Friedrich Summann / Norbert Lossau

Suchmaschinentechnologie und Digitale Bibliotheken: Von der Theorie zur Praxis[1]

Der folgende Aufsatz beschreibt aus technischer Sicht den Weg von der Konzeption und Vision einer modernen suchmaschinenbasierten Suchumgebung zu ihrer technologischen Umsetzung. Er nimmt den Faden, der im ersten Teil (ZfBB 51 (2004), 5/6) beschrieben wurde, unter technischen Gesichtspunkten wieder auf. Dabei werden neben den konzeptionellen Ausgangsüberlegungen schwerpunktmäßig die technologischen Aspekte beleuchtet.

This article describes the development of a modern search engine-based information retrieval setting from its envisioning through to its technological realization. The author takes up the thread of an earlier article on this subject (ZfBB 51 (2004), 5/6), this time from a technical viewpoint. After presenting the conceptual considerations of the initial stages, this article deals principally with the technological aspects of the project

Die Konzeption einer wissenschaftlichen Suchmaschine

Ausgangspunkt für die Überlegungen waren die Erfahrungen des durchaus erfolgreichen Projektes »Digitale Bibliothek NRW«, mit dem in den Jahren 1998 bis 2001 federführend von der Universitätsbibliothek Bielefeld der Systementwurf eines internetbasierten Bibliotheksportals mit verbesserter wissenschaftlicher Suchumgebung umgesetzt worden war. Im Zentrum dieses Systems steht eine Metasuche mit Verfügbarkeitsfunktion, der eine Benutzeroberfläche zur Integration aller für Studium und Forschung relevanten Arbeitsquellen zur Seite gestellt wurde. Die Defizite dieses Ansatzes waren nach der Produktionsaufnahme im Juni 2001 deutlich erkennbar. Probleme gab es mit der Stabilität und Performance der Datenbankzielsysteme, mit der Integration von Volltext-Dokumenten und Internet-Seiten und mit der generellen Akzeptanz, da die Benutzer zunehmend auf Suchmaschinen und weniger auf Bibliothekssysteme zugreifen. Da aber kommerzielle Suchmaschinen wiederum unter wissenschaftlichen Gesichtspunkten eine Reihe von Problemen aufweisen (insbesondere das Auffinden wissenschaftlicher Informationen und die Langzeitverfügbarkeit), entstand die Idee einer Suchmaschine für wissenschaftliche Nutzung. Gleichzeitig war damit die Hoffnung verbunden, die heterogene Landschaft der wissenschaftlichen Informationsversorgung mit Fachdatenbanken, Katalogdatenbanken, elektronischen Zeitschriften, Dokumentservern und wissenschaftlichen Web-Seiten umfassend auf der Basis einer Suchmaschinentechnologie anzugehen und mit einem einheitlichen Zugang zu lösen.

Softwareevaluierung und technische Umsetzung

Ausgehend von diesen grundsätzlichen Überlegungen wurde der Markt nach geeigneten Softwareprodukten durchsucht. Die Bemühungen, mit Google ins Gespräch zu kommen, scheiterten in einem frühen Stadium, da nur Marketingleute als Ansprechpartner zur Verfügung standen. So ließ sich zumindest im Jahre 2002 keine Möglichkeit erkennen, Google-Software vor Ort einzusetzen. Anders hingegen war die Situation bei der Suchmaschine Convera, die kurzfristig in Bielefeld installiert und erprobt werden konnte. Ergebnis der intensiven zweiwöchigen Tests war, dass die Software eher für einen Intraneteinsatz geeignet schien, für den eigenen Ansatz als Internet-Suchmaschine aber Defizite aufwies. Ein Test der russischen Open-Source-Suchmaschine MnoGo zeigte durchaus positive Ergebnisse, allerdings traten auch Performance-Probleme bei großen Datenmengen auf. Ein ausgesprochen guter Kontakt ergab sich mit der norwegischen Software-Firma Fast Search & Transfer[2], die im Jahre 2002 mit der Suchmaschine Alltheweb neben Google zu den Marktführern zählte. Eine Testinstallation wurde kurzfristig vereinbart, auch die technische Umsetzung erfolgte zügig und problemlos. Nach der positiv verlaufenen Testphase stand fest, mit dieser Suchmaschine die Realisierbarkeit des vorgelegten Konzeptes *(proof of concept)* zu prüfen. »Konzept« bedeutete in diesem Zusammenhang die Zusammenstellung und Bereitstellung einer repräsentativen und heterogenen Menge von wissenschaftlichen Online-Inhalten. Dabei sollten unterschiedliche Dokumenttypen, unterschiedliche Dokumentformate, sowohl Volltext als auch Metadaten und Inhalte des »visible« und »invisible« Web erfasst werden. Grundvoraussetzung war, dass dieser Test unter Produktionsbedingungen auf der Basis von Fast Data Search erfolgen sollte. In diesem Kontext sollte mit Interoperabilitätsstandards (OAI, XML) gearbeitet und Prototypen einer intelligenten und flexiblen Benutzeroberfläche geschaffen werden.

Die technische Umsetzung in der Universitätsbibliothek Bielefeld begann im Sommer 2003. Das Team bestand aus zwei Softwareentwicklern, die den Prototypen auf der Basis der FAST Data Search Software erstellten. Diese Aktivitäten wurden als Vorarbeiten in den nationalen Projektantrag »Suchmaschinentechnologie« der Universitätsbibliothek Bielefeld und des

Foto privat

Friedrich Summann

Foto Neue Westfälische

Norbert Lossau

Test-Basis: wissenschaftliche Online-Inhalte

Bielefeld Academic Search Engine (BASE)

Hochschulbibliothekszentrums Köln als Teil der Initiative der AG Verbundsysteme zum »Verteilten Dokumentenserver (VDS)« eingebracht. Konkret begonnen wurde mit der Realisierung des »Math Demonstrators«, der mit der Konzentration auf ein bestimmtes Fach die Entwicklungs- und Diskussionsgrundlage bilden sollte und an dem bis zum Frühjahr 2004 gearbeitet wurde. Diese Konzeptphase ist mittlerweile abgeschlossen, und mit der Bielefeld Academic Search Engine (BASE, http://base.ub.uni-bielefeld.de/) wurde ein öffentlicher »Digital Collections Demonstrator« freigeschaltet.

Technische Details der Suchmaschinenlösung

Die technische Struktur der FAST-Suchmaschine ist modular und transparent aufgebaut und enthält die selbstständigen Systemkomponenten Backend- und Frontend-Server. Zurzeit wird in Bielefeld *ein* Frontend-Server betrieben, die Zahl kann leicht auf mehrere erhöht werden. Ebenso ist die Zahl der Backend-Server skalierbar, und beide Bereiche können entsprechend problemlos zu einem Multi-Node-System ausgebaut werden.

Das Frontend übernimmt die Aufgaben Suchumgebung, Ergebnisanalyse und Ergebnispräsentation und läuft zurzeit auf einem Linux-PC mit zwei Prozessoren unter SUSE 9.0. Die Web-Anbindung erfolgt mit PHP 4 auf einem Apache Web Server. Das Backend bearbeitet die Aufgabenbereiche Datensammlung, Pre-Processing und Datenkonversion, Dateiverarbeitung, Crawling sowie Dokumentbearbeitung und Indexierung.

Die Benutzeroberfläche ist zweisprachig (deutsch und englisch). Neben der Basis-Suchmaske mit Google-ähnlicher einzeiliger Suchzeile steht eine Erweiterte Suche (s. Abb. 1) mit zusätzlicher Funktionalität zur Verfügung, auf der bei der Softwareentwicklung der Schwerpunkt liegt. Hier werden die ergänzenden Funktionen wie differenzierte Suchaspekte, Quellenauswahl und Suchhistorie angeboten. Bei beiden Suchmasken lässt sich die Suche auf freie Dokumente eingrenzen.

Die Ergebnisanzeige (s. Abb. 2) unterscheidet sich vom Suchmaschinenstandard durch eine differenzierte Anzeige von Metadaten, wenn solche im Dokument vorhanden sind. Hier bieten sich auch Möglichkeiten zur Suchverfeinerung, z. B. auf Metadatenebene nach Autoren und Klassifikation und nach formalen Aspekten wie Dokumentformat und Quelle. Dabei werden aus der Ergebnismenge die enthaltenen Felder zu einem Auswahlmenü zusammengestellt. Eine Erweiterung der Suche in Bezug auf das einzelne Dokument ermöglicht es, nach ähnlichen Dokumenten im Gesamtindex (Mehr zum Thema) und in der Treffermenge (Auf Thema einschränken) zu suchen oder gerade die ähnlichen in der Treffermenge auszuschließen (Thema ausschließen). Die Anzeige der Suchhistorie vervollständigt das Angebot der Benutzeroberfläche.

Für das Backend-System existieren ein Live-System auf einem Linux-PC unter Suse Linux 9.0 mit zwei Prozessoren und einem RAID-System mit 290 GB Festplattenkapazität und parallel ein Testsystem, ebenfalls auf Linux-PC-Basis, das interne Entwicklungsänderungen zulässt, die keine Auswirkung auf die Verfügbarkeit des Gesamtsystems haben.

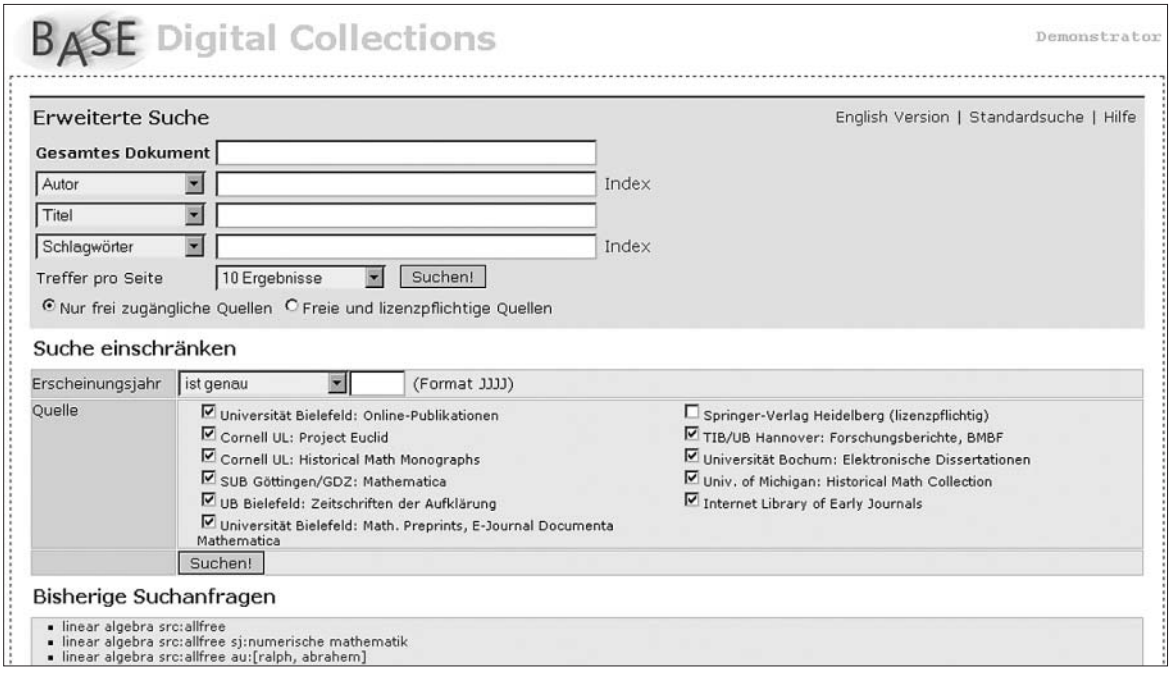

Abbildung 1:
Erweiterte Suchmaske

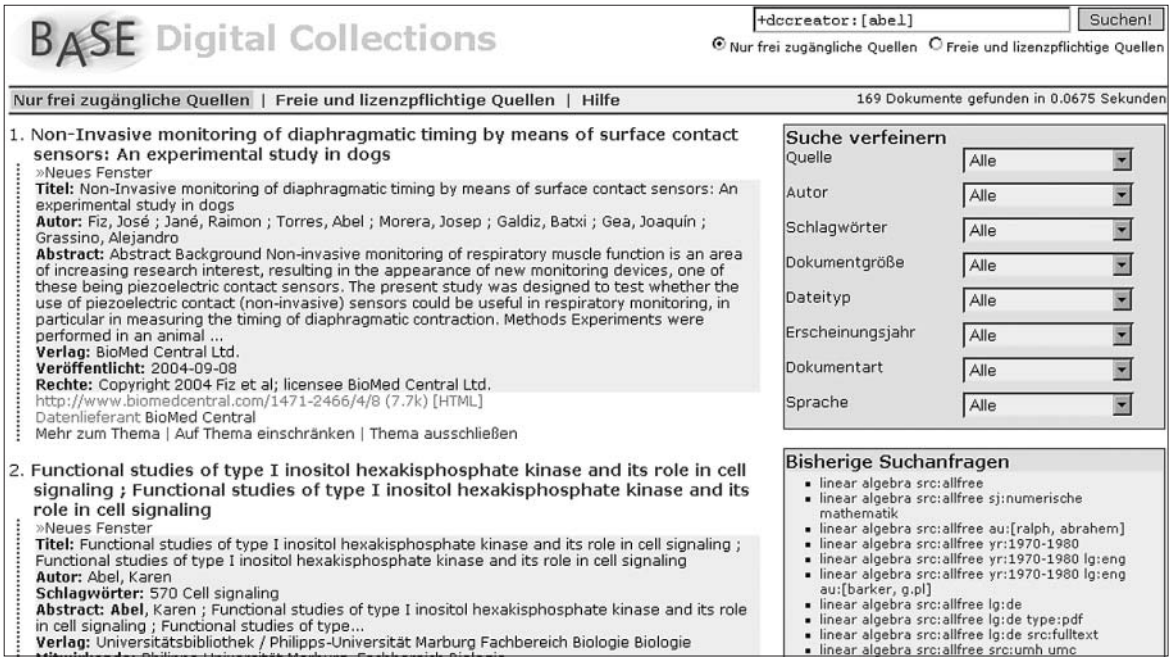

Abbildung 2: Ergebnisanzeige

Die Inhalte

Zurzeit (Stand Juni 2004) sind rund 600.000 Dokumente erfasst und in 15 »Kollektionen« aufgeteilt. Der Backend-Server benötigt dafür einen Speicherplatz von rund 25 GB. Die Auswahl der erfassten Quellen wurde unter dem Aspekt getroffen, exemplarische Daten von verschiedenem Typus zu erfassen. Dazu wurden bearbeitet

Metadaten

Projekt Euclid (OAI) (6.516 Artikel)
Bielefelder Bibliothekskatalog (Datenbankabzug) (70.000 Titel)
Zentralblatt für Mathematik (Datenbankabzug)
Artikel des Projektes »Zeitschriften der Aufklärung« (Datenbankabzug) (57.690 Artikel)

Volltext ohne Metadaten

Documenta Mathematica (e-journal)
Preprint Server an der Universität Bielefeld (Crawling) (18.000 Dokumente)
Projektberichte des Bundesministeriums für Bildung und Forschung (Crawling) (64 Dokumente)

Volltext mit Metadaten

Springer Zeitschriften (224.382 Artikel)
Hochschulschriftenserver Universitätsbibliothek Bochum (OAI-Harvesting) (1.908 Dokumente)
Hochschulschriftenserver Universitätsbibliothek Bielefeld (OAI Harvesting) (369 Dokumente)
Internet Library of Early Journals, Host: Oxford University Library Services (SGML Export) (104.516 Seiten)
University of Michigan Historical Math Collection (OAI Harvesting) (772 Dokumente)
Cornell University Library Historical Math Monographs (OAI Harvesting) (630 Dokumente)
Mathematica der Staats- und Universitätsbibliothek Göttingen (OAI Harvesting) (427 Dokumente)

Die Integration von Datenquellen

Im Umfeld der Metasuche aus der »Digitalen Bibliothek NRW«, die zurzeit einen der Basisdienste des Bielefelder Bibliotheksportals bildet, besteht die Arbeit zur Integration von weiteren Ressourcen darin, Verbindungsinformationen über das Zielsystem zu ermitteln (Z39.50 oder http-basiert), darauf aufbauend die Datenbankanfragen zu implementieren und die gelieferten Ergebnisdaten in ein internes Format umzuwandeln.

Datenerfassung über drei Schnittstellen

Bei der Suchmaschinenanwendung ist ein völlig anderes Vorgehen notwendig. Um Daten in den Index gelangen zu lassen, sind verschiedene Wege möglich. FAST bietet für die Erfassung der Daten die drei Schnittstellen Web Crawler, Datenbank Connector (für den Zugriff auf relationale Datenbanken wie Oracle etc.) und File Traverser (Zugriff auf Dateien) an. Auf den Datenbank Connector konnte im Rahmen von BASE bisher verzichtet werden, da keine diesbezüglichen Daten einbezogen werden mussten. Eine Übersicht des Datenflusses mit den wesentlichen Schritten zeigt Abbildung 3.

Um Nicht-HTML-Dokumente zu bearbeiten, wird mit der FAST-Schnittstelle File Traverser gearbeitet, wo-

Entwicklungsschwerpunkt: Harvesting von OAI-Daten

bei die vorliegenden proprietären Formate per Pre-Processing in ein FAST-definiertes XML-Format umgesetzt werden. Ein wesentlicher Schwerpunkt der Entwicklung war das Harvesting von OAI-Daten, wobei mit Hilfe eines Open-Source-Produktes der Virginia Polytechnic Institute and State University diese Daten geholt und abgelegt werden. Bei der Analyse der OAI-Daten zeigte sich allerdings in der Praxis eine Reihe von Problemen, für die flexible und konfigurierbare Lösungen benötigt werden. Das verwendete Dublin-Core-Format lieferte im Datumsfeld oder beim Language Code formal sehr heterogene Lösungen, die individuell behandelt werden mussten. Die zum Volltext führende URL wurde teilweise im Source-Feld abgelegt, und diese Konstellationen mussten berücksichtigt werden. Nach den bisher gemachten Erfahrungen streben wir in diesem Umfeld weitergehende Softwarelösungen an, die die bisher erstellten Perl- und XSLT-Skripte um universelle Konfigurationsfunktionen erweitern. Insgesamt werden mit diesen Anwendungen im Bereich des Pre-Processing die folgenden Aufgaben erledigt: Language-Code-Erkennung, Datumsfilterung, XML-Konversion, Erzeugung eines eindeutigen Identifiers, generelle Filterung und Fehlerbehebung, Erzeugung der Kategorieninhalte, Ermittlung des Volltextlinks.

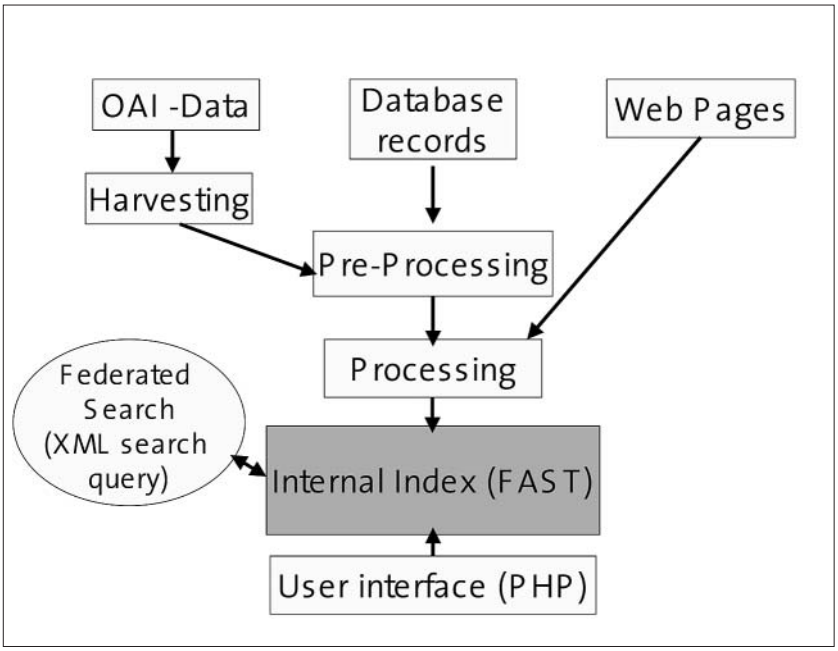

Abbildung 3: Datenfluss BASE

	FAST	Additional developments
Data Loading	Crawler, File Traverser, DB Connector	OAI Harvester DB export
Pre-processing		Perl, XSLT crosswalks
Processing	Standard stages	Python stages
Indexing	Indexer	
Access, Navigation	Search API	PHP scripts

Abbildung 4: Übersicht verwendeter Tools

Im Bereich des Processing dagegen werden im internen Datenimportprozess interne und lokale Filterprogramme definiert, durch die zu bearbeitende Daten verändert werden können. Hierzu gehören Aufgaben bei der Nutzung des File Traversers wie Spracherkennung, Mime-Type-Erkennung, Teasergenerierung und Feldzuordnung. Beim Crawling werden Aufgaben wie Formaterkennung, Dekomprimierung, Setzen des internen Inhaltstyps (Volltext, Metadaten, Mischformen), Formatkonversion von Postscript oder PDF und Sprachfestlegung im Rahmen des Processing bearbeitet.

Eine besondere Aufgabe besteht in der Zusammenführung von Metadaten und Volltextinformation zu einem gemeinsamen Datensatz, der für Suche und Ergebnisanzeige als Einheit behandelt werden kann.

Die für den Erfassungsprozess genutzten FAST-Module und die zusätzlich entwickelten Module und Tools listet Abbildung 4 auf. Diese Aufstellung zeigt zudem, dass die Eigenentwicklungen sämtlich auf der Basis von Open-Source-Lösungen entstanden sind.

Im FAST-System ist eine für BASE definierte Indexstruktur angelegt worden, die im Kern die 15 Dublin-Core-Felder enthält. Darüber hinaus sind zurzeit fünf Zusatzfelder definiert worden, die für die Inhalte ISBN/ISSN, Digital Object Identifier (DOI), Jahr (normierte Fassung), Sourcetype (Metadaten, Volltext usw.) und Quelle vorgesehen sind.

Weitere Entwicklungen und Visionen

Für die weiteren Entwicklungen im Frontend stehen zahlreiche Punkte an. Mit der Einführung von Templating-Technik soll die flexible Integration der Suchmaschine in externe Umgebungen unterstützt werden, um damit lokale Views auf die Suchmaschine mit Festlegung von Such- und Ergebnisparametern zu ermöglichen. Schon jetzt ist es möglich, in beliebige Web-Seiten eine oder mehrere Suchzeilen einzufügen, mit der die BASE-Suchmaschine in eine externe Portalumgebung eingebunden werden kann. Eine Weiter-

entwicklung in diesem Bereich würde dann auch eine Differenzierung der Ergebnisanzeige unterstützen. Weiter soll das Suchinterface auf Basis der Such-API erweitert werden und zudem bei der Ergebnisanzeige eine Kombination aus Metadatenanzeige und zugehörigem Volltext realisiert werden.

Im Bereich des Backend wird ein Schwerpunkt auf der Automatisierung und Konfigurierbarkeit der Harvesting- und Pre-Processing-Abläufe der Dokumente liegen, um insbesondere die oben dargestellten Probleme im Bereich des OAI-Harvesting in den Griff zu bekommen. Der Bereich Verbesserung der Suchresultate (Ranking, Boosting, linguistische Methoden) muss unter dem Gesichtspunkt der wissenschaftlichen Nutzung einem »Feintuning« unterzogen werden. Die Verbesserung der Performance ist immer ein wichtiges Thema und muss daher berücksichtigt werden. Da die Suchmaschine auch Basisdienste für externe Systeme und Portale bereitstellen soll, ist die Implementierung von Standard-Schnittstellen (Z39.50, OAI, SOAP) vorgesehen. Für die Zusammenarbeit mit anderen Systemen ist darüber hinaus die Aktivierung der Features Verteilte Suche und Verbindung mit externen Suchindexen geplant.

[1] Der Artikel ist die überarbeitete Fassung einer Präsentation auf dem Frühjahrstreffen 2004 der amerikanischen Digital Library Federation (DLF) in New Orleans (www.diglib.org/forums/Spring2004/springforum04abs.htm) und ist in englischer Übersetzung in D-Lib Magazine, September 2004, erstveröffentlicht worden: Search Engine Technology and Digital Libraries: Moving from theory to practice; www.dlib.org/dlib/september04/lossau/09lossau.html

[2] www.fastsearch.com

Die Verfasser

Friedrich Summann ist Leiter der EDV-Abteilung der Universitätsbibliothek Bielefeld, Universitätsstraße 25, 33615 Bielefeld, summann@ub.uni-bielefeld.de

Dr. Norbert Lossau ist Direktor der Universitätsbibliothek Bielefeld, Universitätsstraße 25, 33615 Bielefeld, lossau@ub.uni-bielefeld.de