CHI*, pages 51–58, 1995.

[Rocha *et al.*, 2004] Cristiano Rocha, Daniel Schwabe, and Marcus Poggi de Aragão. A hybrid approach for searching in the semantic web. In *WWW*, pages 374–383, 2004.

[Ziegler *et al.*, 2005] Jürgen Ziegler, Steffen Lohmann, and Joachim Wolfgang Kaltz. Kontextmodellierung für adaptive webbasierte systeme. In C. Stary, editor, *Mensch & Computer 2005: Kunst und Wissenschaft*. Oldenbourg Verlag, München, 2005.

Mediating expert knowledge and visitor interest in art work recommendation

Leendert van Maanen
Department of Artificial Intelligence, University of Groningen
Grote Kruisstraat 2/1, 9712 TS Groningen, the Netherlands
leendert@ai.rug.nl

Abstract

In this paper, we will present an outline for an online recommender system for art works. The system, termed Virtual Museum Guide, will take the interest that visitors of an online museum express into account in recommending suitable art works, as well as the relationships that exist between art works in the collection. To keep the Virtual Museum Guide similar to a human museum guide, we based its design on principles from research on human memory. This way, the Virtual Museum Guide can 'remember' which is the most suitable art work to present, based on its perception of the visitor's interests and its knowledge of the works of art.

1 Introduction

With the advent of online information presentation, cultural heritage institutions are starting to make their collections available online. Many museums already have websites displaying digital reproductions of part of their collection. Some of these online repositories are annotated, making it possible to search for specific art works: For example, the website of the Amsterdam Rijksmuseum in The Netherlands [Rijksmuseum, 2007] is driven by an ontology on art and artists.

With the online presentation of cultural heritage content, new issues arise. While one of the advantages of digitalization and online presentation is the greater accessibility of cultural heritage [e.g., because of better search capabilities, Van Ossenbruggen *et al.*, 2007], one of the drawbacks is that there is less control over what is presented to an individual visitor. Cultural heritage institutions have as one of their aims to educate people on history and culture, which becomes harder to realize once the contents of their collection is accessible from anywhere; They can no longer cater the individual interests of museum visitors while maintaining coherence in the presented information. Besides the decreased control that cultural heritage institutions experience, finding interesting art works in an online museum poses a problem. Just like in a real museum, most online museum visitors are not aware of their specific interests or of the exact contents of the museum's collection [Bell, 2002]. Instead, they only have a general impression of what they want to see and what is available. This makes it difficult to adjust the presentation of the art works to the visitors' personal interests.

Consider the example of a professional, educated museum guide, touring a party of interested visitors through a museum. The guide can (and has to) select information on the art works from her extensive knowledge that relates to the personal interests of the party, and can choose which art work to present next from the collection on display. To reproduce a similar personal experience in an online setting, personal interests as well as relationships between art works have to be known. A successful recommender system for the cultural heritage domain should incorporate both issues mentioned above: On the one hand, it should take care of the educational role of a cultural heritage institution, and on the other hand it should provide an enjoyable and personalized experience.

1.1 Overview

In this paper, we will present an online recommender system that presents art works from the Amsterdam Rijksmuseum collection. In our approach we will try to model the way a human museum guide will behave while touring a visitor through a museum. The assumption is that if the recommender system mimics the behavior of the museum guide, we will have a successful recommender system. In order to achieve this, we will ground the structure of the recommender system in cognitive theories on how human declarative memory works [Anderson *et al.*, 2004].

To stress the analogy with a museum guide touring a group of visitors through a museum, we termed the system the Virtual Museum Guide (VMG). The VMG combines the relationships that art works have to each other with the personal interests of the visitor to arrive a suitable art recommendations. We will first give an overview of the most important aspects of the system, and then discuss each aspect in more detail.

In the system we will present here, the art works presented online are accompanied by sets of key words that indicate what are the interesting aspects of the art work. As these key words are provided by the museum's art experts, expert knowledge on the art works and their interrelations are contained therein. We have applied statistical inference tools from natural language research [Landauer *et al.*, 1998] to infer how the art works relate to each other (details will be provided in the implementation section below). This way, all art works are related to each other with an association value indicating the relevance of one art work for another. This structure can be thought of as a *semantic* or *spreading activation network* [Collins and Loftus, 1975; Quillian, 1968].

Based on the visitor's feedback on presented art works, the guide generates hypotheses on the visitor's interest. For the system presented here we opted for the use of an explicit interest indicator using an *Interesting* and a *Not*

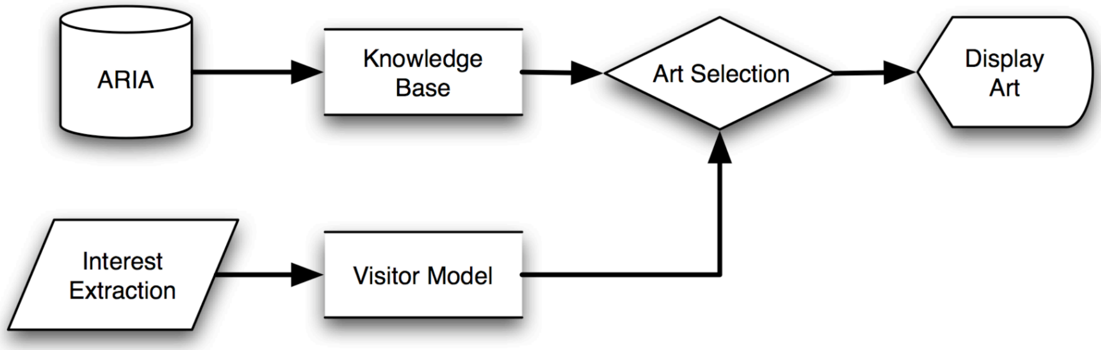


Figure 1. A flowchart of the Virtual Museum Guide.

interesting button. The interest hypotheses are represented as declarative facts that are stored in the VMG's memory.

Each time a user indicates interest in an art work by clicking one of the two interest-buttons, a new art work will be selected by computing the most relevant and interesting art work *given the current context*. First, the visitor's interest in the already visited art works will be assessed. Using a spreading activation algorithm (described in more detail below) a combined measure of interest and relevance will be computed.

2 ACT-R

The Virtual Museum Guide's memory is based on a formal theory of human cognition called ACT-R [Anderson *et al.*, 2004]. A major part of ACT-R is its model of declarative memory functioning [Anderson and Milson, 1989; Anderson and Schooler, 1991], and we will apply this approach in the context of a recommender system.

The key insight here is that human memory is optimally adapted to deal with information that has been presented in the past [Anderson and Milson, 1989; Anderson and Schooler, 1991]. Following this line of reasoning, the way information is represented in memory may also be optimal for storing a model of a person's interactions with information presented in the past.

Anderson and Schooler [1991] demonstrated that for each declarative fact stored in memory, the probability that that piece of information will be needed in the immediate future reflects the history of usage of that piece. That is, information that has been presented recently is more likely to be needed again than items that have been presented in the more distant past. Also, information that has been presented more frequently is more likely to be needed again. In ACT-R, the probability that information will be needed in the immediate future is represented by a quantity called *activation*. The declarative memory representation consists of small pieces of declarative knowledge, called chunks, that together represent a person's long-term memory. Each chunk has an activation value and associations with other chunks.

The two environmental observations (recency and frequency) have crystallized [Anderson *et al.*, 2004] into the following activation equation:

$$B_i = \ln\left[\sum_{j=1}^{n} \frac{1}{\sqrt{t_j}}\right] \qquad \text{(Equation 1)}$$

B_i represents the base-level activation of a chunk (indicated by the index i). The equation captures the effect of frequency of presentation by summing over multiple presentations, and the effect of recency of presentation by dividing by the square root of each presentation time lag (represented by t_j), that is, the time since the presentation of the chunk. This equation has been used in numerous studies predicting memory retrieval effects, both for theoretical purposes [e.g., Anderson *et al.*, 1998; Van Maanen and Van Rijn, *in press*] and for application-based research [e.g., Pirolli, 2005; Van Maanen *et al.*, 2006].

Besides the frequency and the recency with which memory facts are encountered, also the contexts in which they are encountered adds to their activation. This is determined by the likelihood that two facts have co-occurred in the past [Anderson and Milson, 1989]. The likelihood that one fact needs to be retrieved from memory is predicted by the recent retrieval from memory of another fact, and this prediction is based on how often it has been accurate in the past.

3 Virtual Museum Guide

Figure 1 presents a flowchart of the Virtual Museum Guide. We start out with extraction of a Resource Description Framework (RDF) specification of each art work from the online ARIA (Amsterdam Rijksmuseum Inter-Actief) repository, which can be inspected at http://media.cwi.nl/sesame/[1]. The RDF specification is transformed to an associative network structure, called the Knowledge Base. The Knowledge Base contains the knowledge the VMG has on the art works and their interrelations. Besides the knowledge on the art works, the VMG forms hypotheses on the interests of the visitors. These are extracted from the visitors' behavior and stored in a Visitor Model. Based on both knowledge sources, the VMG selects a suitable art work and displays it for the visitor, together with a little background information on the art work.

3.1 Knowledge Base

A human museum guide might present two similar art works right after each other, for instance because they are painted by the same artist. Therefore, the similarity be-

[1] To inspect the RDF repository, select *Topia's RDF Aria for Sesame* in the drop-down menu and slect one of the read actions. More information on how to query this repository can be found on openRDF.org [openRDF, 2007].

tween two art works might also represent the likelihood of two art works co-occurring. The similarity between art works will be represented by an associative network structure [Collins and Loftus, 1975; Quillian, 1968], in which the association values indicate similarity: The stronger the association between two art works, the more similar they are considered to be. The associative values in the associative network are based on the frequency statistics of the key words that occur in the RDF specifications of the art works. The general idea is that two sets of key words that greatly overlap might be considered similar to one another. This can be thought of as two feature-vectors that lie close together in a highly dimensional space. We applied Latent Semantic Analysis (LSA) [Deerwester *et al.*, 1990; Landauer and Dumais, 1997; Landauer *et al.*, 1998] on the scaled frequency vectors representing the art works. For this we used the standard TF-IDF weighting scheme [Salton and McGill, 1983], which scales the frequency of terms in a document by the number of documents in which the terms occur. In the VMG's knowledge base, this means that the frequency of the key words in all RDF specifications is taken into account.

At this point it is important to note that LSA is more than just a correlation of frequency counts [Deerwester *et al.*, 1990; Landauer and Dumais, 1997; Landauer *et al.*, 1998]. Instead, LSA depends on a mathematical analysis (singular value decomposition) that is capable of a higher-order inference. For example, let's assume that the specification of Rembrandt's *The Night Watch* contains the key word *claire-obscure,* and Gerard van Honthorst's *The Merry Fiddler* contains the key word *caravaggists*. LSA is capable of inferring that these two art works share a feature (namely, the technique used) if these key words co-occur in similar contexts. For example, a specification of Dirck van Baburen's *Prometheus Being Chained by Vulcan* might mention the key words *caravaggists, light, dark,* and *contrast*, and Rembrandt's *Ecce Homo* might contain the key words *light, dark, contrast*, and *claire-obscure*. In a sense, LSA estimates the likelihood that the word *claire-obscure* would occur in the specification of *The Merry Fiddler*, and the likelihood that *caravaggists* would occur in the specification of *The Night Watch*. For a more detailed, but still non-technical introduction to Latent Semantic Analysis, the reader is referred to Landauer, Foltz, and Laham [1998].

After constructing the semantic space using LSA, the similarity between two art works is computed by calculating the cosine between their representing vectors [Salton *et al.*, 1975]. The cosine between two feature-vectors represents their angle. This indicates how much they deviate in the semantic space, with small angles representing greater similarity than large angles.

3.2 Visitor Model

In the current setup of our virtual museum, visitors encounter each art work only once. Therefore, we simplified Equation 1 to

$$B_i = \ln\left[\frac{1}{\sqrt{t_i}}\right] \qquad \text{(Equation 2)}$$

in which t_i is the time stamp of the presentation of an art work (i). Note that there is no reason why this simplification will remain in future versions of the VMG, since both Equation 1 and Equation 2 provide an activation value. Figure 2 presents an example of the dynamics of two interest hypotheses in the current setup on the VMG.

For the purpose of designing a virtual museum guide we are not really interested in the probability that a piece of information will be needed, but rather in the likelihood that user interest in an art work extends to interest in similar art works. Thus, in this case, the base level activation represents the activation of the interest hypotheses that the VMG has about the visitor's interest in the already presented art works. Note that the activation of the interest hypotheses decays over time, indicating that the VMG forgets interests that have been expressed in the past. Although this seems as if the VMG incorporates a negative value of the human museum guide (forgetting), we believe that this is actually a positive effect. Museum visitors might switch the main topic of their interests, either because their knowledge on the art works has increased, or simply because they want to be presented with something different. The VMG can cope with this by forgetting hypotheses formed in the past.

A virtual museum visitor can express her interest in the art work that is currently being presented by clicking the *interest* button on the interface. Also, she can express disinterest by clicking a *disinterest* button. A positive interest expression will increase the likelihood that similar art is of interest for this visitor, whereas a negative interest expression will decrease the likelihood that similar art works are of interest. Therefore, the activation value of an art work is determined by positive spreading activation from art works in which interest is expressed, and negative spreading activation from art works in which disinterest is expressed.

3.3 Art Selection

The selection of art works depends on a weighted scheme of visitor interest. For each art work, the spreading activation from already visited art is computed. Art that is rated as uninteresting spreads negative activation; art that is rated as interesting spreads positive activation. Because of

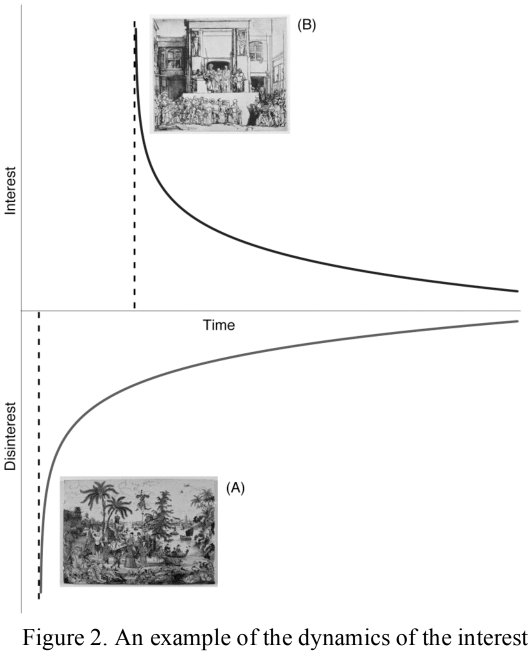


Figure 2. An example of the dynamics of the interest hypotheses. The visitor has expressed disinterest in art work A and interest in art work B. The activation of the hypothesis facts decays over time, decreasing the influence the hypotheses have on the art selection mechanism.

the inclusion of the recency component in the activation equation discussed above, the influence of recently presented art work is higher than the influence of art work presented longer ago. The spreading activation is scaled according to the similarity between art works. Thus, art works that are highly similar spread relatively more activation towards each other. These considerations result in the following equation (Equation 3), in which A_i represents the activation of a certain art work *i*, B_j represents the hypothesis on visitor interest in already presented art works (*j*), and S_{ji} represents the similarity between art works *i* and *j*.

$$A_i = \sum_j B_j S_{ji} \qquad \text{(Equation 3)}$$

This equation represents how suitable an art work will be to present to the current visitor given what the system knows from the visitor's interest, and the relations that exist between art works. Since B_j can be either a positive value or a negative value (depending on the VMG's hypothesis on the visitor's (dis)interests), art works that were considered uninteresting decrease the activation of related art works, while art works that were considered interesting increase the activation of related work. Thus, the resulting activation of an art work will be high if a visitor expressed interest in related art work, and did not expressed disinterest in related art work. Similarly, the activation will be low (that is, negative), if a visitor only expressed disinterest in related art work. The art work with the highest activation will be selected next for presentation.

After an art work has been selected for presentation, a web page will be generated that contains a digital reproduction of the art work under consideration and some information on the art work. These snippets of information are taken from the Rijksmuseum database, so it is ensured that the information is correct and relevant to the art work.

4 Discussion and Conclusion

This section will wrap up the paper by contrasting our approach to already existing tools for personalized information presentation in the museum domain. Also, we will indicate the future directions of our work.

4.1 Related Work

The key features of the VMG are the combination of the spreading activation network structure of the knowledge base combined with the decaying level of visitor interest. Also, the generation of the knowledge base using Latent Semantic Analysis is an important aspect, as well as the dynamic generation of web content.

Although most of these features have been applied in previous information presentation tools for the museum domain, the combination we apply is, to our knowledge, unique. Also, most other applications focus on the presentation aspects of dynamically generated content, especially in the context of a real, non-virtual, museum, where the mobility of the visitors poses specific challenges for the presentation of information [e.g., Hatala and Wakkary, 2005; Stock *et al.*, 2007; for a review see Raptis *et al.*, 2005]. A third obvious difference between related work and our approach is that while most applications focus on the personalized presentation of *background information* with an artefact, personalization in the VMG involves the selection of the museum artefacts themselves. In this section, we will discuss two systems that seem to be most similar to ours in the key features we have identified for the VMG. That is, both systems – ec(h)o [Hatala and Wakkary, 2005] and PEACH [Stock *et al.*, 2007] – are constructed around a conceptual network, in which selection of concepts is mediated by expressed visitor interests.

Similar to VMG, PEACH [Stock *et al.*, 2007] also adopts an activation based network. Since PEACH's main output modality is video, the nodes in the network represent video segments, and the edges represent semantic relations between these video segments. Interest expressed in one video segment propagates as activation through the network to all related other segments, and new information will be presented based on the activation values of all video segments. This seems to be a similar approach as the VMG deploys, although the level of semantic relatedness is less fine-grained, due to the Latent Semantic Analysis performed on the edges of the VMG associative network.

PEACH also differs from the VMG in the temporal aspects of the relevance feedback. Visitor's expressed interest in a video segment in PEACH does not extend to another artefact, but only applies to the current art work. Therefore, decay of visitor interest values is unneccesary. Since the VMG is intended for the dynamic selection of art works, visitor interest must extend to other art works.

Just like the VMG, ec(h)o [Hatala and Wakkary, 2005] uses a conceptual ontology as a knowledge base. In ec(h)o, the ontology is based on the Conceptual Reference Model [Crofts *et al.*, 2003] which is specifically developed for cultural heritage concepts. Selection of information is subsequently established by reasoning over the relationships in the ontology. ec(h)o also has a decay mechanism to ensure that more recent interests are more important than older ones. The mechanism implemented in ec(h)o is however not time-based (as is the decay mechanism of the VMG), but rather the interest values of concepts are normalized such that the highest value stays under a certain upper bound. An advantage of that approach could be that a longer visit to an art work does not result in 'forgetting' of interests, which is a side-effect of the way interest decay is modeled in the VMG.

The ec(h)o system differs from the VMG and PEACH in the way relevance feedback can be expressed. Were VMG and PEACH adopt an explicit strategy in which interest as well as disinterest can be expressed, ec(h)o presents the user with three small audio snippets, from which the visitor can choose. The assumption is that the visitor chooses the audio fragment that is the most interesting to her. As a result of this design choice, visitors cannot express disinterest. Moreover, they have to base their decision on a small snippet of the actual information, and could well change their minds after they hear all the information. In this sense, ec(h)o does not really incorporate a relevance feedback mechanism.

4.2 Conclusion and Future Work

An analysis of the selection of art works provided to the author suggests that the VMG is capable of recommending art works that relate to those recently indicated as interesting. Also, the VMG's knowledge base correctly relates items that seem similar upon visual inspection of the art work and the description. However, since the author is no art specialist, we are planning two evaluation studies. First, we are planning a study in which art experts can assess the relationships between art works that are present in the VMG's knowledge base. Secondly, a user study to measure how visitors of the online museum respond to personalized recommendations of art works is currently conducted. This user study can be visited at the website of our institute: www.ai.rug.nl/cogmod [AI, 2007].

In the current setup, we opted to generate interest hypothesis by having visitors press one of two interest-buttons. However, the way visitor feedback is provided can be very diverse, ranging from the simple button press to more unobtrusive methods, including the time spent observing the art work [e.g., Claypool *et al.*, 2001] or even eye gaze analysis [e.g., Van Maanen *et al., submitted*]. In a future version of the Virtual Museum Guide, we plan to incorporate less obtrusive methods to infer visitor interest. This will also include a more gradient sense of the likelihood that a visitor is interested. We will implement this by adding a parameter to Equation 2 that can control the impact of each presentation of information.

Using principles from cognitive science, we were able to implement a working system that recommends art works, based on both visitor interests and expert knowledge on the relations between the art works. In the context of a museum, both aspects are important. Because of the educational role of museums, recommending art work is more than mapping visitor interest on the museum's collection. The museum needs to ensure that the resulting sequence of art works is coherent and transfers (part of) the museum's message. It seems that the Virtual Museum Guide ensures both aspects in art work recommendation.

Acknowledgements

The art works in Figure 2 are from the collection Rijksmuseum Amsterdam. This research is supported by the Netherlands Organisation for Scientific Resarch (ToKeN/I^{2}RP project grant no. 634.000.002). Hedderik van Rijn, Chris Janssen, and three anonymous reviewers are acknowledged for valuable comments on earlier versions of this paper.

References

[AI, 2007] Cognitive Modeling, University of Groningen (www.ai.rug.nl/cogmod). Retrieved 4-7-2007.

[Anderson *et al.*, 2004] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060, 2004.

[Anderson *et al.*, 1998] Anderson, J. R., Bothell, D., Lebiere, C., and Matessa, M. An integrated theory of list memory. *Journal of Memory and Language, 38*(4), 341-380, 1998.

[Anderson and Milson, 1989] Anderson, J. R., and Milson, R. Human memory: An adaptive perspective. *Psychological Review, 96*(4), 703-719, 1989.

[Anderson and Schooler, 1991] Anderson, J. R., and Schooler, L. J. Reflections of the environment in memory. *Psychological Science, 2*(6), 396-408, 1991.

[Bell, 2002] Bell, G. *Making sense of museums: The museum as 'cultural ecology'*: Intel Labs, 2002.

[Claypool *et al.*, 2001] Claypool, M., Brown, D., Le, P., and Waseda, M. Inferring user interest. *IEEE Internet Computing, 5*(6), 32-39, 2001.

[Collins and Loftus, 1975] Collins, A. M., and Loftus, E. F. A spreading activation theory of semantic processing. *Psychological Review, 82*(6), 407-428, 1975.

[Crofts *et al.*, 2007] Crofts, R., Doerr, M, and Gill. T. The CIDOC conceptual reference model: A standard for communicating cultural contents. *Cultivate Interactive, 9*, 2003.

[Deerwester *et al.*, 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391-407, 1990.

[Hatala and Wakkary, 2005] Hatala, M., and Wakkary, R. Ontology-based user modeling in an augmented audio reality system for museums. *User Modeling and User-Adapted Interaction, 15*(3-4), 339-380, 2005.

[Landauer and Dumais, 1997] Landauer, T. K., and Dumais, S. T. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*(2), 211-240, 1997.

[Landauer *et al.*, 1998] Landauer, T. K., Foltz, P. W., and Laham, D. An introduction to latent semantic analysis. *Discourse Processes, 25*(2-3), 259-284, 1998.

[openRDF, 2007] openRDF.org (www.openrdf.org). Retrieved 29-8-2007.

[Pirolli, 2005] Pirolli, P. Rational analyses of information foraging on the web. *Cognitive Science, 29*(3), 343-373, 2005.

[Quillian, 1968] Quillian, M. R. Semantic memory. In M. Minsky (Ed.), *Semantic information processing* (pp. 216-270). MIT Press, Cambridge, MA, 1968.

[Raptis *et al.*, 2005] Raptis, D., Tselios, N., and Avouris, N. Context-based design of mobile applications for museums: A survey of existing practices, *Mobile-HCI'05* (pp. 153-160). Salzburg, Austria: ACM.

[Rijksmuseum, 2007] Rijksmuseum Amsterdam, national museum for art and history (www.rijksmuseum.nl). Retrieved 4-7-2007.

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

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

[Sesame, 2007] Sesame@media.cwi.nl, public repositories (http://media.cwi.nl/sesame/). Retrieved 21-6-2007.

[Stock *et al.*, 2007] Stock, O., Zancanaro, M., Busetta, P., Callaway, C., Kruger, A., Kruppa, M., et al. Adaptive, intelligent presentation of information for the museum visitor in peach. *User Modeling and User-Adapted Interaction, 17*(3), 257-304, 2007.

[Van Maanen *et al.*, 2006] Van Maanen, L., Borst, J. P., Janssen, C. P., and Van Rijn, H. Memory structures as user models, *13th Annual ACT-R Workshop*. Pittsburgh, PA, 2006.

[Van Maanen and Van Rijn, *in press*] Van Maanen, L., and Van Rijn, H. An accumulator model of semantic interference. *Cognitive Systems Research,* in press.

[Van Maanen *et al.*, *submitted*] Van Maanen, L., Van Rijn, H., and Janssen, C. P. Eye-gaze based interest awareness for adaptive multimedia art presentations, submitted.

[Van Ossenbruggen *et al.*, 2007] Van Ossenbruggen, J., Amin A., Hardman L., Hildebrand M., Van Assem M., Omelayenko B., Schreiber G., Tordai A., De Boer, V., Wielinga B., Wielemaker J., De Niet, M., Taekema J., Van Orsouw, M.-F., and Teesing A.. Searching and Annotating Virtual Heritage Collections with Semantic-Web Techniques. In *Proceedings of Museums and the Web 2007*, San Francisco, CA, March, 2007.

Towards Learning User-Adaptive State Models in a Conversational Recommender System

Tariq Mahmood
University of Trento
Trento, Italy
tariq@itc.it

Francesco Ricci
Free University of Bozen-Bolzano
Bolzano, Italy
fricci@unibz.it

Abstract

Typical conversational recommender systems support interactive strategies that are hard-coded in advance and followed *rigidly* during a recommendation session. In fact, Reinforcement Learning techniques can be used in order to *autonomously learn an optimal (user-adaptive) strategy*, basically by exploiting some information encoded as features of a state representation. In this regard, it is important to determine the set of *relevant* state features for a given recommendation task. In this paper, we address the issue of feature relevancy, and determine the relevancy of adding four different features to a baseline representation. We show that adding a feature might not always be beneficial, and that the relevancy could be influenced by the user behavior. The results motivate the application of our approach online, in order to acquire the right mixture of online user behavior for addressing the relevancy problem.

1 Background and Motivation

Recommender systems [Resnick and Varian, 1997] are online applications that assist users in their information-seeking tasks by offering personalized product recommendations during an interaction session. In order to support more interactive sessions, *conversational recommender systems* [Ricci *et al.*, 2003; Thompson *et al.*, 2004; Shimazu, 2001] have been recently proposed. These systems adopt the conversation style of real-life dialogues in order to guide users through complex product spaces. Mainly, at each dialogue stage, the system selects one from amongst a set of available system actions, e.g., recommend some product, ask the user for some information etc. The particular action selection is specified by the system's *recommendation strategy*. For instance, suppose that a conversational recommender is queried for hotels that would suit the user preferences. Then, it could employ the following two strategies (among many others): 1) initially query the user in detail for her preferences, in order to extract a small product subset, or 2) recommend a set of products, and exploit the user feedback to refine future recommendations. Conversational systems typically employ a *rigid* strategy, i.e., one which is hard-coded inside the system and followed rigidly during a session [Ricci *et al.*, 2003; Thompson *et al.*, 2004; Shimazu, 2001]. A limitation of this approach is that there could be numerous rigid strategies for a given recommendation task. Furthermore, as system designers rely on intuition and previous experience in order to select a strategy, it is is possible that they shall have to evaluate several strategies before they discover a (almost) "good" one. This is an infeasible process in terms of budget, time, task force etc. These limitations would be tamed if conversational systems could be capable of determining *by themselves* the *best* (optimal) strategy for assisting the users.

In a previous paper [Ricci and Mahmood, 2007], we have tackled these requirements by proposing a new type of recommender system which is able to *autonomously improve an initial strategy in order to learn and adopt an optimal one*. We validate our approach by applying it within the NutKing travel recommender system [Ricci and Mahmood, 2007], and by improving NutKing's (current) rigid strategy in order to learn an optimal one. We learn the strategy through the Reinforcement Learning (RL) paradigm [Sutton and Barto, 1998] in the context of the Markov Decision Process (MDP) framework (we avoid a lengthy explanation due to space constraints). Briefly, our system, at each stage of the session, observes the current *state* (or situation) of the user and the session activity. In one or more states (labeled as *System Decision Points* (SDPs)), multiple actions could be available for execution. The system executes these different actions as the interaction proceeds, and in return, receives a *numerical reward* informing it of the action's acceptability for the user. As the interaction continues, the system learns to *maximize the reward it receives in each possible state*, i.e., it learns to avoid the unacceptable actions and to take the acceptable ones, until it learns the optimal action in each state, i.e., the *optimal policy* (strategy). Also, in the *non-SDP* states, only a single action is available which we consider as the optimal action. The optimal behavior is hence characterized by information in the observed state, which is represented as a combination of feature values. Considering that online user behavior is generally detailed and complex, the selection of the *relevant* state features for a given recommendation task becomes quite crucial.

In this paper, we address several issues related to the feature relevancy problem. We introduce two relevancy criteria, and use them to determine the relevancy for four different feature sets (or *state representations*). We show that adding a feature might not always be relevant for a given task. We also investigate relevancy across different (simulated) user behaviors models, and show that relevancy is influenced by the user behavior. These results motivate the application of our approach in an online context, where we will be able to acquire behavior for a real user population, and to learn an optimal policy for this population.

2 Evaluation Setup

We will now present the evaluation setup, i.e., our proposed sets of state representations and user behavior models, in order to address the feature relevancy problem. For our evaluation, we will use the scenario presented in [Ricci and Mahmood, 2007]. Specifically, we simulate a user (details provided later on) who formulates a logical query to a product catalogue of NutKing, by constraining zero or more product features. If the query retrieves a large result set, i.e., greater than a certain threshold (set to 50 in this evaluation), the system's rigid policy is to suggest a set of features to the user for *tightening* (or reducing the result size) of the current query (Action: *Suggest*). Otherwise, the system executes the query and shows all the results to the user (Action: *Execute*). The system's action set is shown in Table 1. Although NutKing supports other actions as well, we show only these two in Table 1 because, in order to learn an optimal policy, the system must decide to choose only between these two actions, i.e., these are the actions available in the SDP states. In [Ricci and Mahmood, 2007], we have improved NutKing's rigid policy in order to learn an optimal one. Furthermore, our measure of assigning numerical rewards to the system is shown in Table 2: in our scenario, the user's main goal is to add a product to her cart. Hence, we assign a large positive reward (+1) in this situation. Also, until some product is selected, the system is *punished* with a small reward at each stage, in order to convey that the user's goal has (as yet) not been achieved.

Action	Description
Suggest	Suggest Tightening Features
Execute	Execute Query and Show Results

Table 1: System Action Set

$Value$	$Situation$
$+1$	user adds a product to her cart
-0.05	an interaction session stage elapses

Table 2: Reward Function

2.1 State Representations

Given a state representation, the system's job is to learn the optimal policy, i.e., in each SDP, decide whether to select *Suggest* or *Execute*. For our current study, we will select a baseline state representation (*Baseline*). Then we will add four different features separately to this baseline, and determine the relevancy of adding each new feature. Our selected state feature set (and the corresponding value discretizations) is shown in Table 3, and the set of representations is shown in Table 4. We now present our rationale for selecting these state features. The feature *PUA* depicts the combinations of the web pages (of NutKing) and the possible user actions in these pages [Ricci and Mahmood, 2007] (here, *S-go* represents the situation where the user has logged on to the system). We will describe the remaining combinations later on. Also, *CRS* depicts the result set size when the user executes a query. We select *PUA* and *CRS* in *Baseline* as they provide the minimum information required by the system in order to decide between *Suggest* and *Execute*. *The evaluation task is to add the remaining features, one-by-one, to Baseline, and to determine (using one or more relevancy criteria) whether adding this feature was really beneficial for the system or not.* In this context, we say that the task is to *determine the relevancy of each representation in our representation set R*, where *R*={*Rep1, Rep2, Rep3, Rep4*}.

The feature *ERS* is the system's estimation (based on summary information about data distribution in the catalogue) of the result set size if the best available feature for tightening (the features are ranked according to a feature selection method) is used for tightening by the user. In fact, in [Ricci and Mahmood, 2007] we used representation *Rep1*, but we did not investigate the relevancy of adding *ERS* (as we shall do now). Moreover, in a previous online evaluation, we found that users were not too willing to respond to the action *Suggest*, i.e., they constrained a suggested feature only 26% of the time [Ricci *et al.*, 2003]. Thus, it is probably necessary for the system to know the frequency of times *Suggest* has been executed (feature *FTSugg*) and the general response of the user to *Suggest* (feature *UserTResp*), where *accept* implies that the user has always accepted a suggestion whenever *Suggest* has been executed, *reject* means that the user has never accepted any constraint suggestion, and *mixed* implies that the user has accepted tightening only a certain percentage of the time. Besides this, we believe that the number of session stages which have elapsed up to some stage (feature *NStages*) might affect how the user responds to *Suggest*, e.g., users might accept suggestions earlier on but might get frustrated as the session prolongs.

2.2 Evaluation Procedure

In order to address the feature relevancy problem, we conducted some off-line simulations, in which we initially defined a simulated user behavior model. This model specifies how the user responds to (all the possible) system actions during a session. Then, we simulated a sequence of user-system sessions or *trials*. Each trial is composed of some stages, and simulates the user as incrementally modifying a query to finally select/add a product to her cart. We ran this procedure for a set of randomly selected products from the catalogue. In this simulation we performed a leave-one-in selection, i.e., for each trial we selected a product *t*, in which values are specified for product features, i.e., $t = (v_1, ..., v_n)$. We call *t* as the *test item*, and we simulated a user searching for this item. The values used by the user to modify her query, e.g., when a suggested feature is accepted are those of the test item. Note that not all the features in *t* have a specified value, i.e., some of these v_i may be *Null*.

In our evaluation, we determined the relevancy for the set

State Feature	Discretized Feature Values
Page-UserAction (*PUA*)	{*S-go, QF-execq, T-acct, T-rejt, T-modq* *R-modq, R-add, G*}
Current Result Size (*CRS*)	{*small (0 - 20) , medium (20 - 50)* *large (50 - 100) , verylarge (100 - INF)*}
Expected Result Size (*ERS*)	{*small (0 - 20) , medium (20 - 50)* *large (50 - 100) , verylarge (100 - INF)*}
Freq Tight Sugg (*FTSugg*)	{*small (0 - 2) , medium (2 - 4) ,* *large (4 - INF)*}
Number Int Stages (*NStages*)	{*small (0 - 3), medium (3 - 6) ,* *large (6 - INF)*}
User Tighten Resp (*UserTResp*)	{*accept, mixed, reject*}

Table 3: State Features (*INF*=Infinity)

Rep	State Feature Set
Baseline	$\{$*PUA, CRS*$\}$
Rep1	$\{$*PUA, CRS, ERS*$\}$
Rep2	$\{$*PUA, CRS, FTSugg*$\}$
Rep3	$\{$*PUA, CRS, NStages*$\}$
Rep4	$\{$*PUA, CRS, UserTResp*$\}$

Table 4: Different State Representations (*Rep*=Representation)

R with a set of five different user behavior models. Specifically, for *each* user model, we determined the relevancy for each representation in *R*. The rationale is to *investigate whether a representation, which is relevant for a particular group of users, is also relevant for some other group(s)*. The result is important because online user behavior is typically quite diverse. So, in order to select a relevant representation for a given recommendation scenario, we must determine how the relevancy is actually influenced by these different behaviors.

2.3 User Models

In this section, we will describe the different user models used in our evaluation. These models differ in how the simulated user responds to the action *Suggest*. Specifically, in our scenario, each user model specifies how the simulated user behaves during the session in the following three cases:

- (Case 1) When *PUA=QF-execq*, i.e., when the user formulates and executes a query, how does she actually *constrain* the features in (or modify) her current query?
- (Case 2) When the system executes a query (*Execute*), whether the user will add the test item to the cart (*PUA=R-add*) or modify the query (*PUA=T-modq*).
- (Case 3) When the system suggests tightening (*Suggest*), whether the user will decide to modify her current query (*PUA=T-modq*), or to accept tightening (*PUA=T-acct*), or to reject tightening (*PUA=T-rejt*).

Let us now describe these three situations. (Case 1): At the beginning of each session, we sort the features of the test product according to their frequency of usage (as observed in real-user interactions with NutKing [Ricci *et al.*, 2003]). Then we use this sorting to choose the first feature to constrain in the initial query, and also to incrementally select the next feature to constrain when the user system again observes the state with *PUA=QF-execq*. In Case 2, the user will add the test item to the cart (*PUA=R-add*) if the test item is found in the top *N* items returned by the query (we set N=3). Otherwise the user will modify her current query (*PUA=R-modq*).

For Case 3, we model five user behavior models (which differ only in their response to *Suggest*): 1) a generic user model (*GUM*) which was used in [Ricci and Mahmood, 2007], 2) a willing user (*WillUM*), i.e., a user who is always willing to accept tightening, 3) a moderately-willing user (*ModwillUM*), i.e., a user who is willing to accept tightening only sometimes during her session, 4) an unwilling user (*UnwillUM*), i.e., a user who is never willing to accept tightening, and 5) all the above users (*AllUM*), i.e., a model which simulates the behavior of a population of users. Let us define the set *UM* as $\{$*UM=GUM, WillUM, UnWillUM, ModwillUM, AllUM*$\}$. We now detail *UM* as follows:

1. ***GUM***: This model simulates a generic response behavior to *Suggest*: the user accepts tightening if any one of the suggested features has a *non-Null* value in the test item, and this feature is also the next preferred feature of the user (according to the feature usage order of the test item). If the user doesn't accept tightening and the result size is smaller than 50 (*CRS=small,medium*), then the user rejects it and executes the original query. In the remaining cases (i.e., when *CRS=large,very large* and when acceptance cannot be simulated, the user autonomously modifies her query (as in Case 1) (*T-modq*).
2. ***WillUM***: The user accepts tightening if any one of the suggested features has a *non-Null* value in the test item, *even if it is not her next preferred feature*. This latter condition allows us to simulate the user's "willingness" to accept tightening (as compared to *GUM* where the user accepted tightening if the test item feature was also her next preferred one). If acceptance cannot be simulated, the user rejects tightening or manually modifies her query similarly to the corresponding behavior in *GUM*.
3. ***ModwillUM***: The user considers accepting tightening only 26% of the time that *Suggest* is executed during a session. Hence, *ModwillUM* simulates the real-user response to *Suggest* [Ricci *et al.*, 2003]. If the user considers accepting tightening, acceptance is simulated similarly to the behavior in *WillUM*. If acceptance cannot be simulated in this case, or if the user does not consider accepting tightening (74% of the time), then the user either rejects tightening or manually modifies her query as in *GUM*.
4. ***UnwillUM***: The user never accepts tightening; if *CRS=small,medium*, the user rejects tightening and executes the query. Otherwise, if *CRS=large,verylarge*, the user modifies her query as in *GUM*.
5. ***AllUM***: This model simulates the behavior of a population of users: each time *Suggest* is executed, we randomly select (from a uniform distribution) and simulate one from amongst the above four user behaviors. We note that *ModwillUM* also simulates a population behavior, but *AllUM* adds more diversity. It is important to determine the relevancy of *R* under a diverse population because, if we apply our approach in an online setting, the optimal policy will be learnt for a user population showing different behaviors) rather than for users exhibiting just a single type of behavior (e.g., always rejecting tightening).

3 Relevancy Criteria

In this section, we will describe our proposed criteria for determining the relevancy of a given representation. In order to apply these criteria, we will initially learn the optimal policy (OP) for all possible "State Representation - User Model" combinations. As we have a total of five representations (Table 4) and five user models (the set *UM*), we learn a total of $5 * 5 = 25$ Optimal Policies (OPs). For all the OPs, we are interested only in the SDP states, i.e., those which have *PUA=QF-execq*, and for all the state combinations listed in the rest of the paper, we assume that *PUA=QF-execq*.

Our first relevancy criterion, *OPEval*, is based on an *evaluation* of the optimal policies, i.e., on determining the

UM	Different State Representations				
	Baseline	*Rep1*	*Rep2*	*Rep3*	*Rep4*
GUM	0.521	**0.5369**	**0.5623**	**0.5213**	**0.5491**
Will	0.5696	**0.5749**	0.5628	0.5576	0.5436
Mod	0.5326	**0.5469**	**0.5625**	**0.5633**	**0.5526**
Unwill	0.5636	0.5629	0.5491	0.5636	0.5539
All	0.5459	0.5399	**0.5626**	0.5437	0.5418

Table 5: Average cumulative rewards under different "User Model - State Representation" combinations (*UM*=user model, *Will = WillUM*, *Mod = ModwillUM*, *Unwill = UnwillUM*, *All = AllUM*)

total reward which the system can accumulate while employing a particular OP. The rationale is that if an OP for a representation $r \in R$ (called OP_r) allows the system to accumulate more reward than the reward for the OP of *Baseline* representation (OP_{base}), then r is a more relevant representation than *Baseline*. Specifically, we select a set of 300 items from the product catalogue. For each item, we simulate an interaction session and calculate the total reward which the system accumulates in that session. At the end, we compute the average cumulative reward (*AvgCumRwd*) for all the 300 sessions. When we evaluate an optimal policy OP of a representation rep and learnt under a user model $um \in UM$, we mean that during the session, the system takes actions according to OP with the user responses being generated through um. For *each user model in* UM, we evaluate OP_{Base} and the four OPs obtained for each representation in R. Then, a representation $r \in R$ is relevant if the *AvgCumRwd* obtained after evaluating OP_r is greater than the *AvgCumRwd* obtained after evaluating OP_{base}. Otherwise, we say that r is irrelevant. In RL, policy evaluation is a robust metric which is commonly used in order to determine the suitability of an OP. Hence, $OPEval$ is a robust metric for determining relevancy.

We propose another relevancy criterion (*OPComp*) which is based on a comparison of the optimal policies. Here, *for each user model in UM*, we compare OP_{base} with each of the four OPs obtained under the set *R*. Then, we say that a representation $r \in R$ is *relevant under some state s* in OP_{base} if OP_r is able to learn a *different* system action than the optimal action for *s* in OP_{base}. Otherwise, we say that *r* is *irrelevant under s*. For instance, suppose that for some user model $um \in UM$, we compare OP_{base} with the optimal policy for *Rep1* (OP_{rep1}), specifically for the state (S_{base}) in OP_{base} where *CRS*=*small*. Let us define $R1CRS_{small}$ as the set of all states in OP_{rep1} where *CRS*=*small*. Also, suppose that the optimal action for S_{base} is *Execute*, and that the optimal action for one or more states in $R1CRS_{small}$ is different than *Execute*, i.e., *Suggest*. Such a behavior implies that adding the feature *ERS* is allowing the system to learn different optimal actions and hence change its optimal behavior. Hence, we imply that *R1* is relevant under S_{base}. Also, we say that *R1* is relevant if it is relevant under one or more states in OP_{base}, and that it is irrelevant if it is not relevant under any state in OP_{base}.

4 Results

In this section, we present our results in order to determine the relevancy of the set R under our proposed relevancy criteria, *OPEval* and *OPComp*.

4.1 *OPEval*

For *OPEval*, we first evaluated the 25 optimal policies learnt for the five representations in Table 3, under each user model $um \in UM$. Table 5 shows the *AvgCumRwd* values obtained for each policy evaluation. For each user model, we compare the reward obtained for *Baseline* with the reward obtained for each representation $r \in R$. All the rewards for the set *R* which are greater than their corresponding *Baseline* reward are shown in bold. The results show that all representations in *R* are relevant under *ModwillUM*, showing that for a particular class of real users, adding *ERS*, *FTSugg*, *NStages* and *UserTResp* separately to *Baseline* leads the system to accumulate more reward. Similar results are also obtained under the model *GUM*. However, none of the representations in *R* are relevant under *UnwillUM*, i.e., for unwilling users it is best for the system to always execute the query (see OP_{base} in Table 9) without considering any further information. Also, only *Rep1* is relevant under *WillUM*, i.e., for willing users, it is best to add only *ERS* to *Baseline*. These results prove that a representation which is relevant for a given user group might not be relevant for some other group, i.e., *the relevancy is influenced by the user behavior*. In this context, it is best to determine relevancy under a behavior for a user population, i.e., under *AllUM*. In this case, only *Rep2* is relevant in *R*, i.e., it is best to add only *FTSugg* to *Baseline*. Let us analyze the OP for *Rep2* in Table 10, which allows the system to acquire more reward. This policy shows that our user population is willing to follow tightening suggestions only when a large number of products are retrieved, and then also, only for the first few times that it is suggested. If the system suggests tightening just more than 3 times, the population ignores it even if a lot of products are retrieved.

4.2 *OPComp*

Let us now determine relevancy under *OPComp*. Tables 6-10 show the optimal policies for the representations in Table 3, under the user model *GUM*, *WillUM*, *ModwillUM*, *UnwillUM* and *AllUM* respectively. In each table, the different state combinations are shown in an abbreviated form in brackets, for instance, the state (s, m) of *Rep1* is the state {*PUA*=*QF-execq, CRS*=*small, ERS*=*medium*}. For each table, all the optimal policy actions of a representation $r \in R$, which are different from their *Baseline* counterparts are shown in bold. In order to facilitate understanding, we have shown the different actions under each value of *CRS* (*small, medium, large, verylarge*) in separate columns. Let us briefly analyze the behavior of the learnt policies.

An interesting result is that the OP_{Base} learnt for models *WillUM*, *ModwillUM* and *AllUM* is the rigid policy (RP) of NutKing which we have improved in [Ricci and Mahmood, 2007], i.e., execute the query for small result sizes (*CRS*={*small, medium*}) and suggest tightening for larger ones (*CRS*={*large, verylarge*}). This shows that, under *Baseline*, RP is optimal for these classes of users. It would be also interesting to analyze how adding a feature to *Baseline* affects RP. The results show that adding features doesn't significantly change RP for states with *CRS*={*small, medium, verylarge*}. In fact, the total number of *different* optimal actions learnt (i.e., the action *Suggest* for *CRS*={*small, medium*} and *Execute* for *CRS*=*verylarge*) for these states are only 29 out of a total of 102 states, i.e., a difference of only $29/102 = 28.4\%$. On the contrary, a significant change is observed for states

Rep	Optimal Policies for states with *PUA=QF-execq*												
Baseline OP_{base}	s exec			m exec			l exec			vl sugg			
Rep1	(s,s) exec			(m,s) exec	(m,m) exec		(l,s) **sugg**	(l,m) exec	(l,l) exec	(vl,s) sugg	(vl,m) **exec**	(vl,l) **exec**	(vl,vl) sugg
Rep2	(s,s) exec	(s,m) exec	(s,l) exec	(m,s) exec	(m,m) exec	(m,l) exec	(l,s) **sugg**	(l,m) **sugg**	(l,l) exec	(vl,s) sugg	(vl,m) sugg	(vl,l) **exec**	
Rep3	(s,s) exec	(s,m) exec	(s,l) exec	(m,s) **sugg**	(m,m) exec	(m,l) exec	(l,s) exec	(l,m) exec	(l,l) exec	(vl,s) **exec**	(vl,m) sugg	(vl,l) sugg	
Rep4	(s,ac) exec	(s,mx) exec	(s,rj) exec	(m,ac) **sugg**	(m,mx) exec	(m,rj) exec	(l,ac) exec	(l,mx) exec	(l,rj) exec	(vl,ac) sugg	(vl,mx) **exec**	(vl,rj) sugg	

Table 6: Optimal Policies learnt under *GUM* (*s=small, m=medium, l=large, vl=very large, ac=accept, mx=mixed, rj=reject*)

Rep	Optimal Policies for states with *PUA=QF-execq*												
Baseline OP_{base}	s exec			m exec			l sugg			vl sugg			
Rep1	(s,s) exec			(m,s) **sugg**	(m,m) exec		(l,s) sugg	(l,m) sugg	(l,l) **exec**	(vl,s) sugg	(vl,m) sugg	(vl,l) sugg	(vl,vl) sugg
Rep2	(s,s) exec	(s,m) exec	(s,l) exec	(m,s) exec	(m,m) exec	(m,l) exec	(l,s) sugg	(l,m) **exec**	(l,l) **exec**	(vl,s) sugg	(vl,m) sugg	(vl,l) **exec**	
Rep3	(s,s) exec	(s,m) exec	(s,l) exec	(m,s) exec	(m,m) exec	(m,l) exec	(l,s) **exec**	(l,m) sugg	(l,l) **exec**	(vl,s) sugg	(vl,m) sugg	(vl,l) sugg	
Rep4	(s,ac) **sugg**	(s,mx) exec	(s,rj) exec	(m,ac) **sugg**	(m,mx) exec	(m,rj) exec	(l,ac) sugg	(l,mx) **exec**	(l,rj) **exec**	(vl,ac) sugg	(vl,mx) **exec**	(vl,rj) **exec**	

Table 7: Optimal Policies learnt under *WillUM* (*s=small, m=medium, l=large, vl=very large, ac=accept, mx=mixed, rj=reject*)

Rep	Optimal Policies for states with *PUA=QF-execq*												
Baseline OP_{base}	s exec			m exec			l sugg			vl sugg			
Rep1	(s,s) exec			(m,s) exec	(m,m) exec		(l,s) sugg	(l,m) **exec**	(l,l) **exec**	(vl,s) sugg	(vl,m) sugg	(vl,l) **exec**	(vl,vl) sugg
Rep2	(s,s) exec	(s,m) exec	(s,l) exec	(m,s) exec	(m,m) exec	(m,l) exec	(l,s) sugg	(l,m) **exec**	(l,l) **exec**	(vl,s) sugg	(vl,m) sugg	(vl,l) **exec**	
Rep3	(s,s) exec	(s,m) exec	(s,l) exec	(m,s) exec	(m,m) exec	(m,l) exec	(l,s) **exec**	(l,m) sugg	(l,l) **exec**	(vl,s) sugg	(vl,m) **exec**	(vl,l) **exec**	
Rep4	(s,ac) **sugg**	(s,mx) exec	(s,rj) exec	(m,ac) **sugg**	(m,mx) exec	(m,rj) exec	(l,ac) sugg	(l,mx) **exec**	(l,rj) **exec**	(vl,ac) sugg	(vl,mx) **exec**	(vl,rj) **exec**	

Table 8: Optimal Policies learnt under *ModwillUM* (*s=small, m=medium, l=large, vl=very large, ac=accept, mx=mixed, rj=reject*)

Rep	Optimal Policies for states with *PUA=QF-execq*												
Baseline OP_{base}	s exec			m exec			l exec			vl exec			
Rep1	(s,s) exec			(m,s) exec	(m,m) exec		(l,s) exec	(l,m) exec	(l,l) exec	(vl,s) **sugg**	(vl,m) exec	(vl,l) exec	(vl,vl) exec
Rep2	(s,s) exec	(s,m) exec	(s,l) exec	(m,s) exec	(m,m) exec	(m,l) exec	(l,s) **sugg**	(l,m) **sugg**	(l,l) **sugg**	(vl,s) **sugg**	(vl,m) **sugg**	(vl,l) exec	
Rep3	(s,s) exec	(s,m) exec	(s,l) exec	(m,s) exec	(m,m) exec	(m,l) exec	(l,s) exec	(l,m) exec	(l,l) exec	(vl,s) exec	(vl,m) exec	(vl,l) exec	
Rep4	(s,ac) exec	(s,mx) exec	(s,rj) exec	(m,ac) **sugg**	(m,mx) exec	(m,rj) exec	(l,ac) **sugg**	(l,mx) **sugg**	(l,rj) exec	(vl,ac) **sugg**	(vl,mx) **sugg**	(vl,rj) exec	

Table 9: Optimal Policies learnt under *UnwillUM* (*s=small, m=medium, l=large, vl=very large, ac=accept, mx=mixed, rj=reject*)

Rep	Optimal Policies for states with *PUA=QF-execq*												
Baseline OP_{base}	s exec			m exec			l sugg			vl sugg			
Rep1	(s, s) exec			(m, s) exec	(m, m) exec		(l, s) sugg	(l, m) sugg	(l, l) **exec**	(vl, s) sugg	(vl, m) **exec**	(vl, l) **exec**	(vl, vl) sugg
Rep2	(s, s) exec	(s, m) exec	(s, l) exec	(m, s) exec	(m, m) exec	(m, l) exec	(l, s) sugg	(l, m) sugg	(l, l) **exec**	(vl, s) sugg	(vl, m) sugg	(vl, l) **exec**	
Rep3	(s, s) exec	(s, m) exec	(s, l) exec	(m, s) exec	(m, m) exec	(m, l) exec	(l, s) **exec**	(l, m) sugg	(l, l) **exec**	(vl, s) sugg	(vl, m) sugg	(vl, l) sugg	
Rep4	(s, ac) exec	(s, mx) exec	(s, rj) exec	(m, ac) **sugg**	(m, mx) exec	(m, rj) exec	(l, ac) sugg	(l, mx) **exec**	(l, rj) **exec**	(vl, ac) sugg	(vl, mx) sugg	(vl, rj) **exec**	

Table 10: Optimal Policies learnt under *AllUM* (*s=small, m=medium, l=large, vl=very large, ac=accept, mx=mixed, rj=reject*)

with *CRS=large*, which yield a difference of 58.3%. Furthermore, the results for *GUM* and *UnwillUM* show a similar behavior for states with *CRS*={*small, medium*} (the difference being only 9.1%). Also, most differences are obtained for states with *CRS=verylarge* followed by states with *CRS=large*, with the respective percentages being 38.4% and 33.3% respectively (which are less than 58.3%. Generally speaking, these results imply that, if more features are added to *Baseline*, 1) it is best to execute the query for smaller result sizes, 2) the user population is not too willing to accept tightening even for large result sizes, and 3) it is best to suggest tightening only for very large result sizes.

We now determine the relevancy of our representation set R under each user model $um \in UM$. The results show that each representation in R is relevant under *GUM*, *WillUM*, and *ModwillUM*. However, *Rep1*, *Rep2* and *Rep4* are relevant under *UnwillUM* but *Rep3* is irrelevant. Under *AllUM*, each representation in *R* is relevant, i.e., for our user population, it is better to add all our proposed features to *Baseline*. These results again prove that the relevancy is influenced by the user behavior. We also note that the results for *GUM* and *ModwillUM* are similar to those for *OPEval*, but the results for the other models are different. Considering that *OPEval* is a more robust criterion for relevancy, this result shows that even if adding a new feature to some baseline representation allows the system to learn different optimal actions, it does not guarantee that the new OP is indeed a suitable policy.

5 Related Work and Conclusions

In this paper, we have addressed the problem of determining a relevant state representation, and we have shown that adding a new feature is not always beneficial for the system, and that the relevancy is influenced by the user behavior. Also, we have justified considering real-user behavior in order to learn the optimal strategy, which would allow us to determine relevancy of a larger representation set R, and also to depict the users' willingness level through a much larger system action set (rather than only on tightening/executing the query). To this end, we have applied our recommendation methodology within an online travel recommender system and are now running experiments with real users in the context of the etPackaging project funded by Austrian Network for E-Tourism (ANET). We have proposed a set of 10 state features for this system (along with a more detailed system action set), and we intend to develop a technique for performing a sequential feature selection on this set, in order to determine the relevant features. To the best of our knowledge, our work is the first attempt in addressing the relevancy problem in the domain of recommender systems. The relevancy problem has also been addressed in the domain of dialogue systems: [Tetreault and Litman, 2006] exploit the *OPComp* criteria to prove the relevancy of five state representations under a corpus of real-user sessions. However, their results need further validation because we have shown that simply learning different actions doesn't guarantee that the new policy is optimal for the users. Also, [Frampton and Lemon, 2006] adopt a similar criteria to *OPEval* in order to prove the relevancy of adding two dialogue features to a baseline representation.

References

[Frampton and Lemon, 2006] Matthew Frampton and Oliver Lemon. Learning more effective dialogue strategies using limited dialogue move features. In *ACL '06*, 2006.

[Resnick and Varian, 1997] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[Ricci and Mahmood, 2007] Francesco Ricci and Tariq Mahmood. Learning and adaptivity in interactive recommender systems. In *Proceedings of the ICEC'07 Conference*, August 2007.

[Ricci *et al.*, 2003] Francesco Ricci, Adriano Venturini, Dario Cavada, Nader Mirzadeh, Dennis Blaas, and Marisa Nones. Product recommendation with interactive query management and twofold similarity. In *ICCBR 2003, the 5th International Conference on Case-Based Reasoning*, 2003.

[Shimazu, 2001] Hideo Shimazu. ExpertClerk: Navigating shoppers buying process with the combination of asking and proposing. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001*, Seattle, Washington, USA, 2001.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[Tetreault and Litman, 2006] Joel R. Tetreault and Diane J. Litman. Using reinforcement learning to build a better model of dialogue state. In *EACL*, 2006.

[Thompson *et al.*, 2004] Cynthia A. Thompson, Mehmet H. Goker, and Pat Langley. A personalized system for conversational recommendations. *Artificial Intelligence Research*, 21:393–428, 2004.

Concept of an adaptive training system for production

Barbara Odenthal, Marcel Ph. Mayer, Morten Grandt, Christopher M. Schlick
RWTH Aachen University,
Institute of Industrial Engineering, and Ergonomics
D-52062, Aachen, Germany
{b.odenthal, m.mayer, m.grandt, c.schlick}@iaw.rwth-aachen.de

Abstract

The globalisation and the connected relocation abroad lead to a new situation for German enterprises. In order to react to the changing situation, the project "Integrative Production for High-Wage Countries" examines the research question under which conditions and with which methods and measures successful economic production in high-wage countries is feasible.

In order to find solutions, the project focuses, along with three other research fields, on self-optimising production systems. In this field a cognitive control system is emphasized. In order to cope with innovative and complex systems, workers will have to meet new challenges regarding the work requirements. Considering the effects on task performance a special training system is being developed in order to increase productivity and to support employees within the new tasks.

1 Introduction

One of the effects of globalisation in public view is the reduction of production in high-wage countries especially due to job relocation abroad to low-wage countries e.g. towards Eastern Europe or Asia [von Weizsäcker 2002, 2003]. Based on this a competition between manufacturing companies in high-wage and low-wage countries typically occurs within two dimensions: the production-oriented economy and the planning-oriented economy. Possible disadvantages of production in low-wage countries concerning process times, factor consumption and process mastering are compensated by low productive factor costs.

In contrast, companies in high-wage countries try to utilise the relatively expensive productivity factors by maximising the output (economies-of-scale). Another way to compensate the arising unit cost disadvantages is customisation or fast adaptation to market needs (economies-of-scope). But the escape into sophisticated niche markets does not seem to be a promising way for the future anymore.

Within the second dimension – the planning-oriented economy – companies in high-wage countries try to optimise processes with sophisticated, investment-intensive planning approaches and production systems while companies in low-wage countries implement simple, robust value-stream-oriented process chains. Since processes and production systems do not exceed the limits of an optimal operating range, additional competitive disadvantages for high-wage countries emerge.

It cannot be sufficient to only achieve a better position within one of the dichotomies "scale vs. scope" or "planning-orientation vs. value-orientation", hence the research question must aim at solving both dichotomies. Economies-of-scale and economies-of-scope must be maximised at the same time, while additionally the share of added-value activities must be further maximised without neglecting the planning quality to finally achieve a sustainable competitive advantage for production in high-wage countries.

2 General Research Approach

Cluster of Excellence "Integrative Production for High Wage Countries"

To achieve this goal the Cluster of Excellence "Integrative Production for High Wage Countries" was initiated at RWTH Aachen University. The cluster aims at contributing to an extended production theory that describes the interrelations and interdependencies between the single elements of production systems and ensures efficient resource deployment. The four main research areas "Individualised Production", "Virtual Production Systems", "Hybrid Production Systems" and "Self-optimising Production Systems" - are oriented towards the vision of an Integrative Production Technology.

The issues of optimisation and mastering complexity are addressed by the fourth field "Self-optimising Production Systems". Improving the use of cognition is the preferred way to develop and evaluate more rigorous, model based and systematic methods for reducing the dilemma of scale and scope towards an intelligent manufacturing environment.

The essential step is the application of cognitive control mechanisms. Cognitive control should enable arbitrary production processes on different production levels. Such a system will be able to obtain, store, transform and use knowledge for self-analysis and self-optimisation with respect to the mutable optimisation objectives.

Cognitive Control System

In order to reduce the personnel costs and to increase the system's availability and reliability, enterprises enhance the automation of the production systems. However, in doing so the flexibility of the system decreases and the expenditure during the planning phase and time of planning increase. Therefore the project part „Cognitive Control Systems“ deals with the conception and development of a cognitive and „intelligent“ control system for production systems which enables a high flexibility by reducing the expenditure of planning.

In order to find a solution for this research task, the challenge lies in the developing of a control system which will be applied to a concrete use case in the area of production. Under variable conditions based on insecure and/or incomplete information, the cognitive system should be able to optimise the concerned production processes either autonomously or in cooperation with the (human) operator. In figure 1 the basis of an architecture of a cognitive control system is presented.

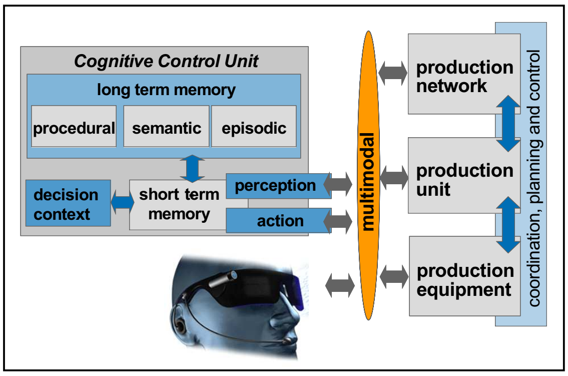


Figure 1: Architecture of a cognitive control system

Cooperation between human operators and autonomous systems in direct interaction or supervisory control requires an adequate system transparency so that the operator is able to build up a mental modal of the system's behaviour and to comprehend the system's actions.

Therefore the basic framework of this system represents an architecture which is similar to well-known models of human information processing. To enable the human operator to interact with the cognitive control system, a multimodal human-machine interface is provided.

Nevertheless, to obtain a full understanding of this novel technology, new concepts and methods are essential to improve and to build the required competence and qualification of the operator. Because of that, the project includes the development of an adaptive instrument for embedded training in order to impart the essential knowledge and the competence to the operator. Since this training functionality shall be integrated in the working process, it is essential to provide the worker with situational relevant information in an intuitive way. One solution to achieve this is the use of technologies from the field of augmented reality (AR).

3 Adaptive AR-supported Training System

In the field of production especially in manual assembly, there are already several AR based systems to support the worker by providing detailed instructions. That means common print out handbooks can be replaced by the use of AR-based assembly guidance (Wiedenmaier, 2004). But these systems have not been accepted in practical applications in production yet. One possible reason for minor acceptance is the fact that the system provides the same information in an identical way respective sequence for each user without taking into account the pre-knowledge, the experiences and the individual capacity of the user. Hence an adaptive Training-System shall be developed. This system will be able to adapt to the individual way of learning of each user. In order to achieve this, a modular structure is essential.

Several categories have to be taken into account developing a modular Teaching-/Learning-/Training-System (TLT-System). On the one hand the focus is on the learner with certain qualifications, competences and capabilities related to the task subject to learn, whereas each learner has individual strategies (learning type) to learn in an effective way. On the other hand, based on the task subject to learn, there are certain requirements to the TLT-System, e.g. the content of learning. The task and the learner combined with the technology used to carry knowledge, have a significant influence on the design of the learning modules. Figure 2 illustrates an integrated approach (technology and leaming modules, learner and task) which finally shall result in an adaptive TLT-System.

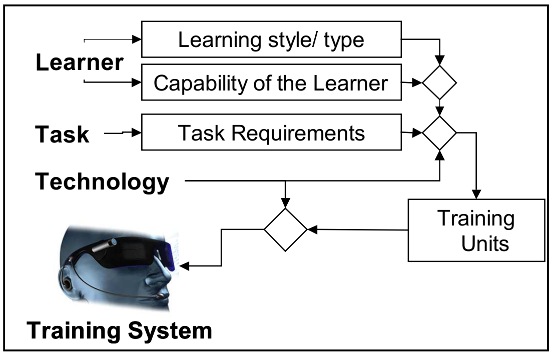


Figure2: Teaching-/Learning-/Training-System

Learner

Different learners often achieve different learning results under equal learning conditions because of differing previous knowledge, motivation and intellectual abilities of the individual learner. Existing/previous TLT systems, especially in the production area, provide standardised learning conditions which are independent of individual user's abilities. This implies the adaptation of the user to the system. Within this project, a TLT system will be developed which adapts the technology to the user. Because the success of the learning process is dependent on the abilities of the individual among other things, a classification of the users must be developed by means of certain attributes of the individual learning process. In the psychological and the educational literature as well as the literature of the cognitive science validated classifications of learning types and styles can be found (see e.g. Cassidy 2004). To improve and optimise the learning process, a process oriented classification is considered to be reasonable. The human learning process can be divided in four cyclic steps:

1) **Making experiences:** collection of data and information from tests and personal experiences
2) **Reflection / cogitation:** observation and reflection lead to an analysis of the meaning of the data
3) **Drawing conclusions:** abstract formation of concepts generates abstract models and mental patterns
4) **Testing of the drafts / concepts / action:** executing new operations, maximising desired effects, proofing the models, planning further steps

The learner classification according to Honey und Mumfort [1992] is based upon the fact, that each learner passes through the four steps of the learning process. However, according to the individual learning type the learner prefers different steps which are used with differing intensity in order to learn.

Task

Regarding the above described cognitive control system the main tasks of the operator in a production area are:

- initial start-up of the system
- monitoring system's operation
- intervention in the case of system errors

In order to support the worker in the above mentioned areas by means of the TLT-system, it is essential to present/teach not only the knowledge but also the strategies for using the knowledge during the decision-making process. The essential knowledge to perform the occurring tasks will be raised by means of a task analysis. A cognitive task analysis is applied to identify the relevant mental processes, requirements and strategies to successfully cope with the task. During the human information processing differing types of behaviour occur depending on the user's standard of knowledge and practise which can be classified by Rasmussen [1987] who differentiates between skill-, rule- and knowledge-based behaviour.

Skill-based behaviour represents the sensomotor performance of operations which occur without a conscious regulation as an automatic, uniform and highly integrative pattern of behaviour. Rule-based behaviour is defined as follows: in known working situations the cognitive subroutines are consciously regulated by stored rules. In unknown, new situations, for which no known procedures exist, the knowledge respective model-based behaviour takes effect. The action goals are formulated explicitly based on a probalistic situation analysis and personal preferences. A strategy is then developed by evaluating alternative plans with respect to their fulfilment regarding the action goals.

In order to teach and train each type of behaviour, different manners of presentation are required. The results of the task analyses are used in order to identify the best manner of presentation and to allocate the requisite knowledge, the strategies of the decision-making process and the required behaviour to tasks.

Technology

To integrate the learning process in the working process and to offer the possibility to work simultaneously, one possible solution is the use of augmented reality technology in the appearance of head-mounted-displays (HMDs) because with this technology the user's attention is not distracted from the object of interest when additional virtual information is supplemented in the field of view [Azuma et al. 2001; Genc et al 2002]. The superimposed information extends the reality in a way that the required information reaches the user at just the right time at just the right location.

In order to identify the specific application, different requirements must be fulfilled. For example, it is essential to know how much information and in which form the information shall be integrated, because this determines the required field of view. Beside this, the distinguishing features of the HMDs are for example the representation of virtual objects (e.g. resolution) and mechanical attributes like weight. Hence an analysis regarding those attributes has to be performed. Furthermore, ergonomic aspects have to be taken into account.

Learning modules

The learning modules of the modular system must be adapted to the content, the learning process and the technology applied. In order to evaluate the system, laboratory tests will be executed. The aim is to compare the success of the learning process supported by the developed system to conventional learning in order to make a statement how the success of learning can be increased effectively.

4 Summary

This paper describes a proposal for the development of an adaptive, AR-based training-system which is embedded in the working process. Based on an overview of the initial situation the overall research questions and the general research approach were described.

The relevant aspects of the training system like user, task and technology were discussed and the general approach for evaluation was pointed out.

Acknowledgements

The depicted research is funded by the German Research Foundation DFG as part of the Cluster of Excellence "Integrative Production Technology for High-Wage Countries".

References

[Azuma *et al.*, 2001] Azuma, R.T.; Baillot, Y.; Behringer, R.; Feiner, S.; Julier, S.; MacIntyre, B.: *Recent Advances in Augmented Reality*. In: Computers & Graphics, 2001, S. 1-15.

[Cassidy, 2004] Simon Cassidy. *Learing Styles: An overview of theories, models, and measures.* Educational Psychology, 24:4, 419 – 444, 2004.

[Genc *et al.*, 2002] Genc, Y.; Tuceryan, M.; Navab, N.: *Practical Solutions for Calibration of Optical See-Through Devices.* In: Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality, 2002, S. 169-175

[Honey und Mumfort, 1992] Honey, P., Mumford, A. *The Manual of Learning Styles*. Maidenhead: Berkshire, 1992

[Rasmussen, 1987] Jens Rasmussen. *Reasons, Causes, and Human Error*. New Technology and Human Error, 1987

[Wiedenmaier, 2004] Stephan Wiedenmaier: *Unterstützung manueller Montage durch Aumented Reality-Technologien.* Dissertation. Aachen: Shaker, 2004.

[von Weizsäcker, 2002] E. U. von Weizsäcker: *Globalisierung der Weltwirtschaft – Herausforderungen und Antworten*. http://www.bundestag.de/gremien/welt/glob_end, sighted: 27.04.2007.

[2003] *Produktionsverlagerung als Element der Globalisierungsstrategie von Unternehmen. Ergebnisse einer Unternehmensbefragung.* In: Deutscher Industrie- und Handelskammertag. Berlin, Brüssel, 4ff. http://www.heilbronn.ihk.de/upload_dokumente/infothek/anlagen/6331_1886.pdf, sighted: 27.04.2007.

Towards Asynchronous Adaptive Hypermedia: An Unobtrusive Generic Help System

Andreas Putzinger
Johannes Kepler University of Linz
A-4040, Linz, Austria
putzinger@fim.uni-linz.ac.at

Abstract

First, this paper introduces the concept and the upcoming features of Asynchronous Adaptive Hypermedia Systems (AAHS). The design of a concrete system will show how the new principles can successfully be applied to build a generic adaptive help module which can be put on top of existing adaptive or non-adaptive web application without the need of refactoring.

1 Introduction

It is widely acknowledged that Adaptive Hypermedia Systems (AHS) can successfully be applied to several different application domains. The first summarizing taxonomy was published in [Brusilovsky, 1996] and later updated in [Brusilovsky, 2001]. Although several research communities with special focus on adaptivity reacted to the upcoming web by establishing AHS, technologies, commonly referred to as "web 2.0" (cf.[O'Reilly, 2005]) have yet to make their mark on the scene.

The author's current focus of research lies in the concept of out-of-band communication in AHS. In this context the term "asynchronous" is used quite often, whereas "asynchronicity" refers to the actual transmission of data, which takes place independently of the main HTTP request-response cycle. The term "asynchronicity" is used in this paper to refer to the concept of out-of-band communication.

So far it seems that to date very few research groups have published results yet, which would specifically focus on the impact of an out-of-band communication on AHS, related privacy issues or the range of upcoming features. [Barla, 2006] uses asynchronous techniques to get more precise information from the client's context. [Boddu *et al.*, 2007] show how the AHA! framework (cf.[de Bra *et al.*, 2002]) could be enhanced by applying asynchronous concepts.

2 Introduction to the Concept of AAHS

2.1 "Asynchronous Web" in General

Techniques for realizing out-of-band communication in the web are widespread and provide the substantial base for many so–called Rich Internet Applications (RIA, first published in [Allaire, 2002]). A rather well known acronym in this context is AJAX (Asynchronous JavaScript and XML), which denotes a set of technologies often used together. The term itself was first introduced in [Garrett, 2005]. One main goal is to bridge the gap between desktop and web applications in several aspects, mainly in communication and latency matters.

The question may arise as to how far this asynchronous technology can actually provide new possibilities and features. Since asynchronously transferred data could theoretically be also transmitted "synchronously" by transparently bundling it with the next page–request; therefore, all messages have to be collected and cached locally at the client and "piggybacked" on the next HTTP request. Yet, on closer examination this technique is not of equal potential as asynchronous transmissions. If the user, for instance, manually closes the browser window, all accumulated data from the point of entering the page until the point of leaving are lost. In addition, and this represents one of the main drawback, the advantage of a communication with only a short latency is lost. Furthermore, it is still not possible for the server to initiate a call to the client. Thus piggybacking data is not always an alternative to an out-of-band communication. Nevertheless, specific privacy issues mentioned in [Putzinger, 2007] as well as security issues discussed in [Sonntag, 2006; Di Paola, 2006] have to be considered.

2.2 Specific Application in AHS

The combination of AHS on the one side and out-of-band communication techniques on the other opens a great variety of new possibilities in the adaptive hypermedia field and will, at least in the opinion of the author, start a new era of AHS. Enhanced adaptive technologies empowered by stable bidirectional channels between browser and server will, for instance, not only ease the provision of feedback for users[1] and enable advanced usage of subsymbolic data from the client side (cf.[Hofmann *et al.*, 2006; Farzan and Brusilovsky, 2005]), but will also form the base for new kinds of adaptations already known from more traditional desktop adaptive systems.

The upcoming concept of asynchronicity in AHS inherently changes many modules involved in such a system. First, the possibilities for retrieving raw data from the user's context is broadened as far as latency and quantity are concerned. The information about the user's current actions within the browser can, for example, be transmitted almost in realtime. This has a direct influence on the point in time when the process of modeling can take place. Instead of traditionally triggering the modeling process on an incoming page request, data is continously being retrieved and can therefore continuously be processed. Furthermore, triggering adaptations can be done at any time. In traditional AHS the adaptation takes place once when the page is created. Also people involved in the process of evalua-

[1] Amazon, for instance, uses out-of-band calls when users perform product ratings.

tion can heavily benefit from the new quantity and quality of data. This is particularly true for meta adaptive systems, which need to perform self evaluation as part of the standard adaptive behaviour.

The following paragraphs show some examples for new low-level techniques together with their high-level impact in adaptivity. More detailed information can also be found in [Putzinger, 2007]:

Monitoring the user's mouse In some cases it could be helpful to a system to get realtime information about the mouse activity on the clientside. Specifically, this could be the current position of the mouse cursor, the object, text or picture which is currently under the cursor. Also mouse movements or miscellaneous timing data are probably interesting, such as the speed of movements, latencies between (double-)clicks, etc. These data obviously represent valuable raw material for applying methods of subsymbolic user behaviour inference. [Atterer *et al.*, 2006], for instance, suggest to record mouse actions for website usability evaluation.

Monitoring Key Strokes A second category of usage data contains raw key strokes, which could be transmitted either key-by-key in realtime or also grouped. Thus, the application not only gets the finally submitted form data, but also the intermediary states, the involved timings, etc. The user's typing speed together with some other aspects can in some cases be regarded as a good indicator for the user's overall computer skills. A second example refers the possibility of introducing adaptive text completion or recommendation. Adapted to the user model the system suggests words or even complete paragraphs, which fit in with the context and presumably the user's current needs.

The two just mentioned categories of events can individually or combined improve results in plan recognition (cf.[Carberry, 2001; Kristina *et al.*, 1996]) by modeling a clearer picture of the user's current activities.

"Still Active" Messages Receiving the information about key strokes or mouse events out-of-band is an implicit and quite reliable indicator for deducing the binary state if the user is still working with the application. If neither of this data is asynchronously transmitted, explicit "still active" messages could be introduced in order to inform the system about the user's activity state. In particular, this could be used, for example, in e-learning environments, where users gets lots of material to locally read and learn. Although there is no synchronous interaction with the server in regular intervals, the learning platform could use the data about the activity to deduce further information.

On–Demand Data Retrieval Due to the bidirectional communication channel it is also possible for the server to (at least logically) initiate a communication and to push data to the browser without a prior client request. Whenever specific information are needed from the client, the server can simply ask for it. The communication itself is done out-of-band therefore probably even without the user's awareness.

Instant Adaptation The author has developed facilities which allow page fragments to be dynamically exchanged according to results of the underlying adaptive system. The chosen name for this technique is "*instant adaptation*". The effects of changes to the user model, which in turn cause (visual) changes on the user's current page, can instantly be pushed to the client as fragments. Thus, the possibilities for adaptation are becoming much richer. In traditional AH the actual adaptation takes place once when the page is generated. From this point onwards, the page keeps static in respect of adaptation, because it is sent back to the client and no further adaptation takes place until the next complete page is generated.

The technique of instant adaptation enables AHS to exchange selected parts within pages on the fly, although are already shown in the browser, by pushing the new fragment to the client. Some trivial client logics dynamically replaces, adds or modifies the specified part. This technology in context of AHS is new and seems to be quite powerful. Nevertheless, the designer of such a system has to be very careful in using these methods. Studies have shown, that a dynamically changed user interface often confuses people and therefore does not always have a positive impact on the overall user experience.

3 Technical Means of Transport

HTTP is still the default transport protocol in the web. It is good for serving traditional sites and applications but shows some shortcomings since web 2.0 features are required. The concept does not foresee, for instance, to load or send data in the background, least of all on a bidirectional channel. The author has investigated several techniques to overcome some of these limitations. In the rest of this section, some solutions are briefly discussed.

XmlHttpRequest is the most often used technique to implement asynchronous web systems. It is well supported by browser implementations, even on mobile devices, and inherently part of many current application frameworks. It's specification provides support for out-of-band calls from the client to the server, but not vice versa. This can be overcome, for example, by periodically polling the server or more elegantly by using a so called continuous http connection aka. "Streaming AJAX" (cf.[Alinone, 2005]) or "Comet". Comet is the technique finally selected for implementing the system described below.

Besides XmlHttpRequest, several other techniques exist which often require special objects to be included in web pages. These objects are used as intermediaries for the actual communication, as shown in fig.1. Javascript within the webpage sends data to a well-defined interface (①) of the proxy. The object performs the actual communication with the server (②) which takes place independently of the browser's main communication cycles (③). The kind of communication (HTTP, web service call, socket connection, etc.) depends on the kind of embedded object as well as the applying sandbox restriction policies. If the server wants to push data to the page, it is sent to the proxy object, which again informs the page about the received data by calling a specified javascript method. The author has by now successfully tested both Java Applets and Flash movies as technical intermediaries. Furthermore, there is no need to display the embedded objects on the page but they can, in fact, be hidden or even be placed within an (invisible) frame.

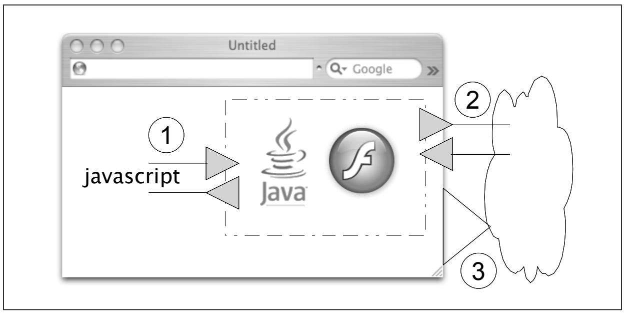


Figure 1: Proxy concept for out-of-band communication

4 Conceptual Design of an AAHS for Help Provision

An adaptive system $\mathcal{S}$ should be designed, which can generically be used within especially input–based web applications or sites without prior information about users. $\mathcal{S}$ should be plugged on top of an existing application $\mathcal{A}$. From a technical point of view, $\mathcal{S}$ and $\mathcal{A}$ should only be losely coupled. A kind of intermediary on any side of the communication adds some lines of javascript code to the pages provided by $\mathcal{A}$ before the pages gets delivered to the user. The additional code instruments the pages to interact with $\mathcal{S}$ autonomously. Furthermore, $\mathcal{S}$ completely relies on asynchronously transmitted data to continuously observe the users' behaviour on the site.

The following features must be carried out by $\mathcal{S}$:

- Generically determine situations in $\mathcal{A}$ when the user needs help in context of the currently performed action. Offer help in these cases.
- Generically determine when the user needs help in using the system, independent of the current context. Offer general help in these situations.
- $\mathcal{S}$ must behave as unobtrusive as possible. This concerns data collection as well as the adaptation processes (providing help).
- $\mathcal{S}$ may not depend on any existing user models or information about users, but works autonomously.
-

4.1 Data Collection

The following data are asynchronously and independently of $\mathcal{A}$ transmitted to the server, whereas every message contains additional information to uniquely identify the user and the current page:

- Mouse movements
- Keystrokes
- Changes of the currently focused input element
- Global events, such as focusing and blurring the window, scroll actions, etc.

4.2 Modeling and Inference

Every adaptive system needs some specific aspects to be modeled. The subsequent sections show the most important ones for $\mathcal{S}$, whereas the following axioms and definitions hereby apply:

- Let $\mathcal{P}$ globally be the *current selected page* in $\mathcal{S}$.
- Let $\mathbf{U}$ be the set of *all existing users* in $\mathcal{S}$ who have already visited $\mathcal{P}$.

$$\mathbf{U} = \{\mathcal{U}_i \,|\, \mathrm{is\_system\_user}(\mathcal{U}_i, \mathcal{S}) \text{ and } \mathrm{has\_visited}(\mathcal{U}_i, \mathcal{P})\}$$

- Let $\mathcal{U}$ be *any (single) user* $\in \mathbf{U}$.
- Let IE be the set of all input elements on $\mathcal{P}$ in $\mathcal{S}$.

$$\mathsf{IE} = \{\varepsilon_i \,|\, \mathrm{is\_input\_element}(\varepsilon_i, \mathcal{P})\}$$

- Let $\mathrm{char\_count}(\varepsilon, \mathcal{U}_i)$ be a function which returns the number of filled–in characters in an input element ε for a user $\mathcal{U}_i$ on $\mathcal{P}$ in $\mathcal{S}$.

User Idle Time

The point in time of the latest user interaction is a very useful information to deduce further data, such as, for example, "time spent reading" in e–Learning systems (cf.[Farzan and Brusilovsky, 2005; Hofmann *et al.*, 2006]). Equ.1 shows the calculation of $\mathcal{U}$'s idle time which is basically the delta of the current time now and the latest point in time last when $\mathcal{U}$ performed an interaction.

$$\mathrm{it}(\mathcal{U}) = \mathrm{now}() - \mathrm{last}(\mathcal{U}) \qquad (1)$$

$\mathcal{S}$ currently recognizes mouse interactions (movements), key interactions (releasing a key) and changes to the current window state (focusing, blurring and scrolling). Furthermore, variants of it are defined which only take individual kinds of interactions into account. Therefore, $\mathrm{it}_{\mathrm{mouse}}$ considers mouse-events only, $\mathrm{it}_{\mathrm{key}}$ key-events and $\mathrm{it}_{\mathrm{window}}$ only changes to the current window state.

Locus of Attention[2]

Another aspect to model concerns $\mathcal{U}$'s attentional focus, formally expressed as $\mathrm{focus}(\mathcal{U})$. A large portion of research on human attention in digital environments is based on the findings of cognitive psychology. "For example [Raskin, 2000] analyses how single locus of attention, and habit formation have important consequences on human ability to interact with computers. [...] Attention therefore refers to the set of processes by which we select information" ([Roda and Thomas, 2005]). Several sensory-based mechanisms for the detection of users' attention have been employed, including gaze tracking, gesture tracking, head pose and acoustic tracking (cf.[Stiefelhagen *et al.*, 2001]). [Horvitz *et al.*, 2003] propose that sensory-based mechanisms could be integrated with other cues about $\mathrm{focus}(\mathcal{U})$. [Horvitz *et al.*, 2003] also strengthen the theory of uncertainty and suggest to turn to models that can be harnessed to reason about a users attention and about the ideal attention-sensitive actions to take under uncertainty. [Horvitz *et al.*, 2003] think that such models and reasoning can unleash new functionalities and user experiences, which completely aligns with the author's opinion. [Owen, 2006] presents first results in tracking the user's attention by tracking the mouse position in browsers.

In the current preliminary version of $\mathcal{S}$, the possible loci of attention on $\mathcal{P}$ are a priori restricted to input elements $\varepsilon \in \mathsf{IE}$. $\mathcal{U}$'s possible focus is determined by simply taking the currently activated, technically spoken "focused", element on $\mathcal{P}$, which is formally represented by ε_{sel}. If no

[2] In this paper the terms "locus" and "focus" of attention are used synonymously, whereas some authors differentiate more strictly, cf.[Raskin, 2000]

input element is activated, i.e. ε_{sel} is undefined, $\mathcal{U}$'s attentional focus can not be determined by $\mathcal{S}$ in the first place. It is planned to enhance $\mathcal{S}$ to support more abstract kinds of loci, such high level controls placed on $\mathcal{P}$.

The single information about ε_{sel} is a very unsure hint to determine $\mathrm{focus}(\mathcal{U})$. Therefore, it seems necessary to quantify the probability of the correctness, which is expressed as $\mathrm{p_{cor}}$ (see equ.2). The higher the value of the function $\mathrm{p_{cor}}$ is, the higher the chance can be assessed that $\mathrm{focus}(\mathcal{U}) = \varepsilon_{sel}$.

The used assessment function is based on the idea that at moments when $\mathcal{U}$ is typing text into ε_{sel}, the locus of attention is known quite precisely, i.e. the activated form field itself, because $\mathcal{U}$ is obviously just concentrated on writing. The higher, however, the value for $\mathrm{it_{key}}$ gets the lower the probability is that ε_{sel} still represents $\mathrm{focus}(\mathcal{U})$. A reciprocal exponential function has been chosen as the base form to express the probability of correctness, as shown in equ.2. The factor τ is customizable and determines the time span (in seconds) after which $\mathrm{p_{cor}}$ results in 50% probability. τ, therefore, parameterizes the aspect ratio of the curve. Fig.2 shows $\mathrm{p_{cor}}$ for $\tau = 20$ seconds.

$$\mathrm{p_{cor}}(\mathrm{it_{key}}(\mathcal{U}),\tau) = \frac{1}{1+(\frac{\mathrm{it_{key}}(\mathcal{U})}{\tau})^2} \qquad (2)$$

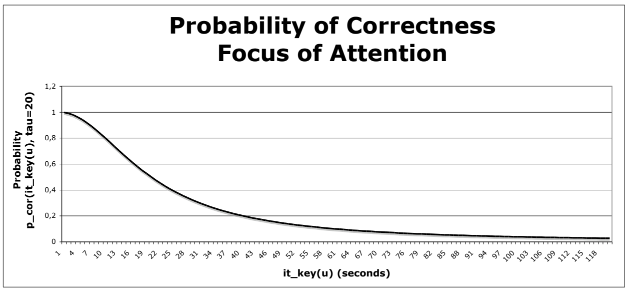


Figure 2: $\mathrm{p_{cor}}(\mathrm{it_{key}}(\mathcal{U}), \tau = 20)$

To finally decide whether $\mathcal{U}$ pays attention to ε_{sel} or not a parameter p_{limit} is introduced which represents the threshold of probability. Equ.3 finally shows the determination of the focus of attention against $\mathcal{U}$ and τ.

$$\mathrm{focus}(\mathcal{U},\tau) = \begin{cases} \varepsilon_{sel} = \text{undefined} & : \text{undefined} \\ \mathrm{p_{cor}}(\mathrm{it_{key}}(\mathcal{U}),\tau) > p_{limit} & : \varepsilon_{sel} \\ \mathrm{p_{cor}}(\mathrm{it_{key}}(\mathcal{U}),\tau) \leq p_{limit} & : \text{undefined} \end{cases} \qquad (3)$$

$\mathcal{S}$ has to determine after how many seconds p_{limit} is reached. For this purpose the inverse function to $\mathrm{p_{cor}}$, $\mathrm{inv_{p_{cor}}}$, is used, which is shown in equ.4.

$$\mathrm{inv_{p_{cor}}}(\mathrm{p_{cor}},\tau) = \sqrt{\tau^2(\frac{1-\mathrm{p_{cor}}}{\mathrm{p_{cor}}})} \qquad (4)$$

Decision for Context Sensitive Help

Another aspect to model is the actual probability that $\mathcal{U}$ needs context sensitive help for ε_{sel}. This kind of help is offered if $\mathcal{U}$ is spending at least $threshold_{as}$ percent longer attention to ε_{sel} than the average of the other users do. The number of seconds $\mathcal{U}$'s attentional focus lies on ε is called "attention span" and is formally expressed by the function $\mathrm{as}(\mathcal{U},\varepsilon)$. Equ.5 defines the average attention span $\mathrm{as_{avg}}$ for a specified ε against all users, whereas equ.6 exlicitely disregards $\mathcal{U}$.

$$\mathrm{as_{avg}}(e) = \frac{\sum\limits_{\mathcal{U}_i \in \mathbf{U}} \mathrm{as}(\mathcal{U}_i,\varepsilon)}{|\mathbf{U}|} \qquad (5)$$

$$\mathrm{as_{avg}}(\mathcal{U},\varepsilon) = \frac{\sum\limits_{\mathcal{U}_i \in \mathbf{U}, \mathcal{U}_i \neq \mathcal{U}} \mathrm{as}(\mathcal{U}_i,\varepsilon)}{|\mathbf{U}|-1} \qquad (6)$$

This inference obviously needs training. Thus, it would be an option to e.g.use a default value for the first $minusers$ instead of $\mathrm{as_{avg}}(\varepsilon)$. In this preliminary version of $\mathcal{S}$, $\mathcal{U}$'s individual speed factor is not taken into account. The personal speed, depending, for instance, on overall computer skills, etc., could also contribute to improve $\mathcal{S}$'s performance and to minimize the just mentioned bootstrap problem, which is based on the lack of user data and therefore experience when starting $\mathcal{S}$ for the first time or for new pages. Furthermore, in more advanced versions of $\mathcal{S}$, possible disabilities of $\mathcal{U}$ should be recognized and specially taken into account.

Example

- Let $\mathrm{as_{avg}}$ for ε_{sel} be 18 seconds.
- Let $\mathcal{S}$ be configured to offer $\mathcal{U}$ help in cases where $\mathrm{as}(\mathcal{U},\varepsilon_{sel})$ is at least 33% higher than the average value. Therefore, $threshold_{as} = 33\%$.
- Let p_{limit} be 60%. This means that $\mathcal{S}$ must be up to 60% sure that $\mathcal{U}$'s focus of attention can be correctly determined by $\mathcal{S}$.
- Let $\mathrm{p_{cor}}$ be parameterized a way that it returns 50% after 12 seconds; therefore, $\tau = 12$.

Effect

- If $as(\mathcal{U},\varepsilon_{sel})$ gets larger than 24 seconds, $\mathcal{S}$ triggers context sensitive help.
- If e.g.$as(\mathcal{U},\varepsilon_{sel})$ = 6 seconds and $\mathcal{U}$ stops typing, $\mathrm{p_{cor}}$ after 24 seconds (at this time $\mathrm{it_{key}}(\mathcal{U}) = 18$ seconds) is calculated as shown in equ.7. The resulting value is less than the required 60%, which results in not offering help.

  $$\mathrm{p_{cor}}(\mathrm{it_{key}}(\mathcal{U}) = 18, \tau = 12) = \\ \approx 30,77\% < p_{limit} = 60\% \qquad (7)$$

  The limit for this configuration can be determined by $\mathrm{inv_{p_{cor}}}(\mathrm{p_{cor}} = 60, \tau = 12) =\approx 9,8$. Therefore, the first 10 seconds of $\mathrm{it_{key}}$ are added to the current value of $as(\mathcal{U},\varepsilon_{sel})$, so that the new value for $as(\mathcal{U},\varepsilon)$ is 16.
- If $\mathcal{U}$ restarts typing (after whatever time span) and does not blur ε_{sel} in the next 8 seconds, help will be offered after 8 seconds, when $as(\mathcal{U},\varepsilon_{sel}) = 24$. This happens even if the user only presses one single key and immediatly stops typing again, as shown in equ.8.

  $$\mathrm{p_{cor}}(\mathrm{it_{key}}(\mathcal{U}) = 8, \tau = 12) = \\ \approx 69,23\% \geq p_{limit} = 60\% \qquad (8)$$

$\mathcal{U}$'s progress on $\mathcal{P}$

To determine if $\mathcal{U}$ probably needs context insensitive help $\mathcal{U}$'s overall progress on $\mathcal{P}$ will be taken into account and has therefore to be calculated in a first step. The following simple model is applied:

To determine $\mathcal{U}$'s overall progress on $\mathcal{P}$ the number of characters in each field $\varepsilon \in \mathbb{E}$ is compared with the average number of characters of that field of other users $\mathcal{U}_i \in \mathbf{U}, \mathcal{U}_i \neq \mathcal{U}$. If the factor is higher than 100% ($\mathcal{U}$ has more typed more text than average), the value is defined to be 100%. Equ.9 shows the calculation of the progress factor for a single ε, equ.10 for the whole page $\mathcal{P}$. In later versions of $\mathcal{S}$ more high-level controls will also be taken into account besides text input fields.

$$\mathrm{prog}_{\varepsilon}(\varepsilon, \mathcal{U}) = \frac{\sum\limits_{\mathcal{U}_i \in \mathbf{U}, \mathcal{U}_i \neq \mathcal{U}} \mathrm{char\_count}(\varepsilon, \mathcal{U}_i)}{|\mathbf{U}| - 1} \tag{9}$$

$$\mathrm{prog}(\mathcal{U}) = \frac{\sum\limits_{\varepsilon \in \mathbb{E}} \mathrm{prog}_{\varepsilon}(\varepsilon, \mathcal{U}) \leq 1 \begin{cases} \text{yes} & : \ \mathrm{prog}_{\varepsilon}(\varepsilon, \mathcal{U}) \\ \text{no} & : \ 1 \end{cases}}{|\mathbb{E}|} \tag{10}$$

Decision for Context Insensitive Help

Here, the term "context insensitive help" is used in a sense which designates general help about the usage of the overall system, in contrast to specific, context sensitive help for single input elements on $\mathcal{P}$. Generally, context insensitive help should be offered in cases when $\mathcal{S}$ determines that $\mathcal{U}$ may be generally confused with the usage of $\mathcal{S}$.

The concept which is used here to determine the "confusion probability" is not only based on the actual time span $\mathcal{U}$ is spending on $\mathcal{P}$ but mainly on $\mathcal{U}$'s individual progress on $\mathcal{P}$. The total number of $\mathcal{U}$'s interactions on $\mathcal{P}$ is summed up and set in relation to $\mathcal{U}$'s current progress. If the resulting factor differs too much from average, $\mathcal{S}$ triggers the offer to provide general help.

Following functions support the actual calculation:

- Let $\mathrm{t_{start}}(\mathcal{U})$ be the time from the point of loading $\mathcal{P}$ until $\mathcal{U}$ starts working, whereas "working" is restricted to typing.
- Let $\mathrm{avg}_{\mathrm{t_{start}}}()$ be a function which returns the average of $\mathrm{t_{start}}$ for all $\mathcal{U}_i \in \mathbf{U}$.
- Let $\mathrm{num_{interaction}}(\mathcal{U})$ be the total number of interactions $\mathcal{U}$ performed on $\mathcal{P}$ to reach the current progress.
- Let $\mathrm{avg}_{\mathrm{num_{interaction}}}(progress)$ be a function which returns the average number of interactions for all $\mathcal{U}_i \in \mathbf{U}, \mathcal{U}_i \neq \mathcal{U}$ to reach the specified $progress$.

Furthermore, the fact has to be taken into account that when entering the page (identified by low progress value) the chance of needing help is much higher than later on. Additionally, the fact that $\mathcal{U}$ needs longer than average to start working is regarded as additional hint for the fact that $\mathcal{U}$ maybe needs help. Because many users orientate first when entering a page, a certain minimum level of deviation may be granted from the beginning.

Equ.11 shows a reciprocal exponential function similar to $\mathrm{p_{cor}}$ in equ.2. Possible values for the parameter $progress$ lie between 0 ($progress = 0\%$) and 1,0 ($progress = 100\%$). The corresponding function results of $\mathrm{dev}(progress)$ are 1,0 for $progress = 0\%$, i.e. the user has not filled in anything, and $\approx 0,01$ for $progress = 100\%$.

$$\mathrm{dev}(progress) = \frac{1}{1 + 100 progress^2} \tag{11}$$

Several different options exist to consider dev within the help calculation. In $\mathcal{S}$'s implementation dev is used to dynamically determine the threshold for $\mathrm{num_{interaction}}$ $\mathcal{U}$ may differ from $\mathrm{avg}_{\mathrm{num_{interaction}}}$. The less the value of $progress$ is the less the threshold is. At the beginning when $\mathcal{U}$ enters $\mathcal{P}$, $\mathcal{S}$ reacts more strict to anormative behaviour and therefore offers help much faster than in phases of advanced progress.

To configure the maximum deviation allowed in case of $progress = 0\%$ and $progress = 100\%$, min_{dev} and max_{dev} are introduced. max_{dev} specifies the maximum percentage $\mathcal{U}$'s behaviour may deviate compared to the average in case of $progress = 100\%$, min_{dev} the maximum percentage $\mathcal{U}$'s behaviour may deviate at the beginning in case of $progress = 0\%$, whereas $0 \leq min_{dev} \leq max_{dev}$. Equ.12 shows the calculation of the general threshold factor. Fig.3 shows the corresponding graph for $min_{dev} = 35\%$ and $max_{dev} = 100\%$. Equ.13 shows the actual threshold value against $\mathcal{U}$ and $progress$.

$$\mathrm{thr_{factor}}(progress) = (min_{dev} + (max_{dev} - min_{dev})(1 - \mathrm{dev}(progress))) \tag{12}$$

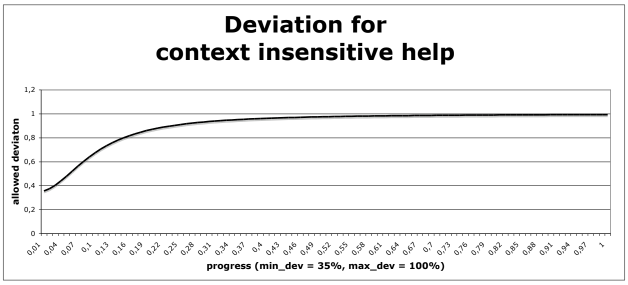


Figure 3: $min_{dev} = 35\%, max_{dev} = 100\%$

$$\mathrm{thr_{value}}(\mathcal{U}, progress) = \mathrm{thr_{factor}}(progress) \cdot \mathrm{num_{interaction}}(\mathcal{U}, progress) \tag{13}$$

4.3 Model Application

Due to the genericness of $\mathcal{S}$ mappings between the input elements and the corresponding help texts have to defined. If $\mathcal{S}$ determines that $\mathcal{U}$ probably needs help in filling in ε_{sel}, $\mathcal{S}$ has to look up the mapping record for ε_{sel} to get the corresponding help text. This is afterwards sent to the client, which reacts with dynamically showing an unobtrusive question mark next to ε. If $\mathcal{U}$ clicks on it, the browser shows the received help text.

The content of the context insensitive help is by default the same for all $\mathcal{P}$. If $\mathcal{S}$ determines that $\mathcal{U}$ maybe needs this general kind of help, it simply sends the hint to the client to show a link to the static help pages. This link could be shown with absolute positioning so that it is e.g. always shown in the right upper corner independent of the window's current scroll state. All the user notifications could furthermore be combined with decent audio jingles if $\mathcal{U}$ shows a preference for that.

5 Future Work and Conclusions

This paper presented certain parts of the results in context of the author's ongoing PhD thesis. First, the author gave a brief introduction to the concept of AAHS. Afterwards, a preliminary version of a generic module was designed which aims to offer both context sensitive and insensitive help by using out-of-band techniques.

Currently, the validation of the shown concepts are prepared. This concerns AAHS in general as well as the presented help system in particular. The evaluation study consists of two independent parts. In a technical evaluation the general feasibility and certain aspects like scalability, latencies and browser-behaviour are investigated and evaluated in order to build AAHS upon a stable and reliable technical basis. The number of simultaneous network connections as well as timings can be modelled quite well "offline" and therefore determined in advance without experiments.

The user-oriented evaluation shows if continuously updated user models in combination with the proposed technique of "instant adaptation" have further impact on the overall quality of AHS. In particular, the presented algorithms for determining $\mathcal{U}$'s locus of attention and the probability of needing help are tested in empirical user studies.

The upcoming field of AAHS aims to bridge the gap between adaptive desktop and hypermedia applications. Therefore, it has to be investigated how traditional and well-established adaptive techniques from the desktop can be applied to web applications by using out-of-band techniques. Due to the won connection directly to the user's desktop the investigation of interpretation of subsymbolic user behaviour offers new interesting challenges. Many new features, challenges and research topics are expected – thus, it's high time, let's launch Adaptive Hypermedia 2.0!

6 Acknowledgements

The work reported in this paper has been partially funded by the Socrates Minerva Adaptive Learning Spaces (ALS) project (229714-CP-1-2006-1-NL-MPP). For information on the ALS project please refer to the projects web site: `http://www.als-project.org`.

References

[Alinone, 2005] Alessandro Alinone. Changing the web paradigm - moving from traditional web applications to streaming-ajax. *published online.*, 2005.

[Allaire, 2002] Jeremy Allaire. Macromedia flash mx - a next generation rich client. *published online.*, 2002.

[Atterer *et al.*, 2006] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 203–212, New York, NY, USA, 2006. ACM Press.

[Barla, 2006] Michal Barla. Interception of user's interests on the web. In *Proceedings of the International Conference on Adaptive Hypermedia (AH)*, volume 4018 of *Lecture Notes in Computer Science*, pages 435–439. Springer, 6 2006.

[Boddu *et al.*, 2007] Raja Sarat Kumar Boddu, Surendra Prasad Babu Maddali, and Siva Prakasa Rao Alti. Ajax interaction in adaptive hypermedia. In *A3H: Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia, UM 2007, 11th International Conference on User Modeling, Corfu, Greece*, 2007.

[Brusilovsky, 1996] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[Brusilovsky, 2001] Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110, 2001.

[Carberry, 2001] Sandra Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.

[de Bra *et al.*, 2002] Paul de Bra, Ad Aerts, David Smits, and Natalia Stash. Aha! meets aham, 2002.

[Di Paola, 2006] Stefano Di Paola. Subverting ajax. In *Proceedings of 23rd CCC Conference, Berlin*, 2006.

[Farzan and Brusilovsky, 2005] R. Farzan and P. Brusilovsky. Social navigation support in e-learning: What are the real footprints? In *3rd WS on Intell. Techniques for Web Personalization (ITWP '05). 19th IJC on AI*, 2005.

[Garrett, 2005] Jesse James Garrett. Ajax: A new approach to web applications. *published online.*, 02 2005.

[Hofmann *et al.*, 2006] Katja Hofmann, Catherine Reed, and Hilary Holz. Unobtrusive data collection for web-based social navigation. In *Workshop on the Social Navigation and Community based Adaptation Technologies*, 2006.

[Horvitz *et al.*, 2003] Eric Horvitz, Carl Kadie, Tim Paek, and David Hovel. Models of attention in computing and communication: from principles to applications. *Commun. ACM*, 46(3):52–59, 2003.

[Kristina *et al.*, 1996] H. Kristina, J. Karlgren, A. Waern, N. Dahlback, C. Jansson, K. Karlgren, and B. Lemaire. A glass box approach to adaptive hypermedia, 1996.

[O'Reilly, 2005] Tim O'Reilly. What is web 2.0? *published online.*, 09 2005.

[Owen, 2006] Robert S. Owen. *Tracking Attention through Browser Mouse Tracking*, pages 615–621. Idea Group Reference, Hersey, London, Melbourne, Singapore, 2006. ISBN: 1-59140-562-9 (hardcover) / 1-59140-798-2 (ebook).

[Putzinger, 2007] Andreas Putzinger. Upcoming privacy issues in asynchronous adaptive hypermedia. In Christoph Hoyer, editor, *Proceedings of IDIMT 2007 (forthcoming)*, 2007.

[Raskin, 2000] Jef Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison Wesley, 2000.

[Roda and Thomas, 2005] C. Roda and J. Thomas. *Encyclopaedia of HCI*, chapter Attention Aware Systems, pages 38–44. IDEA Group, 2005.

[Sonntag, 2006] Michael Sonntag. Ajax security in groupware. In *32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO'06)*, 2006.

[Stiefelhagen *et al.*, 2001] R. Stiefelhagen, J. Yang, and A. Waibel. Estimating focus of attention based on gaze and sound, 2001.

Adaptive Reading Assistance for Dyslexic Students: Closing the Loop

Andreas Schmidt and **Michael Schneider**
FZI Research Center for Information Technologies
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, GERMANY
{andreas.schmidt | michael.schneider}@fzi.de

Abstract

Adaptive reading assistance can improve the reading performance of students, but current dyslexia pedagogical theories do not yet provide sound results on a micro-level. We want to provide a reading assistance solution that both helps the learner and the dyslexia researcher. In order to archive this, we encode adaptation knowledge in a descriptive way by making use of state-of-the-art ontology-based techniques. This enables a closed-loop approach of continuous improvement. In this paper, we want to present the overall approach as well as initial results of our work within the EU project AGENT-DYSL.

1 Introduction

Dyslexia affects a significant number of students (it is estimated that one in ten children is dyslexic) and leads to a considerably slower development of readings skills. As reading skills are key for success at school and later in the job, dyslexic people are turned into low achievers in education and learning, excluding them from several aspects of social living. Dyslexia, however, does not mean that these students do not learn to read well at all, but rather it takes them significantly longer and requires more practice than for non-dyslexic learners.

The most promising approach to help dyslexic learners is to assist their reading skills development while they remain within their normal peer group. As direct teacher support is limited, this needs to be complemented by eLearning solutions. Especially for dyslexic students, such a solution needs to be sensitive to the learner's capabilities and current emotional state. This demands for a highly adaptive reading assistance system (in contrast to current software products for that purpose).

However, research on dyslexia is in many respects not yet able to provide the knowledge for adaptation rules, e.g., how to intervene into the reading process of a dyslexic child with a certain error profile. Thus the adaptation rules themselves will have to be subject of research; we need a closed-loop approach as illustrated in fig. 1. In a first step, adaptation knowledge is encoded based on state-of-the-art dyslexia theories. The effect of applying adaptation knowledge in providing learning assistance to students is then assessed in a subsequent step by dyslexia experts. Based on that assessment, dyslexia theories are modified, which in turn form the basis for an improvement for the learning assistance system. For such evaluation loops, we need to represent the adaptive behavior in a way that is easy to understand and changeable by non-technical dyslexia experts.

Within the EU project AGENT-DYSL[1] [Athanasaki *et al.*, 2007] we have taken an approach that is based on state of the art semantic technologies to represent such adaptation knowledge in a descriptive way, which shall be presented in the remaining part of the paper. In section 2, we give a brief overview of related work and the AGENT-DYSL project as a whole, before we present the adaptation approach in more detail in section 3. Finally, we conclude the paper in section 4.

2 General Overview

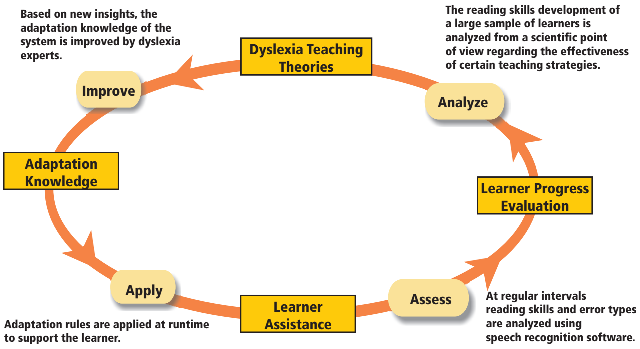
2.1 Related Work

Regarding reading assistance, there are relatively few software application that are specifically targeted towards dyslexia and its specific problems. As a consequence, state of the art commercial software applications like Kurzweil 3000[2] or ReadOn[3] hardly have any form of user adaptivity beyond simple preferences. Recent research approaches like [Amiri, 2006] concentrate particularly on dyslexia and makes use of speech recognition and eye tracking to adapt to readers' progress.

However, all of these approaches do not allow for deep adaptation by, e.g., considering user-specific error types. That way, pedagogical intervention is very limited.

2.2 The project

AGENT-DYSL wants to go beyond that. It aims at providing a truly user adaptive reading assistance system for dyslexic students which allows them for reading arbitrary text documents (e.g., text books). The system will present



Based on new insights, the adaptation knowledge of the system is improved by dyslexia experts.

The reading skills development of a large sample of learners is analyzed from a scientific point of view regarding the effectiveness of certain teaching strategies.

At regular intervals reading skills and error types are analyzed using speech recognition software.

Adaptation rules are applied at runtime to support the learner.

Figure 1: Closed Loop Approach

[1] http://www.agentdysl.eu

[2] http://www.kurzweiledu.com

[3] http://www.readonsoftware.com

the text in an augmented way by using techniques such as text highlighting, segmenting words into its syllables, emphasis on certain characters, or preemptively reading words aloud by the usage of text-to-speech techniques. The key innovative feature of AGENT-DYSL is the adaptivity of the presentation to the individual student and her context such as estimated subjective difficulty of a word, or the current mental and emotional state (e.g., if she is more or less concentrated, or if she gets tired). The system can also suggest individual learning resources for further training, which is useful for a teacher of a dyslexic child.

AGENT-DYSL is a three-year project funded by the European Commission under the eInclusion call and is carried out by a consortium of nine partners from different European countries, consisting of dyslexia researchers, educational experts and teachers, and several technological partners. The evaluation of AGENT-DYSL will take place in three testbeds for the languages English, Greek and Danish. The system will be evaluated directly in classrooms, executed by the different evaluation partners in the project's consortium, and in cooperation with schools and their teachers from England, Greece and Denmark.

2.3 Project Vision

The AGENT-DYSL approach can be divided into three conceptual parts:

- **User Context Acquisition & Management.** Basis for the adaptive system behavior is a thorough acquisition and management of the student's context. This includes individual preferences, but also individual error profiles. In order to detect these typical errors automatically, the student can read the text aloud, and speech recognition will be used to identify reading problems in the text. Also the current mental state will be detected by using image recognition to track and evaluate the face of the child with a simple webcam. At least, one will be able to detect reduced awareness of the child, which can then be used to temporarily pause the text presentation and the word recognition process.
- **Adaptive Annotation of Text.** Based on this context information, upcoming sentences are analyzed with respect to words likely to cause problems for the student. For these words, appropriate changes to the presentation are determined, taking also into account, e.g., the current emotional state of the student.
- **Presentation.** Finally, the text is presented augmented with specific highlighting, word segmentation, etc.

The key challenge in a closed loop approach is now to encode the knowledge that is used for implementing the system behavior in a descriptive way. In the following section, we will present our ontology-based approach to meet this challenge.

3 Descriptive Adaptation Knowledge

3.1 Overview of the Approach

Key idea of the adaptive core of the AGENT-DYSL system is to enable *deep* adaptation to the reader's individual characteristics. This requires (a) a much higher degree of pedagogical knowledge encoded into the system and (b) a wider range of contextual features that allow for a reasonable application of this pedagogical knowledge to provide reading assistance. More specifically (see fig. 2), the adaptation component of the AGENT-DYSL system needs to detect parts of the text (usually words, but also phrases or even sentences) likely to cause problems for the reader. This depends on a classification of error types (like semantic and morphological, visual, or phonological errors) and the words or phrases they typically occur in and an error profile of the learner, indicating the frequency of errors. As soon as we have problematic words, we can apply teaching strategies to adapt the presentation. This includes slowing down reading speed, emphasis on specific letters, or preemptive reading aloud of the word, e.g., when the reader is getting frustrated. Already this example shows that the appropriateness of a teaching strategy does not only depend on the learner characteristics, but also on the situation. This also includes the concentration level of the learner.

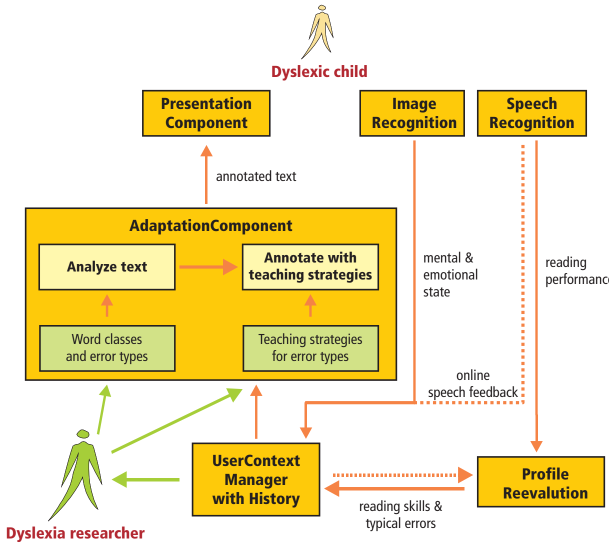


Figure 2: AGENT-DYSL Adaptation Component

3.2 Role of the ontologies

In order to realize the sketched behavior, we use an ontology-based representation for user contexts and adaptation knowledge. There is an upper-level ontology, specifying the relevant concepts and the relations between them, and all user contexts can be regarded as instance knowledge bases. The ontology itself was derived as the result of a conceptual analysis of the regarded domain together with dyslexia experts in the project. It captures different aspects which are of interest for AGENT-DYSL (see fig. 3).

The ontology contains concepts for a **Learner** together with her **Context**. The context is described by a set of **ContextFeature**s, like the learner's age, or his preferred font size. There are highly dynamic features, like recent reading errors made, or the current mental state of the learner. There is also a special group of subfeatures, called **ProfileFeature**s, which build up the **LearnerProfile**. The profile features describe more stable aspects, like the general reading speed, or the typical reading errors, the learner is expected to produce. These profile features are generally the result of aggregating the dynamic features like *current reading error* into a *typical reading error* and eventually into a *reading level.*

ErrorTypes, which build an abstraction for the concrete reading errors, need to be associated with three concepts:

- **Word**s these ErrorTypes may occur in. In order to keep this manageable, we will usually cre-

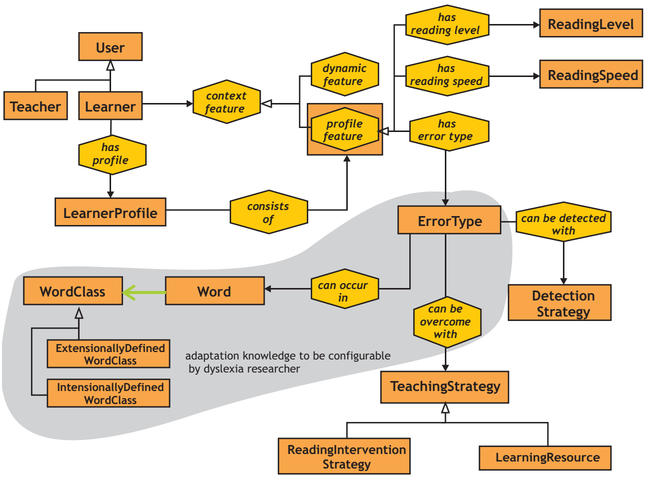


Figure 3: AGENT-DYSL Upper-Level Ontology

ate **WordClass**es. These can be defined extensionally (**ExtensionallyDefinedWorldClass**) by simply listing individual words, or intensionally (**IntensionallyDefinedWorldClass**) by defining a pattern at individual letter, on syllable or other level). Examples for such patterns could be *words containing more than 5 syllables*, word with certain letter combinations or the like.

- **TeachingStrategy** is a pedagogical measure that tries to help the learner to overcome this ErrorType. We can divide these strategies into **ReadingIntervention-Strategy** (i.e., an intervention into the reading process of the learner). Examples for such interventions are different forms of highlighting (color, font size), word segmentation or preemptive reading of difficult words. The other type of strategy is to recommend a **LearningResource** (Exercises, other texts, etc.).
- **DetectionStrategies** are important for deriving ErrorTypes from speech recognition: It answers the question on how we can distinguish one error type from another when the learner mispronounces a word.

3.3 Involving the Dyslexia Expert

As we want to build a system in which a dyslexia expert is able to configure the system behavior, we need to specify which parts of the ontology are expected to be changed by the dyslexia experts and which parts need changes by the software developers. It is clear that DetectionStrategies and TeachingStrategies as well as the acquisition of ContextFeatures needs to be predefined as we need complex software components to apply them. But the dyslexia expert can provide input on two aspects: (1) classification of Words and the association with ErrorTypes on the one hand, and on the other hand (2) the association of ErrorTypes with TeachingStrategies, which has to be contextually dependent.

For the first case, there is currently no classification of words and/or error types we can make use of. We are currently starting with a very basic approach with a few error types and extensionally defined word classes. This has its clear limitations especially with respect to unknown texts. But what we expect to gain also from error statistics of the students is a more general knowledge about intensionally defined word classes (by using patterns). We are currently exploring different alternatives for pattern languages that are easy to understand and still expressive enough.

For the second case, AGENT-DYSL will use a rule-based mechanism with sets of rules of the form:

> "If the word/phrase/sentence W is contained in one of the child's typical errors E, and the child is currently in emotional state S, then choose the following teaching strategy T"

With this approach, we do not encode the way AGENT-DYSL operates on the user context in a procedural programming language, but instead encode it in terms of more descriptive *rule sets*. By making these rule sets accessible and modifiable from outside the system, dyslexia experts will become able to experiment with the way AGENT-DYSL works.

The dyslexia expert is expected to adjust these rules easily via a specific GUI, by which he is able to build new rules, delete old ones, and modify existing ones. She can do this for each rule by selecting word classes, error types and contextual features together with a set of teaching strategies. She can then let proband children use this modified version of AGENT-DYSL, and see whether the modified system has better effects on reading performance of students (using log data information as well as standardized reading performance assessments). If the system's and the child's reaction do not meet the expert's expectation, she can re-adjust the rules step by step to improve the system.

4 Conclusions and Outlook

We have presented our approach of a closed-loop approach to an adaptive educational system. Ontology-based techniques as employed in AGENT-DYSL allow for a descriptive representation of the adaptation knowledge. We are currently in the process of implementing the system and evaluating it with dyslexia experts.

Future work will investigate into more complex error types that are not triggered by individual words, but rather by a specific phrase or semantic context. This will also lead to considering different scopes for applying teaching strategies (like word, phrase, sentence).

References

[Amiri, 2006] Huddia Amiri. *Reading Assistance Program for People with Dyslexia*. PhD thesis, Clayton School of Information Technology, Monash University, 2006.

[Athanasaki *et al.*, 2007] Maria Athanasaki, Maria Avramouli, Kostas Karpouzis, Stefanos Kollias, Klimis Ntalianis, Andreas Schmidt, Antonis Symvonis, and Francesc Valcarcel. Agent-dysl: A novel intelligent reading system for dyslexic learners. In Miriam Cunningham Paul Cunningham, editor, *eChallenges 2007*, 2007.