

Eine vergleichende Analyse von Enterprise- Suchmaschinen für die Indexierung von Katalogdaten

Masterarbeit

Studiengang Bibliotheks- und Informationswissenschaft

Fakultät für Informations- und Kommunikationswissenschaften

Technische Hochschule Köln

vorgelegt von:

Stefan Niesner

am 03.02.2017 bei Dr. Peter Kostädt und Prof. Dr. Achim Oßwald

Abstract

Mit dem Aufkommen von Internetsuchmaschinen haben sich die Anforderungen der Nutzer an bibliothekarische Suchsysteme wesentlich verändert. Bibliotheken haben darauf mit dem Einsatz von Suchmaschinentechnologie in Online-Katalogen reagiert. Die technische Basis solcher Katalogsuchmaschinen bilden Enterprise-Suchmaschinen. In dieser Arbeit wurde ein Vergleich von Enterprise-Suchmaschinen durchgeführt. Dazu wurden zunächst die Enterprise-Suchmaschinen ermittelt, welche die technische Basis der Eigenentwicklungen unter den Katalogsuchmaschinen bilden. Dies sind die Enterprise-Suchmaschinen CIXbase, Elasticsearch, Solr und Xapian. Anschließend wurden die Funktionen von Elasticsearch, Solr und Xapian anhand eines zuvor erstellten Kriterienkatalogs miteinander verglichen. Festgestellt wurde, dass Elasticsearch und Solr zueinander mehr Gemeinsamkeiten als Unterschiede aufweisen. Xapian wiederum weist mehr Unterschiede als Gemeinsamkeiten zu Elasticsearch und Solr auf.

Schlagwörter:

Elasticsearch, Enterprise-Suchmaschine, Katalogsuchmaschine, Solr, Xapian

The user demands for library search systems have changed substantially with the rise of internet search engines. Libraries have reacted to this by using search engine technology in online catalogs. The technical basis of such catalog search engines are enterprise search engines. In this thesis a comparison of enterprise search engines has been carried out. At first the enterprise search engines being the technical basis of in-house developments have been determined. These are the enterprise search engines CIXbase, Elasticsearch, Solr and Xapian. Subsequently the functions of Elasticsearch, Solr and Xapian have been compared based on a beforehand created list of criteria. The findings are that Elasticsearch and Solr show more similarities than differences. Xapian on the other hand shows more differences than similarities when compared to Elasticsearch and Solr.

Keywords:

catalog search engine, Elasticsearch, enterprise search engine, Solr, Xapian

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Abkürzungsverzeichnis.....	VI
1 Einleitung.....	1
1.1 Ziel der Arbeit	3
1.2 Forschungsfragen.....	3
2 Entwicklung von Online-Katalogen und technische Grundlagen von Enterprise-Suchmaschinen.....	4
2.1 Begriffsbestimmung	4
2.1.1 Discovery System.....	4
2.1.2 Enterprise-Suchmaschine	5
2.1.3 Katalogsuchmaschine	5
2.1.4 Online Public Access Catalog (OPAC)	5
2.2 Entwicklung von Online-Katalogen	5
2.3 Technische Grundlagen von Enterprise-Suchmaschinen.....	10
2.4 Zusammenfassung	12
3 Katalogsuchmaschinen in Bibliotheken.....	14
3.1 Ziel	14
3.2 Methodik.....	14
3.2.1 Ermittlung der relevanten Institutionen	14
3.2.2 Literaturrecherche.....	15
3.2.3 Wahl der Untersuchungsmethode	15
3.3 Durchführung.....	16
3.3.1 Verzeichnung der Online-Kataloge.....	16
3.3.2 Identifizierung des Produktes.....	17
3.3.3 Ermittlung und Verifizierung der Eigenentwicklungen	21
3.4 Ergebnis.....	22
3.4.1 Verbundzentralen.....	23

3.4.2	Öffentliche Bibliotheken in Städten mit mehr als 400.000 Einwohnern	24
3.4.3	Bibliotheken mit überregionaler Bedeutung	24
3.4.4	Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden	25
3.4.5	Technische Basis der Eigenentwicklungen	25
3.5	Vorteile und Nachteile der Eigenentwicklung einer Katalogsuchmaschine	26
3.6	Fazit.....	27
4	<i>Vergleichende Analyse von Enterprise-Suchmaschinen.....</i>	29
4.1	Ziel	29
4.2	Methodik.....	29
4.2.1	Festlegung der zu vergleichenden Enterprise-Suchmaschinen.....	29
4.2.2	Literaturrecherche.....	29
4.2.3	Wahl der Untersuchungsmethode	30
4.3	Durchführung.....	31
4.3.1	Erstellung eines Kriterienkatalogs	31
4.3.2	Beschreibung der Enterprise-Suchmaschinen.....	32
4.3.3	Vergleich der Enterprise-Suchmaschinen.....	33
4.4	Beschreibung des Kriterienkatalogs	33
4.4.1	Dimension „Basisinformationen“	34
4.4.2	Dimension „Systemarchitektur“	34
4.4.3	Dimension „Datenhaltung“	35
4.4.4	Dimension „Indexierung“	35
4.4.5	Dimension „Matching“	36
4.4.6	Dimension „Ergebnispräsentation“	36
4.5	Beschreibung der Enterprise-Suchmaschinen	36
4.5.1	Elasticsearch	36
4.5.2	Solr.....	38
4.5.3	Xapian.....	40
4.6	Vergleich der Enterprise-Suchmaschinen	40
4.6.1	Dimension „Basisinformationen“	41
4.6.2	Dimension „Systemarchitektur“	41
4.6.3	Dimension „Datenhaltung“	42
4.6.4	Dimension „Indexierung“	43
4.6.5	Dimension „Matching“	44
4.6.6	Dimension „Ergebnispräsentation“	44

4.6.7	Gemeinsamkeiten, Unterschiede, Alleinstellungsmerkmale	45
4.7	Installation der Enterprise-Suchmaschinen	47
4.8	Fazit.....	48
5	<i>Fazit und Ausblick.....</i>	50
	<i>Literaturverzeichnis</i>	53
	<i>Dokumentationsverzeichnis</i>	62
Anhang 1:	<i>Institutionen des Untersuchungsfeldes.....</i>	66
Anhang 2:	<i>Zusatzmaterial Kapitel 3 (Abbildungen)</i>	69
Anhang 3:	<i>Liste der Referenzinstallationen.....</i>	72
Anhang 4:	<i>Zusatzmaterial Kapitel 3 (Tabellen).....</i>	73
Anhang 5:	<i>Datenblatt zu Elasticsearch.....</i>	76
Anhang 6:	<i>Datenblatt zu Solr.....</i>	77
Anhang 7:	<i>Datenblatt zu Xapian.....</i>	78
Anhang 8:	<i>Installation der Enterprise-Suchmaschinen (Details).....</i>	79

Abbildungsverzeichnis

<i>Abbildung 1: Kopfteil der Trefferliste im Online-Katalog der UB Duisburg-Essen</i>	<i>19</i>
<i>Abbildung 2: Kopfteil der Trefferliste im Wissensportal der Bibliothek der ETH Zürich</i>	<i>19</i>
<i>Abbildung 3: Kopfteil des Online-Katalogs DISCO der ULB Münster</i>	<i>19</i>
<i>Abbildung 4: Ergebnis für die einzelnen Gruppen der untersuchten Institutionen.....</i>	<i>23</i>
<i>Abbildung A1; Suchschlitz auf der Startseite von EconBiz.....</i>	<i>69</i>
<i>Abbildung A2: Suchschlitz auf der Startseite der VuFind Demoseite.....</i>	<i>69</i>
<i>Abbildung A3: Seitenende einer Trefferliste in EconBiz</i>	<i>69</i>
<i>Abbildung A4: Seitenende einer Trefferliste auf der VuFind Demoseite.....</i>	<i>70</i>
<i>Abbildung A5: Einzeltrefferanzeige in EconBiz</i>	<i>70</i>
<i>Abbildung A6: Einzeltrefferanzeige der VuFind Demoseite.....</i>	<i>71</i>

Tabellenverzeichnis

<i>Tabelle 1: Verwendete Referenzen</i>	21
<i>Tabelle 2: Selbst entwickelte Katalogsuchmaschinen</i>	26
<i>Tabelle 3: Beschreibung der Elemente in der Dimension „Basisinformationen“</i>	34
<i>Tabelle 4: Beschreibung der Elemente in der Dimension „Systemarchitektur“</i>	34
<i>Tabelle 5: Beschreibung der Elemente in der Dimension „Datenhaltung“</i>	35
<i>Tabelle 6: Beschreibung der Elemente in der Dimension „Indexierung“</i>	35
<i>Tabelle 7: Beschreibung der Elemente in der Dimension „Matching“</i>	36
<i>Tabelle 8: Beschreibung der Elemente in der Dimension „Ergebnispräsentation“</i>	36
<i>Tabelle 9: Vergleich der Dimension „Basisinformationen“</i>	41
<i>Tabelle 10: Vergleich der Dimension „Systemarchitektur“</i>	42
<i>Tabelle 11: Vergleich der Dimension „Datenhaltung“</i>	43
<i>Tabelle 12: Vergleich der Dimension „Indexierung“</i>	43
<i>Tabelle 13: Vergleich der Dimension „Matching“</i>	44
<i>Tabelle 14: Vergleich der Dimension „Ergebnispräsentation“</i>	45
<i>Tabelle 15: Vergleich zwischen Elasticsearch und Solr</i>	46
<i>Tabelle 16: Xapian im Vergleich zu Elasticsearch und Solr</i>	46
<i>Tabelle A1: Liste der Verbundzentralen</i>	66
<i>Tabelle A2: Liste der Öffentlichen Bibliotheken in Städten mit mehr als 400.000 Einwohnern</i>	67
<i>Tabelle A3: Liste der Bibliotheken mit überregionaler Bedeutung</i>	67
<i>Tabelle A4: Liste der Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden</i>	68
<i>Tabelle A5: Liste der Referenzinstallationen</i>	72
<i>Tabelle A6: Ergebnis für die Verbundkataloge</i>	73
<i>Tabelle A7: Ergebnis für die ASP-Produkte der Verbundzentralen</i>	73
<i>Tabelle A8: Ergebnis für die Öffentlichen Bibliotheken in Städten mit mehr als 400.000 Einwohnern</i>	74
<i>Tabelle A9: Ergebnis für die Bibliotheken mit überregionaler Bedeutung</i>	74
<i>Tabelle A10: Ergebnis für die Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden</i>	75

Abkürzungsverzeichnis

ASP	Active Service Providing
BASE	Bielefeld Academic Search Engine
CSV	Comma-separated values
DBMS	Datenbankmanagementsystem
dbv	Deutscher Bibliotheksverband
DNB	Deutsche Nationalbibliothek
EDS	EBSCO Discovery Service
ETH	Eidgenössische Technische Hochschule
GBV	Gemeinsamer Bibliotheksverbund
hbz	Hochschulbibliothekszenrum des Landes Nordrhein-Westfalen
HDS	HeBIS Discovery System
HeBIS	Hessisches BibliotheksInformationsSystem
HÖB	Hamburger Öffentliche Bücherhallen
JSON	JavaScript Object Notation
KOBV	Kooperativer Bibliotheksverbund Berlin-Brandenburg
KVK	Karlsruher Virtueller Katalog
OPAC	Online Public Access Catalog
PDF	Portable Document Format
PSI	Pica Search & Index
REST	Representational State Transfer
SLUB	Sächsische Landesbibliothek – Staats- und Universitätsbibliothek Dresden
StB	Stadtbibliothek
SUB	Staats- und Universitätsbibliothek
TH	Technische Hochschule
TIB	Technische Informationsbibliothek
UB	Universitätsbibliothek
ULB	Universitäts- und Landesbibliothek
URL	Uniform Resource Locator
USB	Universitäts- und Stadtbibliothek
VÖBB	Verbund der Öffentlichen Bibliotheken Berlins
XML	Extensible Markup Language
ZBW	Deutsche Zentralbibliothek für Wirtschaftswissenschaften

1 Einleitung

Das Sammeln, das Erschließen und das Vermitteln von Informationen ist eine der Kernkompetenzen von Bibliotheken. Durch den Siegeszug des World Wide Web und der damit einhergehenden Etablierung von Suchmaschinen werden nicht nur diese Kernkompetenzen, sondern auch die Bibliothek selbst immer wieder in Frage gestellt. Bibliotheken entgegen dem mit der Einführung neuer nutzerorientierter Dienstleistungen, wie etwa Bibliothekskatalogen mit Suchmaschinentechnologie. Die wichtigsten Merkmale dieser Bibliothekskataloge sind schnelle Antwortzeiten, die Sortierung nach Relevanz, die Unterbreitung von Suchvorschlägen, der Einsatz linguistischer Verfahren und die facetiierte Suche.¹ Die Daten stammen aus heterogenen Datenquellen und werden innerhalb eines gemeinsamen Index bereitgestellt. Die technische Basis dafür bilden sogenannte Enterprise-Suchmaschinen².

Auf Basis kommerzieller Enterprise-Suchmaschinen wurde Suchmaschinentechnologie von Bibliotheken und Verbundzentralen in Deutschland bereits in den 1990er Jahren für die Indexierung von Katalogdaten eingesetzt.³ Eine größere Verbreitung fand der Einsatz von Suchmaschinentechnologie in Online-Katalogen mit dem Aufkommen von Enterprise-Suchmaschinen auf Basis von Open Source-Software.⁴ Gemäß einer im Jahr 2015 durchgeführten Befragung ist der Bereich der Suchmaschinentechnologie das am häufigsten genutzte Einsatzgebiet für Open Source-Software in deutschen wissenschaftlichen Bibliotheken.⁵ Auf Basis von Open Source-Software wird Suchmaschinentechnologie von Bibliotheken in Online-Katalogen, Wissenschaftssuchmaschinen⁶ und Repositorien⁷ genutzt.

¹ Vgl. Jansen 2014, S. 774.

² Definition siehe Kap. 2.1.2.

³ Vgl. Kostädt 2015, S. 517.

⁴ Vgl. Christensen 2013, S. 4.

⁵ Vgl. Maaß 2016, S. 65. Weitere dort genannte Einsatzgebiete sind Integrierte Bibliothekssysteme, Linkresolver und Workflow-Systeme für die Digitalisierung.

⁶ Vgl. Weinhold et. al. 2011.

⁷ Vgl. Dobratz 2007.

Diese Arbeit wird sich zunächst mit sogenannten Katalogsuchmaschinen⁸ und darauf aufbauend mit Enterprise-Suchmaschinen als technischer Basis dieser Katalogsuchmaschinen beschäftigen. Aufgrund der oben erwähnten Bedeutung von Open Source-Software im Bereich der Suchmaschinentechnologie wird der Fokus dieser Arbeit folglich auf Open Source-Software liegen.

Wichtige Bestandteile der Definition von Open Source sind ein frei verfügbarer Quelltext sowie die Erlaubnis zur beliebigen Nutzung, Veränderung und Verbreitung der Software.⁹ Die Motivation für Open Source als Entwicklungsmodell für Software wird in der besseren Arbeit der beteiligten Softwareentwickler, der schnelleren Umsetzbarkeit von relevanten Nutzerwünschen und dem schnelleren Auffinden von Fehlern und Sicherheitslücken in der Software gesehen.¹⁰

Trotz der intensiven Nutzung von Open Source-Software im Bereich der Suchmaschinentechnologie durch Bibliotheken liegen bisher keine Vergleiche von Enterprise-Suchmaschinen im Kontext von Katalogsuchmaschinen vor. Ein genereller Vergleich von Enterprise-Suchmaschinen auf Open Source-Basis wurde von Baeza-Yates, Ribeiro-Neto und Middleton durchgeführt. Dabei wurden sowohl qualitative Informationen zu den untersuchten Suchmaschinen zusammengetragen als auch quantitative Methoden in Form eines Benchmarkings angewendet.¹¹

Darüber hinaus gibt es aktuell weder systematisch und regelmäßig erfasste Informationen darüber, welche Bibliothek welchen Online-Katalog einsetzt, noch darüber, auf welcher technischen Basis dieser aufsetzt.¹²

⁸ Definition siehe Kap. 2.1.3.

⁹ Vgl. Open Source Initiative 2007.

¹⁰ Vgl. Schnalke 2014, S. 47.

¹¹ Vgl. Baeza-Yates/Ribeiro-Neto/Middleton 2011.

¹² Bis 1998 wurde regelmäßig zweijährig vom Deutschen Bibliotheksinstitut (DBI) die Verbreitung von Integrierten Bibliothekssystemen erhoben. Seit der Auflösung des DBI gab es keine derart umfassende Erhebung dieser Art mehr, vgl. Oßwald 2016, S. 1.

Aufgrund der oben dargestellten Vorüberlegungen definiert sich das Ziel dieser Arbeit wie folgt:

1.1 Ziel der Arbeit

Das Ziel dieser Arbeit ist es, einen Vergleich von Enterprise-Suchmaschinen durchzuführen, die als technische Basis bibliothekarischer Katalogsuchmaschinen eingesetzt werden.

1.2 Forschungsfragen

Aus dem Ziel dieser Arbeit ergeben sich die beiden Forschungsfragen der Arbeit.

Forschungsfrage I

Welche Enterprise-Suchmaschinen werden als technische Basis bibliothekarischer Katalogsuchmaschinen verwendet?

Forschungsfrage II

Welche Funktionen bieten die als technische Basis bibliothekarischer Katalogsuchmaschinen eingesetzten Enterprise-Suchmaschinen? Wo gibt es Gemeinsamkeiten, Unterschiede und Alleinstellungsmerkmale?

2 Entwicklung von Online-Katalogen und technische Grundlagen von Enterprise-Suchmaschinen

Der Online-Katalog hat alle übrigen Formen des Bibliothekskatalogs weitgehend abgelöst.¹³ Ein Online-Katalog wird von Umstätter und Wagner-Döbler definiert als eine Datenbank,

„[...] auf die von zugelassenen Terminals aus interaktiv zugegriffen werden kann. [...] Je nach Datenbanksoftware, Datenbankstruktur und Retrievalmöglichkeiten decken Online-Kataloge sehr unterschiedliche Wünsche ab.“¹⁴

Auch von Haller wird ein Online-Katalog beschrieben als Datenbank

„[...] mit strukturierten Datensätzen für Bücher und sonstige Bibliotheksmaterialien, die nach verschiedenen Suchkriterien oder Attributen und mit verschiedenen Suchmethoden auffindbar und auch als Verzeichnis ausgebaut sind.“¹⁵

Nach der Begriffsbestimmung wichtiger Formen des Online-Katalogs sowie des Begriffs „Enterprise-Suchmaschine“ werden im Folgenden größere Entwicklungsschritte des Online-Katalogs nachgezeichnet. Anschließend wird auf die technischen Grundlagen von Enterprise-Suchmaschinen eingegangen.

2.1 Begriffsbestimmung

Die Auflistung der Begriffe erfolgt in alphabetischer Reihenfolge.

2.1.1 Discovery System

Ein Discovery System ist ein Online-Katalog auf Basis einer Enterprise-Suchmaschine, der eine vollständige inhaltliche Abdeckung aller für eine Literaturrecherche relevanten Quellen anstrebt.¹⁶

¹³ Vgl. Gantert 2016, S. 288. Zu Bibliothekskatalogen und deren Formen vgl. Gantert 2016, S. 246 ff.; Haller 1998, S. 16 ff. und Umstätter/Wagner-Döbler 2005, S. 5 ff.

¹⁴ Umstätter/Wagner-Döbler 2005, S. 114.

¹⁵ Haller 1998, S. 18.

¹⁶ Vgl. Kostädt 2015, S. 518 ff. Zur Entstehung des Begriffes „Discovery System“ vgl. Caplan 2012, S. 108 f.

2.1.2 Enterprise-Suchmaschine

Als Enterprise-Suchmaschine wird eine auf die Suche in heterogenen Informationsquellen ausgelegte Suchmaschine bezeichnet.¹⁷ Diese bildet den Kern von Discovery Systemen und Katalogsuchmaschinen sowie verwandter Suchsysteme wie Wissenschaftssuchmaschinen¹⁸ oder Repositorien¹⁹.

2.1.3 Katalogsuchmaschine

Eine Katalogsuchmaschine ist ein Online-Katalog auf Basis einer Enterprise-Suchmaschine, der seinen Schwerpunkt auf der Suche im Medienbestand einer Bibliothek oder einer Verbundregion hat.²⁰

2.1.4 Online Public Access Catalog (OPAC)

Ein Online Public Access Catalog (OPAC) ist die Suchkomponente eines integrierten Bibliothekssystems²¹ und basiert nicht auf Suchmaschinentechnologie.

2.2 Entwicklung von Online-Katalogen

Die ersten Online-Kataloge in den 1960er- und 1970er-Jahren waren Nebenprodukte der Bibliotheksautomatisierung im Bereich der Ausleihverbuchung. Ihr Sucheinstieg bestand wie der des Zettelkatalogs auf einem präkoordinierten Index. Eine Weiterentwicklung der Online-Kataloge führte dann einen postkoordinierten Sucheinstieg mithilfe des Booleschen Modells ein.²²

Diese Form des Online-Katalogs wird als OPAC bezeichnet. In den OPAC lassen sich alle konventionellen Katalogformen überführen, so dass letztere in den 1980er-Jahren nach und nach von den OPACs verdrängt wurden. Zunächst

¹⁷ Vgl. Kostädt 2015, S. 517 f.

¹⁸ Vgl. Weinhold et. al. 2011.

¹⁹ Vgl. Dobratz 2007.

²⁰ Vgl. Kostädt 2015, S. 516 ff.

²¹ Vgl. Herrlich/Ledl/Tréfas 2013, S. 43.

²² Vgl. Antelman/Lynema/Pace 2006, S. 128. Zum Booleschen Modell vgl. Kap. 2.3.

erfolgte der Zugriff auf den OPAC über lokale Netze, bevor in den 1990er-Jahren immer mehr webbasierte OPACs aufkamen. Der OPAC ist seither eine Komponente von Integrierten Bibliothekssystemen.²³

Die Inhalte dieser webbasierten OPACs gehören zum sogenannten Deep Web, dem Teil des World Wide Web, der nicht ohne weiteres für Internetsuchmaschinen zugänglich ist. Um den Zugang zu Informationsquellen im Deep Web zu erleichtern, wurden ab Mitte der 1990er Jahre Metasuchsysteme wie der Karlsruher Virtuelle Katalog (KVK) entwickelt. Ein Metasuchsystem führt im Hintergrund eine parallele Recherche in den vom Benutzer gewählten Zielsystemen aus und fasst anschließend die einzelnen Trefferlisten zu einer Ergebnisliste zusammen. Im Gegensatz zu Metasuchsystemen verfügen Metasuchportale über ein Authentifizierungsmodul. Dadurch ist es möglich, auch lizenzpflichtige Datenquellen in die Recherche einzubeziehen und dem Benutzer verschiedene personalisierte Dienstleistungen anzubieten.²⁴

Mit der weiteren Entwicklung des Internet und dem Aufkommen von Internetsuchmaschinen wurde das Nutzerverhalten wesentlich verändert. Die von Suchmaschinen erfüllten Anforderungen wie intuitive Bedienbarkeit, schnelle Antwortzeiten und effizientes Ranking wurden von den Benutzern auch an bibliothekarische Suchsysteme wie OPACs und Metasuchsysteme gestellt.²⁵

Die Entwicklung von OPACs und Metasuchsystemen hat sich jedoch mehr auf die Benutzeroberfläche als auf die Suchtechnologie konzentriert. Die Suchtechnologie von OPACs baut auf dem Prinzip des Exact Match und dem Booleschen Retrieval-Modell auf, während andere Suchsysteme auf das Prinzip des Best Match und Information-Retrieval-Modelle wie das Vektorraummodell oder das probabilistische Modell setzen.²⁶ Darüber hinaus basieren OPACs in der Regel auf relationalen Datenbanksystemen, deren Struktur für die Speicherung

²³ Vgl. Gantert 2016, S. 228.

²⁴ Vgl. Kostädt 2015, S. 514 f.

²⁵ Vgl. Kostädt 2015, S. 516.

²⁶ Vgl. Kinstler 2013, S. 18 f.

bibliographischer Daten ausgelegt ist. Sowohl die Abfrage der Daten als auch die Indexierung von großen, heterogenen Datenbeständen sind vergleichsweise zeitintensiv.²⁷ Metasuchsysteme haben keine eigene Datenhaltung und haben dadurch ebenfalls Nachteile beim Antwortzeitverhalten, beim globalen Ranking oder dem Einsatz linguistischer Verfahren zur Erhöhung des Recall.²⁸

Um diese Nachteile aufzufangen und auf das veränderte Nutzerverhalten zu reagieren, lag der Einsatz von Suchmaschinentechnologie für die Suche in Katalogdaten nahe.²⁹ Zu den Vorteilen der Suchmaschinentechnologie gehören schnelle Antwortzeiten, die Sortierung der Trefferliste nach Relevanz, die Unterbreitung von Suchvorschlägen während und nach der Eingabe, der Einsatz linguistischer Verfahren sowie die facettierte Suche.³⁰

Die Bibliotheken und Verbundzentralen in Deutschland haben bereits in den 1990er Jahren Suchmaschinentechnologie für die Indexierung von Katalogdaten eingesetzt. Dazu wurde die Suchmaschinentechnologie aus dem Bereich der Enterprise Search weiterentwickelt.³¹ Bei der Enterprise Search handelt es sich um ein „[a]uf digitale Informationsbestände (Dokumente, Daten, Fakten, Dateien) im Unternehmen spezialisiertes Suchsystem.“³²

Dabei kamen zunächst kommerzielle Enterprise-Suchmaschinen für die Entwicklung bibliothekarischer Softwareprodukte zum Einsatz. So nutzten zum Beispiel die HBZ-Suchmaschine und die Wissenschaftssuchmaschine BASE (Bielefeld Academic Search Engine) der UB Bielefeld jeweils die

²⁷ Vgl. Kostädt 2015, S. 516

²⁸ Vgl. Kostädt 2015, S. 515.

²⁹ Vgl. Kostädt 2008, S. 105.

³⁰ Vgl. Jansen 2014, S. 774.

³¹ Vgl. Kostädt 2015, S. 517.

³² Granitzer 2009, S. 272. Für einen Überblick zur Enterprise Search vgl. Bahrs 2009 u. Hawking 2011.

Suchtechnologie von FAST.³³ Der Online-Katalog der North Carolina State University Libraries wurde mithilfe der Suchtechnologie von Endeca realisiert und bot eine Sortierung nach Relevanz sowie facettierte Navigation.³⁴

Dieser Online-Katalog war das Vorbild für die an der Villanova University entwickelte Katalogsuchmaschine VuFind, die im Gegensatz zu ihrem Vorbild jedoch auf Basis von Open Source-Software entwickelt wurde.³⁵ Entscheidend für die Entwicklung von VuFind war die Verwendung der Open Source Enterprise-Suchmaschine Solr als technische Basis.³⁶ Erst das Aufkommen derartiger Open Source Enterprise-Suchmaschinen hat die Entwicklung eigener Katalogsuchmaschinen durch Bibliotheken auf breiterer Basis ermöglicht.³⁷

Einen wichtigen Beitrag bei der Entwicklung eigener Katalogsuchmaschinen durch Bibliotheken hat der sogenannte Katalog 2.0 geleistet. Dieser sollte die digitalen Angebote und Informationen einer Bibliothek unter einer Oberfläche vereinen und ein Instrument der allgemeinen Informationssuche in Zeitschriften, im Deep Web und in Repositorien sein.³⁸ Seine Funktionalitäten sollten sich an Suchmaschinen orientieren und durch soziale Funktionen wie etwa der Vergabe von eigenen Tags für Medien oder der Möglichkeit zur Bewertung und Kommentierung von Medien ergänzt werden.³⁹ Nach Auffassung von Christensen erwarten die Benutzer jedoch nicht soziale Funktionen, sondern leistungsfähige und komfortable Suchfunktionen von einem modernen Online-Katalog.⁴⁰ Auch Lewandowski sieht nicht die Umsetzung von Web 2.0-Funktionen, son-

³³ Vgl. Lewandowski 2006, S. 75 f. Die HBZ-Suchmaschine ist inzwischen zu Elasticsearch gewechselt, vgl. Baron 2016, S. 10. BASE baut inzwischen auf der Katalogsuchmaschine VuFind auf, vgl. Pieper 2011.

³⁴ Vgl. Antelman/Lynema/Pace 2006, S. 129 f.

³⁵ Vgl. Houser 2009, S. 95.

³⁶ Vgl. Nagy 2011, S. 5 f.

³⁷ Vgl. Christensen 2013, S: 4.

³⁸ Vgl. Kneifel 2009, S. 28.

³⁹ Vgl. Kneifel 2010, S. 41 ff.

⁴⁰ Vgl. Christensen 2013, S. 11.

dern eine Steigerung der Qualität des Suchergebnisses als einen Schlüsselfaktor für Online-Kataloge.⁴¹

In Folge der verstärkten Eigenentwicklung von Katalogsuchmaschinen durch Bibliotheken führten die kommerziellen Anbieter neue als Discovery Systeme bezeichnete Produkte auf dem Markt ein.⁴² In ihrem Funktionsumfang sind sich Discovery Systeme und Katalogsuchmaschinen sehr ähnlich.⁴³ Unterscheiden lassen sie sich anhand der darin aggregierten Datenquellen. Während Katalogsuchmaschinen die Datenquellen einer Bibliothek oder Verbundregion umfassen, streben die derzeit ausschließlich von kommerziellen Anbietern verfügbaren Discovery Systeme eine vollständige inhaltliche Abdeckung an.⁴⁴ In Deutschland teilen sich die vier Systeme EBSCO Discovery Service (EDS), Primo, Summon und World Cat Local den Markt für Discovery Systeme.⁴⁵ EDS⁴⁶ und World Cat Local⁴⁷ basieren auf proprietärer Suchtechnologie. Primo (Lucene)⁴⁸ und Summon (Solr)⁴⁹ bauen wie die von Bibliotheken entwickelten Katalogsuchmaschinen auf Open Source Enterprise-Suchmaschinen auf.

Mit der Entwicklung von Katalogsuchmaschinen und Discovery Systemen hat sich die Suchumgebung von den übrigen Bestandteilen eines Integrierten Bibliothekssystems gelöst. Die aktuellen Entwicklungen im Bereich der Bibliothekssysteme in Richtung des Cloud Computing werden nach Auffassung von Kostädt zu einer Vereinigung verschiedener von einer Bibliothek benötigter Funktionalitäten inklusive der Suchumgebung führen⁵⁰.

⁴¹ Vgl. Lewandowski 2009, S. 592.

⁴² Vgl. Tannhof 2013, S. 6.

⁴³ Vgl. Kostädt 2015, S. 518.

⁴⁴ Vgl. Kostädt 2015, S. 518.

⁴⁵ Vgl. Jansen 2014, S. 776. Für eine ausführliche Darstellung zu den genannten Discovery Systemen vgl. Vaughan 2011.

⁴⁶ Vgl. Jansen/Kemner-Heek/Schweitzer 2010, S. 11.

⁴⁷ Vgl. Vaughan 2011, S. 14.

⁴⁸ Vgl. Jansen/Kemner-Heek/Schweitzer 2010, S. 12.

⁴⁹ Vgl. Jansen/Kemner-Heek/Schweitzer 2010, S. 14.

⁵⁰ Vgl. Kostädt 2015, S. 521 f.

2.3 Technische Grundlagen von Enterprise-Suchmaschinen

Eine Suchmaschine besteht aus einem Index als Datenbasis sowie den drei Komponenten Crawler, Indexer und Query Modul. Die Aufgabe des Crawlers besteht darin, neue Inhalte ausfindig zu machen und diese an den Indexer zu übergeben. Dieser Prozess wird Crawling genannt. Der Indexer speichert diese Inhalte in einem Index ab und bereitet diese für eine effiziente Verarbeitung von Suchanfragen auf. Diese Aufbereitung wird als Indexierung bezeichnet. Die Suchanfragen der Nutzer werden vom Query Modul entgegengenommen und für eine Abfrage des Index aufbereitet. Diese Abfrage wird Matching genannt und ihr Ergebnis wird abschließend aufbereitet und dem Nutzer als Suchergebnis präsentiert.⁵¹

Für die Indexierung von Katalogdaten durch eine Enterprise-Suchmaschine werden die Katalogdaten zunächst aus der relationalen Datenbank des Integrierten Bibliothekssystems exportiert. Anschließend werden sie über eine Schnittstelle in den Index der Enterprise-Suchmaschine importiert. Dabei lässt sich die Feldstruktur der Katalogdaten im Index der Suchmaschine abbilden.⁵²

Die Indexierung durch eine Suchmaschine gehört zum Bereich der automatischen Indexierung. Bei dieser werden alle Wörter innerhalb eines Dokumentes⁵³ identifiziert und zu Indextermen verarbeitet. Die Indexterme werden im Index abgespeichert und innerhalb des Index dem Dokument zugeordnet. Die übliche Form des Index ist dabei eine invertierte Liste. Zur Erzeugung der Indexterme werden informationslinguistische Verfahren eingesetzt. Diese werden durch Informationen aus textstatistischen Verfahren ergänzt.⁵⁴

⁵¹ Vgl. Lewandowski 2013, S. 497 ff.

⁵² Vgl. Kostädt 2015, S. 517 f.

⁵³ Als Dokument ist hier das zu indexierende Objekt - also etwa auch Katalogdaten - zu verstehen.

⁵⁴ Vgl. Lepsky 2013, S. 272 f.

Die informationslinguistischen Verfahren sorgen primär für die Zusammenführung von sprachlich unterschiedlichen Formen eines Wortes. Diese Verfahren lassen sich in Stemming-Verfahren und wörterbuchbasierte Verfahren unterteilen. Beim Stemming wird die grammatikalische Grundform eines Wortes durch die Rückverfolgung der Flexionsregeln der jeweiligen Sprache erzeugt. Dieses Verfahren eignet sich gut für stark regelhafte Sprachen wie das Englische. Für komplexere Sprachen wie das Deutsche kommen wörterbuchbasierte Verfahren zum Einsatz. Die Grundform eines Wortes wird dabei mithilfe von dessen Wörterbucheinträgen ermittelt. Darüber hinaus können mit diesem Verfahren Komposita in ihre Bestandteile zerlegt sowie Mehrwörter und Synonyme erkannt werden, wodurch zusätzliche Indexterme generiert werden. Zudem werden sogenannte Stoppwörter mit dem Ziel einer beschleunigten Indexierung von ebendieser ausgenommen.⁵⁵

Davon ausgehend, dass nicht alle Wörter innerhalb eines Dokumentes gleich wichtig sind, werden bei den textstatistischen Verfahren die Worthäufigkeiten innerhalb eines Dokumentes ermittelt. Die Termhäufigkeit gibt die absolute Häufigkeit des Vorkommens eines Wortes in einem Dokument an. Die Dokumenthäufigkeit gibt die absolute Häufigkeit der Dokumente an, in denen ein bestimmtes Wort vorkommt.⁵⁶ Spezielle Gewichtungsverfahren wie die dokumentspezifische Wortgewichtung und die inverse Dokumenthäufigkeit können als Rankingfaktoren eingesetzt werden.⁵⁷

Sowohl die textstatistischen als auch die informationslinguistischen Verfahren müssen außer bei der Indexierung auch bei der Suche angewendet werden, damit ein erfolgreiches Matching durchgeführt werden kann.⁵⁸ Das Matching wird durch Information-Retrieval-Modelle näher spezifiziert.⁵⁹ Die wichtigsten Information-Retrieval-Modelle sind das Boolesche Modell, das Vektorraummodell und das probabilistische Modell.

⁵⁵ Vgl. Lepsky 2013, S. 273 ff.

⁵⁶ Vgl. Lepsky 2013, S. 279 f.

⁵⁷ Vgl. Lewandowski 2005, S. 91 ff.

⁵⁸ Vgl. Nohr 2005, S. 146.

⁵⁹ Vgl. Fuhr 2013, S. 322.

Das Boolesche Modell arbeitet nach dem Prinzip des Exact Match. Es werden nur Dokumente gefunden, deren Indexterme exakt den Suchtermen entsprechen. Dokumente, deren Indexterme den Suchtermen ähnlich sind, werden nicht gefunden. Darüber hinaus werden alle Dokumente in der Trefferliste als gleich relevant eingestuft.⁶⁰

Das Vektorraummodell arbeitet nach dem Prinzip des Best Match und geht davon aus, dass nicht alle Dokumente gleich relevant sind. Die Terme werden durch Vektoren innerhalb eines Vektorraums repräsentiert und mithilfe der Vektorrechnung wird die Ähnlichkeit von Indextermen und Suchtermen berechnet. Dadurch kann eine nach dem Grad der Ähnlichkeit sortierte Trefferliste ausgegeben werden.⁶¹

Das probabilistische Modell arbeitet ebenfalls nach dem Prinzip des Best Match und geht davon aus, dass nur eine Wahrscheinlichkeit dafür ermittelt werden kann, ob ein Dokument für eine Suchanfrage relevant ist oder nicht. Es berechnet die Ähnlichkeit abhängig von der Häufigkeit der Suchterme innerhalb eines Dokumentes. Auch hier kann eine nach dem Grad der Ähnlichkeit sortierte Trefferliste ausgegeben werden.⁶²

2.4 Zusammenfassung

Online-Kataloge haben sich von elektronischen Nachbildungen über OPACs und Metasuchsysteme hin zu Systemen auf Basis von Suchmaschinentechnologie entwickelt. Letztere werden gekennzeichnet durch schnelle Antwortzeiten, eine Sortierung nach Relevanz, Suchvorschläge, den Einsatz linguistischer Verfahren, eine facetiierte Suche sowie die Bereitstellung von Daten aus heterogenen Datenquellen innerhalb eines gemeinsamen Index. Die Suche setzt nicht mehr nur auf das Prinzip des Exact Match (Boolesches Modell), sondern

⁶⁰ Vgl. Lewandowski 2005, S. 80 ff.

⁶¹ Vgl. Lewandowski 2005, S. 83 ff.

⁶² Vgl. Lewandowski 2005, S. 86.

auch auf das Prinzip des Best Match (Vektorraummodell, probabilistisches Modell). Die Online-Kataloge mit Suchmaschinentechnologie lassen sich in Katalogsuchmaschinen und Discovery Systeme unterscheiden. Discovery Systeme streben inhaltlich die Zugänglichmachung aller relevanten Quellen für eine Literaturrecherche an und werden daher bisher nur von kommerziellen Anbietern angeboten. Katalogsuchmaschinen enthalten schwerpunktmäßig den Bestand einer Bibliothek oder Verbundregion und werden auch von den Bibliotheken selbst oder den Verbundzentralen entwickelt und angeboten. Die technische Basis dieser Eigenentwicklungen sind Enterprise-Suchmaschinen.

3 Katalogsuchmaschinen in Bibliotheken

3.1 Ziel

Das Ziel dieses Kapitels ist die Beantwortung der Forschungsfrage I: Welche Enterprise-Suchmaschinen werden als technische Basis bibliothekarischer Katalogsuchmaschinen verwendet?

3.2 Methodik

3.2.1 Ermittlung der relevanten Institutionen

Als relevant für die Untersuchung wurden alle Institutionen angesehen, welche die infrastrukturellen und personellen Kapazitäten aufweisen, um selbst eine Katalogsuchmaschine zu entwickeln. Diese Kapazitäten wurden bei den folgenden vier Gruppen von Institutionen⁶³ gesehen:

1. Verbundzentralen der Bibliotheksverbände⁶⁴,
2. Öffentliche Bibliotheken in Städten mit mehr als 400.000 Einwohnern (Sektion 1 des Deutschen Bibliotheksverbandes dbv⁶⁵),
3. Bibliotheken von überregionaler Bedeutung (Deutsche Nationalbibliothek, Staatsbibliothek zu Berlin, Bayerische Staatsbibliothek, Zentrale Fachbibliotheken)⁶⁶,
4. Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden⁶⁷.

⁶³ Eine Liste von Institutionen aus dem Bibliotheksbereich mit eigener Softwareentwicklung findet sich bei Borst, vgl. Borst 2016. Dieser Beitrag wurde nach Durchführung der in diesem Kapitel beschriebenen Untersuchung veröffentlicht und konnte daher bei der Auswahl der Institutionen nicht berücksichtigt werden.

⁶⁴ Vgl. Arbeitsgemeinschaft der Verbundsysteme 2013, S. 26 f.

⁶⁵ Vgl. Deutscher Bibliotheksverband 2016.

⁶⁶ Vgl. Gantert 2016, S. 13 ff.

⁶⁷ Die Zahl der Studierenden wurde analog zur Einwohnerzahl bei den Öffentlichen Bibliotheken als Bezugsgröße gesehen. Die Bibliotheken in dieser Gruppe lassen sich über die variable Auswertung der Deutschen Bibliotheksstatistik ermitteln, vgl. [hbz 01].

Als Grundlage für die weitere Untersuchung wurde eine Liste der als relevant eingestuften Institutionen erstellt. Die Büchereizentrale Schleswig-Holstein in der Sektion 1 des Deutschen dbv ist eine Fachstelle für Öffentliche Bibliotheken und wird daher nicht untersucht. Die Senatskanzlei für kulturelle Angelegenheiten des Landes Berlin ist der Träger des Verbundes der Öffentlichen Bibliotheken Berlins (VÖBB). Statt der Senatskanzlei wird der VÖBB untersucht.

Insgesamt wurden 54 Institutionen in das Untersuchungsfeld aufgenommen.⁶⁸

3.2.2 Literaturrecherche

Anschließend wurde eine Literaturrecherche nach Bibliotheken durchgeführt, die über die Einführung, die Aktualisierung oder den Wechsel ihres Online-Katalogs berichtet haben. Die Recherche wurde auf Beiträge in Periodika beschränkt, die zwischen 2008 und Ende Juni 2016 erschienen sind. Das Jahr 2008 wurde gewählt, weil die Katalogsuchmaschine VuFind in diesem Jahr an den Start gegangen ist.⁶⁹

Im untersuchten Zeitraum wurde für 23 der 54 Institutionen im Untersuchungsfeld ein entsprechender Beitrag gefunden.

3.2.3 Wahl der Untersuchungsmethode

Die folgenden Untersuchungsmethoden wurden als denkbar erachtet:

- Literaturrecherche,
- Befragung oder
- Webseitenanalyse.

Die in der Literatur enthaltenen Informationen könnten veraltet sein und müssten demzufolge verifiziert werden. Zudem wurden bei der in Kapitel 3.2.2 durch-

⁶⁸ Eine Liste für jede der Gruppen befindet sich in den Tabellen A1 bis A4 in Anhang 1.

⁶⁹ Vgl. Houser 2009, S. 93.

geführten Literaturrecherche nicht zu allen Institutionen des Untersuchungsfeldes entsprechende Berichte gefunden.

Eine wichtige Voraussetzung für den Erfolg der Untersuchung ist ein möglichst vollständiges Ergebnis. Eine Befragung wäre unabhängig von ihrer genauen Form⁷⁰ zu stark von der Rückläuferquote abhängig.

Bei der Webseitenanalyse sind die bisher genannten Nachteile nicht zu erwarten, da die Webseiten aktuell und jederzeit über einen Browser abrufbar sind. Daher wurde entschieden, die Webseiten aller Institutionen aus dem Untersuchungsfeld aufzurufen und die dort angebotenen Online-Kataloge zu untersuchen. Dieses Verfahren und seine Durchführung werden im Folgenden näher erläutert.

3.3 Durchführung

3.3.1 Verzeichnung der Online-Kataloge

Die Webseiten der Bibliotheken wurden im Zeitraum vom 11.07.2016 bis 23.07.2016 mit dem Browser Firefox aufgerufen und der Weg von der Startseite der Bibliothek über die Einstiegsseite des Online-Katalogs bis zur Detailseite eines Katalogeintrags in Form von Screenshots dokumentiert.⁷¹

Verzeichnet wurde dabei jeweils vorrangig der primär genutzte Online-Katalog der jeweiligen Institution. Ein OPAC wurde dabei nur verzeichnet, falls kein anderer Online-Katalog vorgefunden wurde. Im Falle der Verbundzentralen wurden sowohl die Verbundkataloge als auch potenzielle als Produkt für den Einsatz in Bibliotheken angebotene Systeme untersucht.

⁷⁰ Mögliche Formen der quantitativen Befragung sind die persönliche Befragung, die schriftliche Befragung, die telefonische Befragung und die Online-Befragung, vgl. Fühles-Ubach 2013.

⁷¹ Die Screenshots wurden mithilfe des Firefox-Addons FireShot als mehrseitige PDF-Dokumente gesichert und befinden sich inklusive einer Liste der untersuchten Online-Kataloge auf dem beiliegenden Datenträger. Informationen zu FireShot finden sich unter [Mozilla 01].

Die verzeichneten Kataloge wurden im Rahmen dieser Arbeit den folgenden Kategorien zugeordnet:

- selbst entwickelte Katalogsuchmaschine (E)⁷²,
- nachgenutzte Katalogsuchmaschine (N),
- Standardsystem (S) und
- OPAC (O).

Als selbst entwickelte Katalogsuchmaschinen werden die Systeme verstanden, die von der jeweiligen Bibliothek oder Verbundzentrale auf Basis einer Enterprise-Suchmaschine selbst entwickelt wurden. Unter nachgenutzten Katalogsuchmaschinen werden die Systeme verstanden, die auf der von einer anderen Bibliothek entwickelten Katalogsuchmaschine aufbauen. Als Standardsystem klassiert werden alle Discovery Systeme sowie die von einer Verbundzentrale oder einem Konsortium angebotenen Systeme. Als OPAC werden alle nicht auf Suchmaschinentechnologie basierenden Systeme betrachtet.

3.3.2 Identifizierung des Produktes

Anschließend wurden die Screenshots auf Rückschlüsse auf das hinter dem Online-Katalog stehende Produkt untersucht. Solche Rückschlüsse ließen sich aus der Literatur sowie aus den untersuchten Webseiten selbst ableiten. Aus den Webseiten selbst ergaben sich Hinweise etwa in Form von Produktbezeichnungen in Teilen des URL oder aus Text auf der Webseite wie „powered by“-Angaben oder Kennzeichnungen zum Copyright. Auf Basis der vorliegenden Hinweise wurde nach dem genutzten Produkt recherchiert.

Die abschließende Identifizierung des Produktes war über zwei unterschiedliche Verfahren möglich:

1. Eine vom Anbieter veröffentlichte Liste der Installationen oder
2. einen Vergleich mit einer Referenz.

⁷² In Klammern die in den Tabellen innerhalb dieses Kapitels und im Anhang genutzte Abkürzung für die jeweilige Kategorie.

Das erste Verfahren wurde im Falle des Produktes BibDIA der BiBer GmbH⁷³ und des HeBIS Discovery Systems (HDS)⁷⁴ angewendet. Für die übrigen Produkte wurde jeweils eine Referenz ermittelt und diese mit dem zu analysierenden Online-Katalog verglichen. Dieses Verfahren wird nun anhand von Beispielen näher beschrieben.

Die UB Duisburg-Essen hat Ende 2012 das Discovery System Primo eingeführt.⁷⁵ Bereits der Suchschlitz auf der Startseite der Bibliothekswebseite ist mit „Suche in Primo, dem Katalog der UB“⁷⁶ überschrieben. Nach Durchführen einer Suche lassen sich - wie in Abbildung 1 zu sehen - zwei weitere mögliche Erkennungsmerkmale ausmachen: „Primo by Ex Libris“ als Seitentitel sowie das Logo von Ex Libris als sogenanntes Favicon und „primo_library“ als Pfadangabe im URL der Trefferliste.⁷⁷

Um dies zu verifizieren wurde auf der Produktwebseite von Primo⁷⁸ nach einer Referenz gesucht und der Online-Katalog der der Bibliothek der ETH Zürich als solche ausgemacht. Die Abbildung 2 zeigt, dass der URL der Trefferliste dieses Online-Katalogs ebenfalls "primo_library" als Pfadangabe enthält.⁷⁹

Anhand dieses Merkmals im URL konnte dann beispielsweise - wie in Abbildung 3 gezeigt⁸⁰ - der Online-Katalog DISCO der ULB Münster als Primo-System erkannt werden.

⁷³ Vgl. [BibDIA 01].

⁷⁴ Vgl. [HeBIS 01].

⁷⁵ Vgl. Lützenkirchen/Falkenstein-Feldhoff 2014.

⁷⁶ [UB Duisburg-Essen 01].

⁷⁷ Der komplette URL ist „http://primo.ub.uni-due.de/primo_library/libweb/action/search.do?charconv=default&pcAvailabilityMode=false&vid=UDE&tab=localude&mode=Basic&ct=search&fn=search&vl%28freeText0%29=discovery+system“ und wurde am 24.07.2016 abgerufen.

⁷⁸ Vgl. [Ex Libris 01].

⁷⁹ Der komplette URL ist „[http://www.library.ethz.ch/primo_library/libweb/action/search.do?fn=search&ct=search&tab=default_tab&vl\(freeText0\)=discovery%20system&backFromPreferences=true](http://www.library.ethz.ch/primo_library/libweb/action/search.do?fn=search&ct=search&tab=default_tab&vl(freeText0)=discovery%20system&backFromPreferences=true)“ und wurde am 24.07.2016 abgerufen.

⁸⁰ Der komplette URL ist „http://disco.uni-muenster.de/primo_library/libweb/action/search.do?&dscnt=2&scp.scps=scope%3A%28noarticle%29&frbg=&tab=default_tab&dstmp=1469353774“



Abbildung 1: Kopfteil der Trefferliste im Online-Katalog der UB Duisburg-Essen

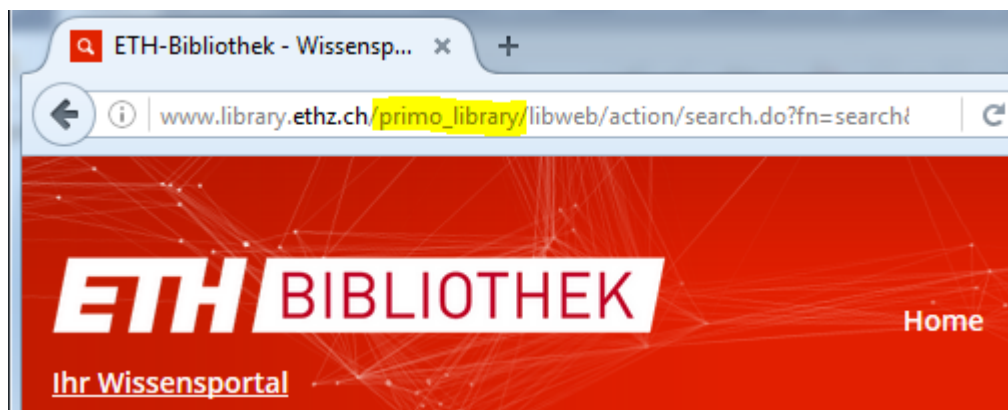


Abbildung 2: Kopfteil der Trefferliste im Wissensportal der Bibliothek der ETH Zürich

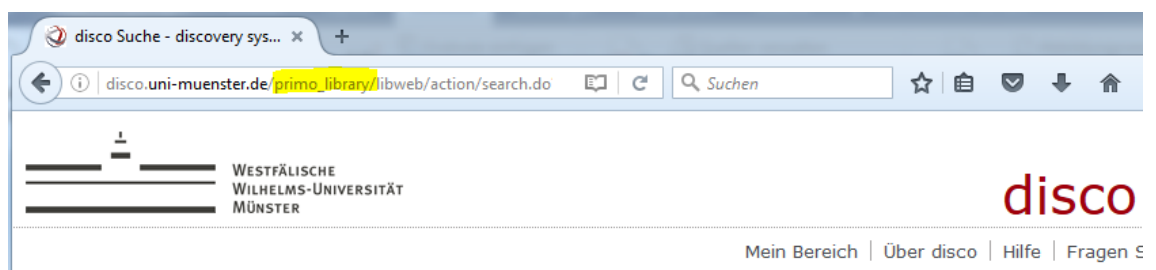


Abbildung 3: Kopfteil des Online-Katalogs DISCO der ULB Münster

390&srt=rank&ct=search&mode=Basic&dum=true&indx=1&fromLogin=true&vl(freeText0)=disco
 very+system&vid=ulb&fn=search“ und wurde am 24.07.2016 abgerufen.

Ebenfalls über den URL identifizierbar waren bei der Untersuchung die Produkte aDIS/BMS, EDS und TouchPoint.

Das Fachportal EconBiz der Deutschen Zentralbibliothek für Wirtschaftswissenschaften (ZBW) wurde im Herbst 2013 auf VuFind als Basis umgestellt.⁸¹ Als Referenz wurde die Demoseite⁸² von VuFind herangezogen. Die Suchschlitze beider Webseiten weisen ähnliche Elemente auf.⁸³

Weitere Ähnlichkeiten lassen sich in der jeweiligen Seite mit der Trefferliste einer Suche ausmachen: So enthalten in beiden Trefferlisten die dargestellten Treffer in identischer Reihenfolge die Elemente Nummer, Cover und Titel mit ausgewählten bibliographischen Angaben. Beide enthalten außerdem die Funktion „In Merkliste speichern“ (EconBiz) bzw. „Zu den Favoriten“ (VuFind Demoseite). Das Seitenende einer Seite der Trefferliste ist gleich aufgebaut und enthält jeweils Sprungmarken für die Seiten sowie die Funktionen „RSS-Feed erhalten“ (EconBiz) bzw. „RSS-Feed abonnieren“ (VuFind Demoseite), „Diese Suche als E-Mail versenden“ und „Suche speichern“.⁸⁴

In der jeweiligen Einzeltrefferanzeige treten weitere Ähnlichkeiten auf. Beide bestehen aus den bibliographischen Angaben, einer Funktionsleiste, einer Box mit ähnlichen Einträgen und Reitern für Exemplare und weitere Informationen.⁸⁵

Aufgrund der beschriebenen Ähnlichkeiten im Aufbau der jeweiligen Seiten konnte VuFind als Basis für EconBiz bestätigt werden.

⁸¹ Vgl. Pianos 2014, S. 54.

⁸² Vgl. [VuFind 01].

⁸³ Die Abbildung A1 in Anhang 2 zeigt den Suchschlitz von EconBiz und die Abbildung A2 in Anhang 2 den Suchschlitz der VuFind Demoseite.

⁸⁴ Die Abbildung A3 in Anhang 2 zeigt den entsprechenden Ausschnitt aus EconBiz und die Abbildung A4 in Anhang 2 den der VuFind Demoseite.

⁸⁵ Die Abbildung A5 in Anhang 2 zeigt die Einzeltrefferanzeige von EconBiz und die Abbildung A6 in Anhang 2 die der VuFind Demoseite.

Während der Untersuchung konnten die Produkte Aleph, BIBLIOTHECAPLUS, BIBLIOTHECAplus OPEN, Pica Search & Index, Portfolio, SISIS-SunRise und DigiBib IntrOX ebenfalls anhand des Seitenaufbaus identifiziert werden. Für die Produkte EDS und TouchPoint konnte dieses Verfahren zusätzlich zur Identifikation über den URL eingesetzt werden.

Die Tabelle 1 enthält eine Liste der Produkte, die während der Untersuchung mithilfe einer Referenz identifiziert wurden.⁸⁶

Tabelle 1: Verwendete Referenzen

Produkt	Hersteller	Referenz
aDIS/BMS	aStec	StB Herne
Aleph	Ex Libris	University of Amsterdam Library
BIBLIOTHECAplus	OCLC	Büchereien Wien
BIBLIOTHECAplus OPEN	OCLC	Öffentliche Bibliothek Ahrntal
DigiBib IntrOX	hbz	Bibliothek der TH Köln
EBSCO Discovery Service	EBSCO	UC Berkeley Library
Pica Search & Index (PSI)	OCLC	Gemeinsamer Verbundkatalog
Portfolio	BiblioMondo	Paris Town Hall
Primo	Ex Libris	Bibliothek der ETH Zürich
SISIS-SunRise	OCLC	StB Bremen
TouchPoint	OCLC	UB Bayreuth
VuFind	Villanova University	VuFind Demo

3.3.3 Ermittlung und Verifizierung der Eigenentwicklungen

Für zwölf Online-Kataloge konnte mithilfe der drei genutzten Verfahren kein Produkt ermittelt werden. Diese Online-Kataloge wurden daher als mögliche Eigenentwicklungen gewertet.

Diese Annahme wurde mithilfe von Informationen aus der Literatur verifiziert und im positiven Fall zusätzlich die Enterprise-Suchmaschine hinter der jeweiligen Eigenentwicklung identifiziert. Verwendet wurde dazu die in Kapitel 3.2.2 ermittelte Literatur sowie für den SLUB-Katalog beta der SLUB Dresden,

⁸⁶ Eine zusätzliche Liste mit den URLs findet sich in Tabelle A5 in Anhang 3.

Lukida der Verbundzentrale des Gemeinsamen Bibliotheksverbundes (GBV) und das TIB-Portal der Technischen Informationsbibliothek (TIB) zusätzlich recherchierte Literatur.⁸⁷

Für zwei Online-Kataloge – das Katalogportal der Deutschen Nationalbibliothek (DNB) und die Mediensuche der Bücherhallen Hamburg (HÖB) – brachte auch eine zusätzliche Literaturrecherche keine hinreichende Antwort auf die Frage, ob es sich um eine Eigenentwicklung handelt. Diese Frage wurde daher mit den HÖB per E-Mail⁸⁸ und mit der DNB im Rahmen eines Telefonates⁸⁹ geklärt. Beide Online-Kataloge sind wie vermutet Eigenentwicklungen. Auch deren technische Basis wurde erfragt.

3.4 Ergebnis

Die beschriebene Untersuchung galt in erster Linie der Verzeichnung des primär genutzten Online-Katalogs der untersuchten Institutionen. Ein OPAC wurde daher nur verzeichnet, falls kein anderer Online-Katalog vorgefunden wurde. Während der Untersuchung wurde dennoch beobachtet, dass ein OPAC als „Zweitkatalog“⁹⁰ durchaus nicht selten vorhanden zu sein scheint.

Zwölf der 59 untersuchten primär genutzten Online-Kataloge⁹¹ sind selbst entwickelte (20 Prozent) und fünf sind nachgenutzte Katalogsuchmaschinen (8 Prozent). 18 Systeme gehören der Kategorie der Standardsysteme (31 Prozent) an und 24 zu den OPAC-Systemen (41 Prozent). Im Untersuchungsfeld überwiegen somit die OPAC-Systeme vor den Standardsystemen. Eigenentwicklungen folgen erst an dritter Stelle. Ebenfalls hervorzuheben ist der Gegensatz zwischen den Bibliothekstypen. Die untersuchten Öffentlichen Bibliotheken

⁸⁷ Die Literaturrecherche in Kap. 3.2.2 wurde in Periodika durchgeführt. Bei der hier beschriebenen zusätzlichen Literaturrecherche wurden weitere Literaturarten wie etwa Protokolle, Mailinglisten oder Webseiten berücksichtigt.

⁸⁸ Vgl. Pinna 2016a und Pinna 2016b.

⁸⁹ Vgl. Meyer-Heß 2016.

⁹⁰ Vgl. dazu auch Herrlich/Ledl/Tréfas 2013, S. 43 f.

⁹¹ Das KOBV-Portal und ALBERT werden als ein System behandelt, vgl. Kap. 3.4.1.

setzen mit einer einzigen Ausnahme ausschließlich OPAC-Systeme ein, während bei den Wissenschaftlichen Bibliotheken die Online-Kataloge mit Suchmaschinentechnologie überwiegen.

Die Abbildung 4 visualisiert das Ergebnis für die einzelnen Gruppen der untersuchten Institutionen.

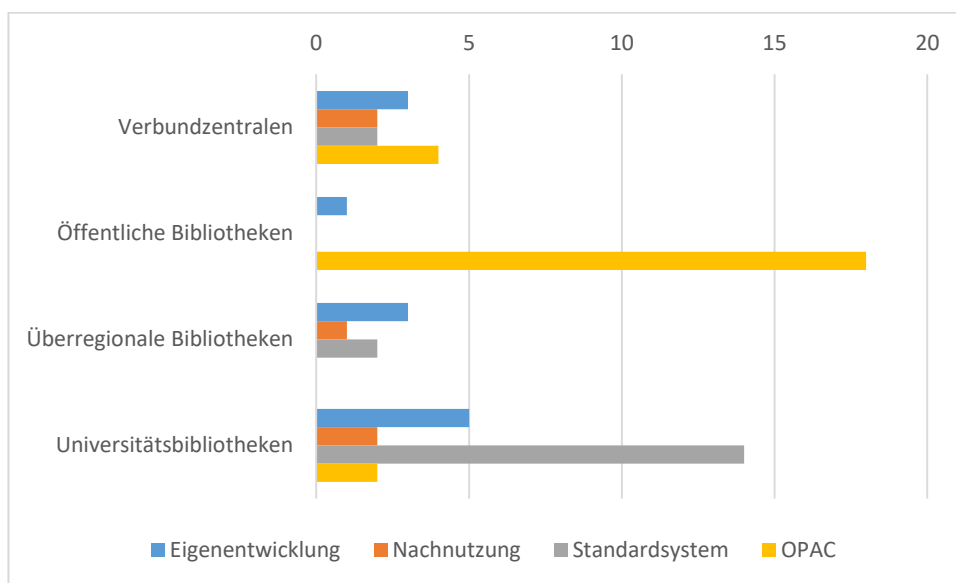


Abbildung 4: Ergebnis für die einzelnen Gruppen der untersuchten Institutionen

Im Folgenden werden die Ergebnisse für die Gruppen der untersuchten Institutionen ausführlicher dargestellt.

3.4.1 Verbundzentralen

Für die Verbundzentralen wurden sowohl die Verbundkataloge⁹² als auch die im Rahmen von Active Service Providing (ASP) angebotenen Produkte⁹³ untersucht.

Von den sechs Verbundkatalogen ist nur das KOBV-Portal eine Eigenentwicklung. Das Gateway Bayern basiert auf einem Standardsystem und die übrigen

⁹² Eine vollständige Liste befindet sich in Tabelle A6 in Anhang 4.

⁹³ Eine vollständige Liste befindet sich in Tabelle A7 in Anhang 4.

Verbundkataloge auf OPAC-Systemen. Das KOBV-Portal nutzt als Plattform das auch als ASP-Produkt angebotene System ALBERT.⁹⁴ Das KOBV-Portal und ALBERT werden daher als ein System gewertet. Im Bereich der ASP-Produkte ist ALBERT eine von drei Eigenentwicklungen und basiert wie Lukida der Verbundzentrale des GBV auf Solr.⁹⁵ DigiBib IntraOX des hbz basiert auf Elasticsearch.⁹⁶ Unter den übrigen ASP-Produkten befinden sich zwei Nachnutzungen von VuFind sowie ein Standardsystem.

3.4.2 Öffentliche Bibliotheken in Städten mit mehr als 400.000 Einwohnern

Mit Ausnahme der Bücherhallen Hamburg werden in den Öffentlichen Bibliotheken in Städten mit mehr als 400.000 Einwohnern⁹⁷ ausschließlich OPACs eingesetzt. Die Bücherhallen Hamburg setzen in ihrer Eigenentwicklung Solr als Suchmaschine ein.⁹⁸

3.4.3 Bibliotheken mit überregionaler Bedeutung

Drei der sechs Bibliotheken mit überregionaler Bedeutung⁹⁹ haben ihre Katalogsuchmaschine selbst entwickelt. Die ZBW nutzt VuFind als Basis für EconBiz nach.¹⁰⁰ Die beiden Staatsbibliotheken setzen jeweils ein Standardsystem ein. Die selbst entwickelten Katalogsuchmaschinen basieren auf der Enterprise-Suchmaschine Solr.¹⁰¹

⁹⁴ Vgl. Quitzsch 2014, S. 798.

⁹⁵ Für ALBERT vgl. Bertelmann/Szott/Höhnnow 2012, S. 373, für Lukida vgl. Beckmann 2016, S. 10.

⁹⁶ Vgl. Baron 2016, S. 10.

⁹⁷ Eine vollständige Liste befindet sich in Tabelle A8 in Anhang 4.

⁹⁸ Vgl. Pinna 2016a.

⁹⁹ Eine vollständige Liste befindet sich in Tabelle A9 in Anhang 4.

¹⁰⁰ Vgl. Pianos 2014, S. 54.

¹⁰¹ Für die TIB vgl. Hohfeld/Tobschall 2014, S. 68 und Heller 2015. Für die Deutsche Nationalbibliothek vgl. Meyer-Heß 2016. Für ZB MED vgl. Pössel 2015, S. 4.

3.4.4 Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden

Die Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden¹⁰² bieten ein sehr heterogenes Bild. 14 der 23 Systeme sind Standardsysteme (61 Prozent), fünf sind Eigenentwicklungen (22 Prozent), zwei gehören zu den OPAC-Systemen (9 Prozent) und zwei sind Nachnutzungen (9 Prozent). Drei der Eigenentwicklungen setzen Solr als Basis ein¹⁰³, während die Eigenentwicklungen der USB Köln auf Xapian¹⁰⁴ und der SUB Bremen auf CIXbase¹⁰⁵ setzen.

3.4.5 Technische Basis der Eigenentwicklungen

Die technische Basis der während der Untersuchung als Eigenentwicklung identifizierten Katalogsuchmaschinen ist in neun Fällen (75 Prozent) die Enterprise-Suchmaschine Solr. Eine andere Enterprise-Suchmaschine setzen nur das hbz (Elasticsearch), die USB Köln (Xapian) und die SUB Bremen (CIXbase) ein. Die Tabelle 2 enthält die Eigenentwicklungen und deren technische Basis im Überblick.

¹⁰² Eine vollständige Liste befindet sich in Tabelle A10 in Anhang 4.

¹⁰³ Für die UB Bochum vgl. Hagenbruch/Heise 2014, S. 817. Für die SLUB Dresden vgl. [SLUB Dresden 01]. Für die UB Heidelberg vgl. Maylein/Langenstein 2013, S. 191.

¹⁰⁴ Vgl. Flimm 2007, S. 187.

¹⁰⁵ Vgl. Blenkle et. al. 2015, S. 38.

Tabelle 2: Selbst entwickelte Katalogsuchmaschinen

Institution	Katalog / Produkt	Basis
KOBV-Zentrale, <i>Berlin</i>	KOBV-Portal, ALBERT	Solr
UB <i>Bochum</i>	"suchen"	Solr
SUB <i>Bremen</i>	E-LIB	CIXbase
SLUB <i>Dresden</i>	SLUB-Katalog beta	Solr
DNB, <i>Frankfurt/Main</i>	Katalogportal	Solr
Verbundzentrale des GBV, <i>Göttingen</i>	Lukida	Solr
Bücherhallen <i>Hamburg</i>	Suche	Solr
TIB, <i>Hannover</i>	TIB-Portal	Solr
UB <i>Heidelberg</i>	HEIDI	Solr
hbz, <i>Köln</i>	DigiBib IntrOX	Elasticsearch
USB <i>Köln</i>	USB-Portal, KUG	Xapian
ZB MED, <i>Köln</i>	LIVIVO	Solr

3.5 Vorteile und Nachteile der Eigenentwicklung einer Katalogsuchmaschine

Die Eigenentwicklung ihrer Katalogsuchmaschine garantiert Bibliotheken eine maximale Gestaltungsfreiheit bei der Implementierung der Software.¹⁰⁶ Um mit dieser Flexibilität handeln zu können ist für Flimm der Einsatz von Open Source-Software entscheidend. Nur dadurch erhalte eine Bibliothek die vollständige Kontrolle über die Software und könne schneller auf Probleme reagieren, Nutzerwünsche umsetzen sowie Projekte insgesamt realisieren.¹⁰⁷ Das während eines solchen offenen Projektes gewonnene Wissen lässt sich auch für andere Projekte gut anwenden und garantiert einen wirtschaftlichen und nachhaltigen Betrieb bei begrenzten finanziellen und personellen Ressourcen.¹⁰⁸

Diese Nachhaltigkeit sieht Borst durch die Art und Weise der Besetzung von Stellen für Softwareentwickler in Bibliotheken gefährdet. Die zumeist befristeten Projektstellen böten dem vorhandenen Personal keine längerfristige Perspek-

¹⁰⁶ Vgl. Christensen 2010, S. 329.

¹⁰⁷ Vgl. Flimm 2007, S. 186.

¹⁰⁸ Vgl. Flimm 2010, S. 315.

tive und seien unattraktiv für potenzielle Bewerber. Dies könne langfristig zu schwerwiegenden negativen Folgen führen, wenn etwa nur eine einzige Person mit der Entwicklung einer Software vertraut sei.¹⁰⁹ Darüber hinaus werden Eigenentwicklungen gegenüber anderen Lösungen als aufwändiger eingeschätzt.¹¹⁰

Aus den geschilderten Vor- und Nachteilen ergibt sich, dass der Einsatz von Open Source-Software und das längerfristige Vorhandensein von qualifiziertem Personal zu den entscheidenden Faktoren für die Eigenentwicklung einer Katalogsuchmaschine gehören.

3.6 Fazit

Das Ziel dieses Kapitels war die Beantwortung der Forschungsfrage I: Welche Enterprise-Suchmaschinen werden als technische Basis bibliothekarischer Katalogsuchmaschinen verwendet? Die Forschungsfrage I konnte für die untersuchten Institutionen beantwortet werden. In den als Eigenentwicklungen identifizierten Katalogsuchmaschinen werden die Enterprise-Suchmaschinen CIXbase, Elasticsearch, Solr und Xapian als technische Basis eingesetzt.

Insgesamt können die Ergebnisse der durchgeführten Untersuchung als repräsentativ für die Verbundzentralen und die drei Gruppen von Bibliotheken angesehen werden, da alle Institutionen jeder Gruppe vollständig untersucht wurden. Aus dem Teilergebnis der Öffentlichen Bibliotheken in Städten ab 400.000 Einwohner kann jedoch nicht auf ein Ergebnis für die gesamte Gruppe der Öffentlichen Bibliotheken geschlossen werden, da die Stichprobe nicht repräsentativ für alle Öffentlichen Bibliotheken ist. Ähnliches gilt für das Teilergebnis der Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden in Bezug auf alle Universitätsbibliotheken.

¹⁰⁹ Vgl. Borst 2016.

¹¹⁰ Vgl. Lützenkirchen/Falkenstein-Feldhoff 2014, S. 116; Tiemann 2016, S. 42 f. u. Wonke-Stehle/Christof, S. 18.

Für die während der Untersuchung gemachte Beobachtung, dass OPACs als Zweitkataloge durchaus nicht selten vorhanden sind, besteht weiterer Untersuchungsbedarf. Ebenso besteht weiterer Untersuchungsbedarf für die in der Untersuchung festgestellten Unterschiede bei der Nutzung von Discovery Systemen und Katalogsuchmaschinen durch Wissenschaftliche und Öffentliche Bibliotheken. Dafür wäre es notwendig und wünschenswert, wenn die für die durchgeführte Untersuchung erforderlichen Informationen in Zukunft regelmäßig vollständig für alle Bibliothekstypen erhoben werden.

Der erste entscheidende Faktor für die Eigenentwicklung einer Katalogsuchmaschine ist der Einsatz von Open Source-Software. Diese Erkenntnis stützt die in Kapitel 1 getroffene Entscheidung, den Fokus dieser Arbeit auf Open Source-Software zu legen. Die Ermittlung der tatsächlichen Auswirkungen von Open Source-Software bedarf weiterer Forschung. Der zweite entscheidende Faktor für die Eigenentwicklung einer Katalogsuchmaschine ist das längerfristige Vorhandensein von qualifiziertem Personal. Auch hier besteht weiterer Forschungsbedarf.

In Kapitel 4 werden die Funktionen der in diesem Kapitel als Ergebnis ermittelten Enterprise-Suchmaschinen untersucht und in einem Vergleich deren Unterschiede, Gemeinsamkeiten und Alleinstellungsmerkmale herausgearbeitet.

4 Vergleichende Analyse von Enterprise-Suchmaschinen

4.1 Ziel

Das Ziel dieses Kapitels ist die Beantwortung der Forschungsfrage II: Welche Funktionen bieten die als technische Basis bibliothekarischer Katalogsuchmaschinen eingesetzten Enterprise-Suchmaschinen? Wo gibt es Gemeinsamkeiten, Unterschiede und Alleinstellungsmerkmale?

4.2 Methodik

4.2.1 Festlegung der zu vergleichenden Enterprise-Suchmaschinen

Als technische Basis der in Kapitel 3 als Eigenentwicklungen identifizierten Katalogsuchmaschinen wurden die Enterprise-Suchmaschinen CIXbase, Elasticsearch, Solr und Xapian ermittelt. Bei einer ersten Recherche zu diesen Enterprise-Suchmaschinen ergab sich, dass die Projektwebseite von CIXbase nicht mehr existiert.¹¹¹ Eine mögliche Beschaffung auf Anfrage bei der SUB Bremen entspricht nicht der freien Verfügbarkeit im Sinne der Definition von Open Source.¹¹² Daher wurde CIXbase von der weiteren Untersuchung ausgeklammert und somit beschäftigt sich dieses Kapitel mit dem Vergleich von Elasticsearch, Solr und Xapian.

4.2.2 Literaturrecherche

Die Dokumentation für Elasticsearch besteht aus der *Elasticsearch Reference*¹¹³ und „Elasticsearch: The Definitive Guide“ (im Folgenden: *Elasticsearch Guide*). Beide stehen online zur Verfügung und können von

¹¹¹ Laut Blenkle/Ellis/Haake 2009, S. 619, FN 4 ist „<http://cixbase.dyndns.org/CiXbase/cixdocs/>“ der URL der Projektwebseite. Diese wurde am 01.08.2016 aufgerufen und es erschien die Meldung „Der Server unter cixbase.dyndns.org konnte nicht gefunden werden.“

¹¹² Vgl. Kap. 1.

¹¹³ Die Einzelnachweise für alle in diesem Abschnitt kursiv gesetzten Dokumente können dem Dokumentationsverzeichnis entnommen werden.

Benutzern mit entsprechendem Zugang editiert werden. Die Elasticsearch Reference gibt es jeweils für spezifische Versionen und den Elasticsearch Guide für die Versionsfamilien 1.x und 2.x. Außerdem ist der Guide mit dem Stand der Version 1.4.0 als Buch¹¹⁴ erschienen.

Die *Apache Solr Documentation* liegt für die aktuelle Version 6.1.0 vor und enthält eine Dokumentation der Java-Schnittstelle sowie verschiedene als Nachschlagewerke konzipierte Dokumente. Zu letzteren gehören das *Apache Solr Wiki*, der *Solr Quick Start* und der *Apache Solr Reference Guide*. Alle Dokumente werden kollaborativ erstellt. Zu mehreren Versionen von Solr wurden Bücher¹¹⁵ veröffentlicht.

Für Xapian gibt es die *Xapian Documentation*, den Guide „*Getting Started with Xapian*“ und den *Xapian Administrator's Guide* für Systemadministratoren. Für die Xapian Documentation und den Xapian Administrator's Guide fehlen genaue Angaben darüber, für welche Xapian-Version die Dokumente gültig sind. Der kollaborativ erstellte Guide „*Getting Started with Xapian*“ liegt für die Version 1.2 in einer Online-Version vor.

Über die bisher erwähnten offiziellen Dokumentationen und Buchveröffentlichungen hinaus gibt es Webseiten mit Vergleichen verschiedener Enterprise-Suchmaschinen.¹¹⁶

4.2.3 Wahl der Untersuchungsmethode

Die folgenden Untersuchungsmethoden wurden als denkbar erachtet:

- Funktionstests der Enterprise-Suchmaschinen,
- Literaturanalyse (Primärliteratur) oder
- Literaturanalyse (Sekundärliteratur).

¹¹⁴ Vgl. Gormley/Tong 2015.

¹¹⁵ Vgl. exemplarisch die Bibliographie bei Apache Solr 2016a.

¹¹⁶ Als Beispiel sei [DB-Engines 01] genannt.

Die Funktionstests sind im Rahmen dieser Arbeit in zwei Varianten möglich:

1. Die Nutzung vorgefertigter Virtueller Maschinen mit installierten Versionen der Enterprise-Suchmaschinen oder
2. selbst durchgeführte Installation und Einrichtung solcher Virtueller Maschinen.

Die zweite Variante ist der ersten vorzuziehen, da so eine vollständigere Kontrolle über die verwendeten Virtuellen Maschinen besteht. Bei vorgefertigten Systemen ist eine Prüfung der Installation nicht möglich. Außerdem können weder die Gleichheit der Basissysteme gewährleistet noch mögliche an der Software vorgenommene Änderungen ausgeschlossen werden.

Für die Durchführung von Funktionstests muss zunächst bekannt sein, welche Funktionen es überhaupt gibt. Genau dies festzustellen, ist das Ziel dieser Untersuchung. Daher ist ein Funktionstest nicht als Untersuchungsmethode geeignet.

Eine Literaturanalyse der Primärliteratur, etwa der offiziellen Dokumentationen der Enterprise-Suchmaschinen, ist einer Analyse der Sekundärliteratur (Vergleichswebseiten, Buchveröffentlichungen) vorzuziehen. Die Sekundärliteratur ist i.d.R. inhaltlich veraltet und für die Vergleichswebseiten fehlen genaue Angaben zu den Quellen der aufgeführten Informationen.

Daher wurde entschieden, die Untersuchung mithilfe einer Literaturanalyse der Primärliteratur durchzuführen. Die Durchführung der Untersuchung wird im Folgenden beschrieben.

4.3 Durchführung

4.3.1 Erstellung eines Kriterienkatalogs

Zu Beginn der Untersuchung wurde ein Kriterienkatalog mit den bei der Untersuchung zu betrachtenden Funktionen erstellt.

Bei der Sichtung der Literatur zu den Enterprise-Suchmaschinen wurde festgestellt, dass insbesondere der Funktionsumfang von Elasticsearch und Solr größer als der von einer Katalogsuchmaschine benötigte Funktionsumfang ist. Es wurde daher festgelegt, dass bei der Untersuchung nur der für eine Katalogsuchmaschine benötigte Funktionsumfang betrachtet wird. Dieser ergibt sich aus den in Kapitel 2.3 herausgearbeiteten Kennzeichen von Katalogsuchmaschinen:

- schnelle Antwortzeiten,
- Sortierung nach Relevanz,
- Suchvorschläge,
- Einsatz linguistischer Verfahren,
- facettierte Suche und
- Bereitstellung von Daten aus heterogenen Datenquellen innerhalb eines gemeinsamen Index.

Aufgrund der gewählten Untersuchungsmethode kann der Aspekt der Antwortgeschwindigkeit nicht gemessen werden und wird daher nicht berücksichtigt. Die Bereitstellung der Daten kann ebenfalls nicht untersucht werden. Dieser Aspekt soll jedoch indirekt berücksichtigt werden, indem Informationen zur Systemarchitektur und Datenhaltung einbezogen werden.

Der Kriterienkatalog wird in Kapitel 4.4 beschrieben.

4.3.2 Beschreibung der Enterprise-Suchmaschinen

Im zweiten Schritt wurden die offiziellen Dokumentationen gesichtet und für jede Enterprise-Suchmaschine eine Beschreibung verfasst. Anhand des Kriterienkatalogs wurde ein Datenblatt erstellt.

Die Beschreibungen befinden sich in Kapitel 4.5 und die zugehörigen Datenblätter können den Anhängen 5 bis 7 entnommen werden.

4.3.3 Vergleich der Enterprise-Suchmaschinen

Im letzten Schritt wurden die zuvor erstellten Beschreibungen und Datenblätter anhand des Kriterienkatalogs miteinander verglichen. Dazu wurden zunächst die einzelnen Dimensionen des Kriterienkatalogs betrachtet und dann Gemeinsamkeiten, Unterschiede und Alleinstellungsmerkmale der Enterprise-Suchmaschinen herausgearbeitet.

Das Ergebnis des Vergleichs wird in Kapitel 4.6 dargestellt.

4.4 Beschreibung des Kriterienkatalogs

Der Kriterienkatalog besteht aus den folgenden sechs Dimensionen:

- Basisinformationen,
- Systemarchitektur,
- Datenhaltung,
- Indexierung,
- Matching und
- Ergebnispräsentation.

Die einzelnen Elemente dieser Dimensionen werden im Folgenden in Tabellenform beschrieben.

4.4.1 Dimension „Basisinformationen“

Tabelle 3: Beschreibung der Elemente in der Dimension „Basisinformationen“

Hersteller	Wer ist der Hersteller der Enterprise-Suchmaschine?
Aktuelle Version	Welche Version der Enterprise-Suchmaschine lag am 05.01.2017 vor? Diese Information kann im Vergleich zur untersuchten Version einen Rückschluss auf den Aktualisierungszyklus geben.
Untersuchte Version	Welche Version der Enterprise-Suchmaschine wurde in dieser Arbeit untersucht?
Markteinführung	Seit welchem Jahr ist die Enterprise-Suchmaschine auf dem Markt?
Lizenz	Unter welcher Open Source-Lizenz steht die Enterprise-Suchmaschine? ¹¹⁷

4.4.2 Dimension „Systemarchitektur“

Tabelle 4: Beschreibung der Elemente in der Dimension „Systemarchitektur“

Typ	Was für eine Art von Software ist die Enterprise-Suchmaschine?
Unterbau	Auf welcher anderen Software baut die Enterprise-Suchmaschine ggf. auf?
Programmiersprache	In welcher Programmiersprache wurde die Enterprise-Suchmaschine programmiert?
Schnittstellen	Für welche Programmiersprachen existieren Schnittstellen zur Enterprise-Suchmaschine? Davon ist abhängig, ob diese in der eigenen Applikation (Katalogsuchmaschine) verwendet werden kann bzw. wie aufwändig dies ist.
Betriebssystem	Unter welchem Serverbetriebssystem kann die Enterprise-Suchmaschine betrieben werden?

¹¹⁷ Für einen Überblick zu Open Source-Lizenzen vgl. Lang 2014.

4.4.3 Dimension „Datenhaltung“

Tabelle 5: Beschreibung der Elemente in der Dimension „Datenhaltung“

DBMS	Auf welchem Typ von Datenbankmanagementsystem (DBMS) basiert die Datenhaltung innerhalb der Enterprise-Suchmaschine?
invertierte Liste	Arbeitet die Enterprise-Suchmaschine mit einer invertierten Liste?
Datenformat	Mit welchem Datenformat arbeitet die Enterprise-Suchmaschine?
Feldstruktur	Sind die Daten innerhalb der Enterprise-Suchmaschine in einer Feldstruktur abgelegt?
Aktualisierung	Ist es möglich, Datensätze innerhalb des Index zu aktualisieren?
Löschung	Ist es möglich, Datensätze innerhalb des Index zu löschen?
Größe	Welche Anzahl an Datensätzen kann im Index gespeichert werden?

4.4.4 Dimension „Indexierung“

Tabelle 6: Beschreibung der Elemente in der Dimension „Indexierung“

Indexer	Was für einen Indexer setzt die Enterprise-Suchmaschine ein?
Stemming	Welche Formen des Stemming beherrscht der Indexer?
Stoppwörter	Kann der Indexer Stoppwörter von der Indexierung ausschließen?
Synonyme	Welche Formen der Synonymbehandlung beherrscht der Indexer?

4.4.5 Dimension „Matching“

Tabelle 7: Beschreibung der Elemente in der Dimension „Matching“

Query Parser	Welche Query Parser stehen für die Enterprise-Suchmaschine zur Verfügung? Der Query Parser ist die Komponente, welche die Suchanfragen entgegen nimmt.
Rechtschreibkorrektur	Welche Formen der Rechtschreibkorrektur kennt die Enterprise-Suchmaschine?
Autovervollständigung¹¹⁸	Welche Formen der Autovervollständigung kennt die Enterprise-Suchmaschine?

4.4.6 Dimension „Ergebnispräsentation“

Tabelle 8: Beschreibung der Elemente in der Dimension „Ergebnispräsentation“

Ranking	Welche Formen des Ranking sind innerhalb der Trefferliste möglich?
IR-Modell	Auf welchem Information Retrieval-Modell (IR-Modell) basiert das Ranking innerhalb der Trefferliste?
Boosting	Welche Möglichkeiten zum Boosting innerhalb der Trefferliste bietet die Enterprise-Suchmaschine?
Highlighting	Können die Suchbegriffe innerhalb der Trefferliste hervorgehoben dargestellt werden?
Facetten	Wird eine facetthierarchische Suche angeboten?
Relevance Feedback¹¹⁹	Wird eine Suche nach ähnlichen Treffern angeboten?

4.5 Beschreibung der Enterprise-Suchmaschinen

4.5.1 Elasticsearch¹²⁰

Elasticsearch ist ein in der Programmiersprache Java geschriebener Webservice und baut auf der Softwarebibliothek Lucene auf. Die erste Version von Elasticsearch wurde von Shay Banon entwickelt und im Jahr 2010 veröffent-

¹¹⁸ Vgl. dazu auch Lewandowski/Quirnbach 2013, S. 278 f.

¹¹⁹ Vgl. dazu auch Lewandowski 2005, S. 151 f.

¹²⁰ Ein Datenblatt zu Elasticsearch befindet sich in Anhang 5.

licht.¹²¹ Neben einer Schnittstelle für Java gibt es eine REST-Schnittstelle (Representational State Transfer) für verschiedene weitere Programmiersprachen.¹²²

Elasticsearch speichert alle Objekte dokumentorientiert im Format JSON (JavaScript Object Notation).¹²³ Die Speicherung der Objekte erfolgt in einem Index in Form einer invertierten Liste.¹²⁴ Ein Index besteht dabei aus einzelnen Instanzen von Lucene (Shards).¹²⁵ Dokumente können aktualisiert und gelöscht werden, wobei bei einer Aktualisierung die bestehende Version als gelöscht markiert wird.¹²⁶

Die Generierung von Indextermen geschieht mittels eines sogenannten Analyzers. Dieser vereint drei Funktionen: Zeichenfilter, Tokenizer und Token-Filter. Diese Funktionen werden bei der Indexierung und der Suche durchlaufen.¹²⁷ Neben einem sprachunabhängigen Standardanalyzer¹²⁸ verfügt Elasticsearch über sprachbezogene Analyzer¹²⁹. Der Vorgang des Stemmings¹³⁰ ist in Elasticsearch ebenso als Token Filter realisiert wie die Behandlung von Stoppwörtern¹³¹ und Synonymen.¹³²

¹²¹ Vgl. Elasticsearch Guide 2.x, D. 1. Der Elasticsearch Guide 2.x und die Elasticsearch Reference 2.3 bestehen aus Einzeldokumenten. Alle verwendeten Dokumente wurden durchnummeriert und werden mit einem "D." für "Dokument" und ihrer Nummer referenziert. Der zugehörige Einzelnachweis kann dem Dokumentationsverzeichnis entnommen werden.

¹²² Vgl. Elasticsearch Guide 2.x, D. 2.

¹²³ Vgl. Elasticsearch Guide 2.x, D. 4.

¹²⁴ Vgl. Elasticsearch Guide 2.x, D. 5.

¹²⁵ Vgl. Elasticsearch Guide 2.x, D. 9.

¹²⁶ Vgl. Elasticsearch Guide 2.x, D. 10.

¹²⁷ Vgl. Elasticsearch Guide 2.x, D. 11.

¹²⁸ Vgl. Elasticsearch Guide 2.x, D. 20.

¹²⁹ Vgl. Elasticsearch Guide 2.x, D. 19.

¹³⁰ Vgl. Elasticsearch Guide 2.x, D. 21 u. D. 22.

¹³¹ Vgl. Elasticsearch Guide 2.x, D. 23.

¹³² Vgl. Elasticsearch Guide 2.x, D. 26.

Eine Suchanfrage an Elasticsearch wird über eine Abfragesprache namens Query DSL (Domain Specific Language) gestellt.¹³³ Die Funktionen für die Trefferliste umfassen die Hervorhebung von Suchtermen (Highlighting)¹³⁴, Facetten¹³⁵, verschiedene Formen von Suchvorschlägen¹³⁶ und eine Möglichkeit zum Boosting nach Popularität¹³⁷. Die Ermittlung der Relevanz eines Treffers erfolgt wahlweise mit einer Kombination aus Booleschem Retrieval und Vektorraummodell¹³⁸ oder mit dem probabilistischen Modell¹³⁹.

4.5.2 Solr¹⁴⁰

Solr ist ein in der Programmiersprache Java¹⁴¹ geschriebener Webservice und baut auf der Softwarebibliothek Lucene auf.¹⁴² Solr wurde im Jahr 2004 von Yonik Seeley bei CNET Networks entwickelt. 2006 wurde der Programmcode von Solr an die Apache Software Foundation übergeben.¹⁴³ Neben einer REST-Schnittstelle verfügt Solr über Schnittstellen zu verschiedenen Programmiersprachen.¹⁴⁴

Solr speichert alle Objekte dokumentorientiert¹⁴⁵ in einem Index in Form einer invertierten Liste.¹⁴⁶ Das primär genutzte Datenformat ist XML, Solr beherrscht nativ zusätzlich CSV und JSON.¹⁴⁷ Ein Solr-Index kann über mehrere Server

¹³³ Vgl. Elasticsearch Guide 2.x, D. 6.

¹³⁴ Vgl. Elasticsearch Guide 2.x, D. 7.

¹³⁵ Vgl. Elasticsearch Guide 2.x, D. 8.

¹³⁶ Vgl. Elasticsearch Guide 2.x, D. 27, D. 14 u. D. 28.

¹³⁷ Vgl. Elasticsearch Guide 2.x, D. 24.

¹³⁸ Vgl. Elasticsearch Guide 2.x, D. 15.

¹³⁹ Vgl. Elasticsearch Guide 2.x, D. 18.

¹⁴⁰ Ein Datenblatt zu Solr befindet sich in Anhang 6.

¹⁴¹ Vgl. Apache Solr Reference Guide 6.1, S. 5.

¹⁴² Vgl. Apache Solr Reference Guide 6.1, S. 12.

¹⁴³ Vgl. Apache Solr 2016b.

¹⁴⁴ Vgl. Apache Solr Reference Guide 6.1, S. 681 f.

¹⁴⁵ Vgl. Apache Solr Reference Guide 6.1, S. 48.

¹⁴⁶ Vgl. Apache Solr Reference Guide 6.1, S. 95.

¹⁴⁷ Vgl. Apache Solr Reference Guide 6.1, S. 12.

verteilt werden (Sharding).¹⁴⁸ Dokumente können aktualisiert und gelöscht werden, wobei bei einer Aktualisierung die ursprüngliche Version überschrieben wird.¹⁴⁹

Die Generierung von Indextermen geschieht mittels eines sogenannten Analyzers. Dieser besteht aus sogenannten Tokenizern und Filtern. Ein Tokenizer unterteilt einen Text in lexikalische Einheiten, die Token genannt werden. Ein Filter sorgt dann für die Transformation des Tokens in einen Indexterm.¹⁵⁰ Ein solcher Analyseprozess findet sowohl bei der Indexierung als auch bei der Suche statt.¹⁵¹ Der Vorgang des Stemming¹⁵² ist in Solr ebenso als Filter realisiert wie die Behandlung von Stoppwörtern¹⁵³ und Synonymen¹⁵⁴.

Die drei wichtigsten Query Parser von Solr sind der auf dem Lucene Query Parser aufbauende Standard Query Parser sowie der DisMax Query Parser und der Extendend DisMax Query Parser.¹⁵⁵

Die Funktionen für die Ergebnispräsentation umfassen die Hervorhebung von Suchtermen (Highlighting), Facetten, verschiedene Formen von Suchvorschlägen und die Möglichkeit zur Bevorzugung bestimmter Treffer (Query Elevation Component).¹⁵⁶ Die Ermittlung der Relevanz eines Treffers erfolgt wahlweise mit einer Kombination aus Booleschem Retrieval und Vektorraummodell¹⁵⁷ oder mit dem probabilistischen Modell¹⁵⁸.

¹⁴⁸ Vgl. Apache Solr Reference Guide 6.1, S. 13.

¹⁴⁹ Vgl. Apache Solr Reference Guide 6.1, S. 184 f.

¹⁵⁰ Vgl. Apache Solr Reference Guide 6.1, S. 104 ff.

¹⁵¹ Vgl. Apache Solr Reference Guide 6.1, S. 106.

¹⁵² Vgl. Apache Solr Reference Guide 6.1, S. 108.

¹⁵³ Vgl. Apache Solr Reference Guide 6.1, S. 136.

¹⁵⁴ Vgl. Apache Solr Reference Guide 6.1, S. 138 f.

¹⁵⁵ Vgl. Apache Solr Reference Guide 6.1, S. 243 ff.

¹⁵⁶ Vgl. Apache Solr Reference Guide 6.1, S. 242.

¹⁵⁷ Vgl. Grainger/Potter 2014, S. 65 ff.

¹⁵⁸ Vgl. Apache Solr Reference Guide 6.1, S. 71.

4.5.3 Xapian¹⁵⁹

Xapian ist eine im Jahr 2001 aus einem anderen Softwareprojekt hervorgegangene Softwarebibliothek.¹⁶⁰ Xapian wurde in der Programmiersprache C++ geschrieben und verfügt über Schnittstellen zu weiteren Programmiersprachen.¹⁶¹

Xapian speichert alle Objekte dokumentorientiert in einem Index in Form einer invertierten Liste. Die Speicherung erfolgt dabei in einem proprietären Format.¹⁶² Dokumente können aktualisiert oder gelöscht werden.¹⁶³

Die Generierung von Indextermen erfolgt durch einen Termgenerator. Dieser beherrscht Stemming und Stoppwortbehandlung. Zusätzlich können mithilfe von Präfixen bestimmte Terme feldbezogen erfasst werden.¹⁶⁴

Eine Suchanfrage an den Query Parser¹⁶⁵ von Xapian kann um Synonyme aus einem Wörterbuch oder Thesaurus erweitert werden.¹⁶⁶

Die Funktionen für die Trefferliste umfassen Facetten¹⁶⁷ und verschiedene Formen von Suchvorschlägen¹⁶⁸. Die Ermittlung der Relevanz eines Treffers erfolgt nach einem probabilistischen Modell.¹⁶⁹

4.6 Vergleich der Enterprise-Suchmaschinen

In diesem Abschnitt werden zunächst die Ergebnisse für die einzelnen Dimensionen des Kriterienkatalogs vorgestellt. Anschließend werden die dabei her-

¹⁵⁹ Ein Datenblatt zu Xapian befindet sich in Anhang 7.

¹⁶⁰ Vgl. [Xapian 02].

¹⁶¹ Vgl. [Xapian 01].

¹⁶² Vgl. Getting Started with Xapian 1.2, S. 10.

¹⁶³ Vgl. Getting Started with Xapian 1.2, S. 26.

¹⁶⁴ Vgl. Getting Started with Xapian 1.2, S. 13 f.

¹⁶⁵ Vgl. Getting Started with Xapian 1.2, S. 18 ff.

¹⁶⁶ Vgl. Getting Started with Xapian 1.2, S. 54 f.

¹⁶⁷ Vgl. Getting Started with Xapian 1.2, S. 42 ff.

¹⁶⁸ Vgl. Getting Started with Xapian 1.2, S. 20 f. u. S. 51 f.

¹⁶⁹ Vgl. Getting Started with Xapian 1.2, S. 56 f.

ausgearbeiteten Gemeinsamkeiten, Unterschiede und Alleinstellungsmerkmale zusammengefasst.

4.6.1 Dimension „Basisinformationen“

Hinter Elasticsearch (Unternehmen) und Solr (Stiftung) steht jeweils eine Organisation als Hersteller, während hinter Xapian die Community der beteiligten Entwickler steht. Den für diesen Vergleich untersuchten Versionen sind bei allen Enterprise-Suchmaschinen bereits neuere Versionen¹⁷⁰ gefolgt. Solr und Xapian sind deutlich länger am Markt als Elasticsearch. Alle Systeme beruhen auf gängigen Open Source-Lizenzen. Einen Überblick über die Ergebnisse der Dimension „Basisinformationen“ liefert Tabelle 9.

Tabelle 9: Vergleich der Dimension „Basisinformationen“

	Elasticsearch	Solr	Xapian
Hersteller	Elastic	Apache Software Foundation	xapian.org
Aktuelle Version (Stand: 05.01.2017)	5.1.1 vom 08.12.2016	6.3.0 vom 08.11.2016	1.4.2 vom 26.12.2016
Untersuchte Version	2.3.5 vom 03.08.2016	6.1.0 vom 17.06.2016	1.4.0 vom 25.06.2016
Markteinführung	2010	2004	2001
Lizenz	Apache	Apache	GPL

4.6.2 Dimension „Systemarchitektur“

Elasticsearch und Solr sind in Java programmierte Webservices und basieren auf der Softwarebibliothek Lucene. Xapian ist selbst eine Softwarebibliothek und wurde in C++ programmiert. Eine Schnittstelle zu den Programmiersprachen Java, JavaScript, PHP, Perl, Python und Ruby ist für alle Enterprise-Suchmaschinen vorhanden. Einen Überblick über die Ergebnisse der Dimension „Systemarchitektur“ liefert Tabelle 10.

¹⁷⁰ Bei Elasticsearch folgte auf die Version 2.4.x direkt die Version 5.0.x, vgl. Elasticsearch 2016.

Tabelle 10: Vergleich der Dimension „Systemarchitektur“

	Elasticsearch	Solr	Xapian
Typ	Webservice (REST)	Webservice (REST)	Softwarebibliothek
Unterbau	Lucene	Lucene	N/A
Programmiersprache	Java	Java	C++
Schnittstellen	Groovy, Java, JavaScript, .NET, Perl, PHP, Python, Ruby	AJAX, C#, ColdFusion, Forrest/Cocoon, Java, JavaScript, .NET, Perl, PHP, Python, Rails, Ruby	C#, Erlang, Java, JavaScript (Node.js), Lua, Perl, Python, PHP, Ruby, Tcl
Betriebssystem	plattformübergreifend	plattformübergreifend	plattformübergreifend

4.6.3 Dimension „Datenhaltung“

Alle Enterprise-Suchmaschinen speichern ihre Objekte dokumentorientiert¹⁷¹ in einem Index in Form einer invertierten Liste. Bei allen ist die Aktualisierung und Löschung von Dokumenten möglich. Bei Elasticsearch und Solr ist das jeweilige primäre Datenformat ein Standardformat mit einer Feldstruktur, während Xapian ein proprietäres Datenformat mit einer indirekt durch Präfixe erzeugbaren Feldstruktur nutzt. Die Indexgröße von Elasticsearch und Solr ist theoretisch unendlich, während die des Xapian-Index begrenzt ist. Einen Überblick über die Ergebnisse der Dimension „Datenhaltung“ liefert Tabelle 11.

¹⁷¹ Zu dokumentorientierten Datenbanksystemen vgl. Störl 2015, S. 372 ff.

Tabelle 11: Vergleich der Dimension „Datenhaltung“

	Elasticsearch	Solr	Xapian
DBMS	dokumentorientiert	dokumentorientiert	dokumentorientiert
invertierte Liste	ja	ja	ja
Datenformat	JSON	XML (primär), CSV, JSON	proprietär (Chert, Glass)
Feldstruktur	ja	ja	ja (durch Präfixe)
Aktualisierung	ja (alte Version wird gelöscht)	ja (alte Version wird ersetzt)	ja
Löschung	ja (wird deaktiviert)	ja	ja
Größe	unendlich (Sharding)	unendlich (Sharding)	ca. 4,3 Milliarden Datensätze

4.6.4 Dimension „Indexierung“

Elasticsearch und Solr beherrschen technisch grundsätzlich identische Indexierungsmethoden.¹⁷² Der Termgenerator von Xapian ist dem gegenüber deutlich einfacher aufgebaut. Einen Überblick über die Ergebnisse der Dimension „Systemarchitektur“ liefert Tabelle 12.

Tabelle 12: Vergleich der Dimension „Indexierung“

	Elasticsearch	Solr	Xapian
Indexer	Analyzer (Tokenizer, Filter)	Analyzer (Tokenizer, Filter)	Termgenerator
Stemming	algorithmisch, wörterbuchbasiert	algorithmisch, wörterbuchbasiert	ja
Stoppwörter	ja	ja	ja
Synonyme	einfach, mehrfach	einfach, mehrfach	Wörterbuch, Thesaurus

¹⁷² Die Anzahl der jeweils angebotenen Analyzer, Tokenizer und Filter unterscheidet sich, vgl. Elasticsearch Guide 2.x, D. 15 bzw. Apache Solr Reference Guide 6.1, S.104 ff. Diese Unterschiede wurden nicht näher untersucht.

4.6.5 Dimension „Matching“

Für Solr stehen mehr Query Parser als für Elasticsearch und Xapian zur Verfügung. Bei der Rechtschreibkorrektur und der Autovervollständigung bietet nur Solr jeweils indexbasierte und wörterbuchbasierte Verfahren an. Xapian bietet beide Verfahren bei der Rechtschreibkorrektur an, während Elasticsearch nur indexbasierte Verfahren kennt. Einen Überblick über die Ergebnisse der Dimension „Systemarchitektur“ liefert Tabelle 13.

Tabelle 13: Vergleich der Dimension „Matching“

	Elasticsearch	Solr	Xapian
Query Parser	Query DSL	Standard, DisMax, Extended DisMax	Xapian QueryParser
Rechtschreibkorrektur	indexbasiert	indexbasiert, wörterbuchbasiert	indexbasiert, wörterbuchbasiert
Autovervollständigung	indexbasiert	indexbasiert, wörterbuchbasiert	indexbasiert

4.6.6 Dimension „Ergebnispräsentation“

Alle Enterprise-Suchmaschinen bieten grundsätzlich identische Möglichkeiten beim Ranking an, auch Facetten gehören bei allen zum Funktionsumfang. Elasticsearch und Solr setzen identische Retrieval-Modelle ein und beide bieten ein Highlighting von Suchtermen an. Letztere Funktion fehlt bei Xapian, das zudem allein auf einem probabilistischen Retrieval-Modell beruht. Xapian hat kein Relevance Feedback in seinem Funktionsumfang. Größere Unterschiede zwischen den drei Enterprise-Suchmaschinen gibt es beim Boosting. Ein Boosting nach Popularität wird nur von Elasticsearch angeboten, eine Möglichkeit zum Boosting einzelner Treffer (Sponsored Search)¹⁷³ gibt es nur bei Solr. Einen Überblick über die Ergebnisse der Dimension „Systemarchitektur“ liefert Tabelle 14.

¹⁷³ Für eine kurze Erläuterung vgl. Apache Solr Reference Guide 6.1, S. 397.

Tabelle 14: Vergleich der Dimension „Ergebnispräsentation“

	Elasticsearch	Solr	Xapian
Ranking	Relevanz, Feldwerte; mehrstufig	Relevanz, Feldwerte; mehr- stufig	Relevanz, Feldwerte; mehrstufig
IR-Modell	Lucene (Boolean, Vektorraum) BM25 (probabilistisch)	Lucene (Boolean, Vektorraum) BM25 (probabilistisch)	BM25 (probabilistisch)
Boosting	Indexierung, Suche, Popularität	Indexierung, Suche, Sponsored Search	Suche
Highlighting	ja	ja	nein
Facetten	ja	ja	ja
Relevance Feedback	ja	ja	nein

4.6.7 Gemeinsamkeiten, Unterschiede, Alleinstellungsmerkmale

Bei der Betrachtung des Vergleichs fällt zunächst auf, dass Elasticsearch und Solr zueinander mehr Gemeinsamkeiten als Unterschiede aufweisen. Bei 22 von 29 Elementen¹⁷⁴ des Kriterienkatalogs ist eine Gemeinsamkeit von Elasticsearch und Solr erkennbar. Nur bei sieben Elementen sind Unterschiede zwischen beiden vorhanden. Dieses Ergebnis ist insofern nicht überraschend, da sowohl Elasticsearch als auch Solr auf der Softwarebibliothek Lucene aufbaut. Beide weisen mit Ausnahme der Schnittstellen eine identische Systemarchitektur und mit Ausnahme des Datenformats eine identische Datenhaltung auf. Die größten Unterschiede weisen Elasticsearch und Solr beim Matching und beim Boosting auf. Einen Überblick der Gemeinsamkeiten und Unterschiede zwischen Elasticsearch und Solr bietet Tabelle 15.

¹⁷⁴ Die Elemente „Aktuelle Version“ und „Untersuchte Version“ werden hier als ein Element behandelt.

Tabelle 15: Vergleich zwischen Elasticsearch und Solr

Gemeinsamkeiten	Unterschiede
Basisinformationen	
Hersteller	
Version	
	Markteinführung
Lizenz	
Systemarchitektur	
Typ	
Unterbau	
Programmiersprache	
	Schnittstellen
Betriebssystem	
Datenhaltung	
DBMS	
Invertierte Liste	
	Datenformat
Feldstruktur	
Aktualisierung	
Löschung	
Größe	
Indexierung	
Indexer	
Stemming	
Stoppwörter	
Synonyme	
Matching	
	Query Parser
	Rechtschreibkorrektur
	Autovervollständigung
Ergebnispräsentation	
Ranking	
IR-Modell	
	Boosting
Highlighting	
Facetten	
Relevance Feedback	

Tabelle 16: Xapian im Vergleich zu Elasticsearch und Solr

Gemeinsamkeiten	Unterschiede
Basisinformationen	
	Hersteller
Version	
	Markteinführung
Lizenz	
Systemarchitektur	
	Typ
	Unterbau
	Programmiersprache
	Schnittstellen
Betriebssystem	
Datenhaltung	
DBMS	
Invertierte Liste	
	Datenformat
Feldstruktur	
Aktualisierung	
Löschung	
	Größe
Indexierung	
	Indexer
	Stemming
Stoppwörter	
	Synonyme
Matching	
	Query Parser
	Rechtschreibkorrektur
	Autovervollständigung
Ergebnispräsentation	
Ranking	
	IR-Modell
	Boosting
	Highlighting
Facetten	
	Relevance Feedback

Xapian wiederum weist mehr Unterschiede als Gemeinsamkeiten zu Elasticsearch und Solr auf. Bei 18 von 29 Elementen des Kriterienkatalogs ist ein Unterschied von Xapian zu Elasticsearch und Solr erkennbar. Nur bei 11 Elementen gibt es eine Gemeinsamkeit. Die Ursache wird darin gesehen, dass Xapian eine Softwarebibliothek ist und nicht wie Elasticsearch und Solr auf einer Softwarebibliothek aufbaut. Neben der Systemarchitektur sind die meisten Unterschiede bei der Indexierung, beim Matching und bei der Ergebnispräsentation zu finden. Einen Überblick der Gemeinsamkeiten und Unterschiede von Xapian im Vergleich zu Elasticsearch und Solr bietet Tabelle 16.

Alleinstellungsmerkmale wurden nur beim Boosting gefunden. Ein Boosting nach Popularität gibt es nur bei Elasticsearch und eine Möglichkeit zum Boosting einzelner Treffer (Sponsored Search) nur bei Solr.

4.7 Installation der Enterprise-Suchmaschinen

Dieser Abschnitt beschreibt die vom Verfasser während seiner Vorüberlegungen bei der Installation der Enterprise-Suchmaschinen gemachten Erfahrungen. Die Installationen wurden nicht auf verschiedenen Systemen mehrfach bei identischem Ergebnis wiederholt, was im Sinne einer wissenschaftlichen Verwertbarkeit zwingend erforderlich gewesen wäre. Die folgende kurze Darstellung¹⁷⁵ ist daher als informative Ergänzung zu den oben beschriebenen Ergebnissen des Vergleichs gedacht.

Zunächst wurde für jede Enterprise-Suchmaschine mithilfe der Virtualisierungssoftware VirtualBox eine Virtuelle Maschine mit der Linux-Distribution Xubuntu 16.04.1 LTS als Betriebssystem erstellt.

Elasticsearch wurde gemäß der Anleitung aus der offiziellen Dokumentation installiert und gestartet, war allerdings nicht wie darin beschrieben im Terminal erreichbar.¹⁷⁶

¹⁷⁵ Eine ausführlichere Darstellung befindet sich in Anhang 8.

¹⁷⁶ Vgl. Elasticsearch Reference 2.3, D. 3.

Solr wurde gemäß der Installationsanleitung installiert und die Adminkonsole war wie angegeben über den Browser erreichbar.¹⁷⁷

Zusätzlich zu Xapian wurde die auf Xapian aufbauende Suchapplikation Omega installiert. Da nur eine veraltete Installationsanleitung in der offiziellen Dokumentation verfügbar war und in dieser verschiedene inzwischen benötigte Systemkomponenten nicht aufgeführt waren, musste die Installation mehrfach wiederholt werden. Allerdings war Omega anschließend nicht wie angegeben über den Browser erreichbar.¹⁷⁸

4.8 Fazit

Die Forschungsfrage II konnte beantwortet werden. Es wurde ein Vergleich der Enterprise-Suchmaschinen Elasticsearch, Solr und Xapian durchgeführt und deren Gemeinsamkeiten, Unterschiede und Alleinstellungsmerkmale herausgearbeitet. Als Ergebnis wurde festgestellt, dass Elasticsearch und Solr zueinander mehr Gemeinsamkeiten als Unterschiede aufweisen. Xapian wiederum weist mehr Unterschiede als Gemeinsamkeiten zu Elasticsearch und Solr auf. Alleinstellungsmerkmale gibt es nur im Bereich des Boosting.

Der für die Untersuchung erstellte und verwendete Kriterienkatalog kann grundsätzlich auch dazu genutzt werden, um weitere Enterprise-Suchmaschinen in den Vergleich einzubeziehen. Ein möglicher Ansatz dazu könnte darin bestehen, den in diesem Kapitel durchgeführten Vergleich um die von Baeza-Yates, Ribeiro-Neto und Middleton¹⁷⁹ untersuchten Enterprise-Suchmaschinen zu ergänzen. Dadurch könnte ermittelt werden, ob es weitere Enterprise-Suchmaschinen gibt, die als technische Basis für die Entwicklung einer Katalogsuchmaschine geeignet sind.

¹⁷⁷ Vgl. Apache Solr Reference Guide 6.1, S. 6 ff.

¹⁷⁸ Vgl. [Xapian 03].

¹⁷⁹ Vgl. Baeza-Yates/Ribeiro-Neto/Middleton 2011.

Bei der Wahl der Methode für die Untersuchung in diesem Kapitel wurden Funktionstests ausgeschlossen, weil erst die zu untersuchenden Funktionen bekannt sein müssen. Die im Kriterienkatalog festgehaltenen Funktionen könnten nun in einem nächsten Schritt die Basis für einen Funktionstest bilden. Ergänzend dazu könnte ein Retrievaltest durchgeführt werden. Im Rahmen dieser Arbeit hat der Verfasser aufgrund seiner negativen Erfahrungen bei der Installation und im Hinblick auf die zeitlichen Rahmenbedingungen auf Funktions- und Retrievaltests verzichtet.

5 Fazit und Ausblick

Das Ziel dieser Arbeit war es, einen Vergleich von Enterprise-Suchmaschinen durchzuführen, die als technische Basis bibliothekarischer Katalogsuchmaschinen eingesetzt werden. Aus diesem Ziel ergaben sich zwei Forschungsfragen, die beide im Rahmen dieser Arbeit beantwortet werden konnten. Das Ziel dieser Arbeit wurde somit erreicht.

Die Forschungsfrage I lautete: Welche Enterprise-Suchmaschinen werden als technische Basis bibliothekarischer Katalogsuchmaschinen verwendet? Zur Beantwortung dieser Frage wurden die Verbundkataloge, die ASP-Produkte der Verbundzentralen und die Online-Kataloge von relevanten Bibliotheken untersucht. Bei zwölf der 59 untersuchten Online-Kataloge handelt es sich um Katalogsuchmaschinen, die von der jeweiligen Institution selbst entwickelt wurden. Die technische Basis von neun dieser Eigenentwicklungen (75 Prozent) ist die Enterprise-Suchmaschine Solr. Von den übrigen Eigenentwicklungen basiert jeweils eine auf den Enterprise-Suchmaschinen CIXbase, Elasticsearch und Xapian.

Die zur Beantwortung der Forschungsfrage I durchgeführte Untersuchung könnte durch eine vollständige Erhebung für alle Bibliothekstypen sinnvoll fortgeführt werden. Würde eine solche Erhebung regelmäßig wiederholt, wäre zudem die Beobachtung von Langzeitentwicklungen möglich.

Darüber hinaus könnten die Entscheidungsprozesse von Bibliotheken bei der Auswahl eines Suchsystems systematisch untersucht werden. Dadurch könnten etwa die in dieser Arbeit ermittelten Unterschiede bei der Nutzung von Discovery Systemen und Katalogsuchmaschinen durch Wissenschaftliche und Öffentliche Bibliotheken erklärt werden. Auch könnte das beobachtete Vorkommen des OPACs als Zweitkatalog weiter untersucht werden.

Die Forschungsfrage II lautete: Welche Funktionen bieten die als technische Basis bibliothekarischer Katalogsuchmaschinen eingesetzten Enterprise-Suchmaschinen? Wo gibt es Gemeinsamkeiten, Unterschiede und Alleinstellungs-

merkmale? Zur Beantwortung dieser Frage wurden die offiziellen Dokumentationen der als Ergebnis der Forschungsfrage I ermittelten Enterprise-Suchmaschinen anhand eines Kriterienkatalogs untersucht (mit Ausnahme von CIXbase¹⁸⁰). Die verwendeten Kriterien wurden aus den Kennzeichen von Katalogsuchmaschinen¹⁸¹ gebildet. Festgestellt wurde, dass Elasticsearch und Solr zueinander mehr Gemeinsamkeiten als Unterschiede aufweisen. Xapian wiederum weist mehr Unterschiede als Gemeinsamkeiten zu Elasticsearch und Solr auf. Alleinstellungsmerkmale gibt es nur im Bereich des Boosting.

Der in dieser Arbeit erstellte Kriterienkatalog könnte in künftigen Untersuchungen dazu eingesetzt werden, um weitere Enterprise-Suchmaschinen in den Vergleich einzubeziehen. So könnten weitere Enterprise-Suchmaschinen auf ihre Eignung als technische Basis für die Entwicklung einer Katalogsuchmaschine überprüft werden.

Eine systematische Aufarbeitung der Entstehungsprozesse der selbst entwickelten Katalogsuchmaschinen stellt ebenfalls einen lohnenswerten Gegenstand für die weitere Forschung dar. Insgesamt ist wenig über die Motive für die Entscheidung einer Bibliothek oder Verbundzentrale bekannt, eine bestimmte Enterprise-Suchmaschine für die Entwicklung der eigenen Katalogsuchmaschine einzusetzen. Nur in wenigen Anwenderberichten¹⁸² wird näher auf die gewählte Enterprise-Suchmaschine eingegangen. So wird etwa die Bedeutung des Sharding von Solr für die getrennte Durchführung von Indexierung und Suche¹⁸³ oder die kooperative Nutzung von Artikelindizes¹⁸⁴ genannt. Außerdem wird die einfache Parametrierbarkeit von Solr als Erfolgsfaktor hervorgehoben.¹⁸⁵ Diese wenigen Hinweise bieten jedoch weder eine ausreichende Erklärung dafür, warum Solr als technische Basis von 75

¹⁸⁰ Die Projektwebseite von CIXbase stand nicht mehr zu Verfügung, vgl. Kap. 4.2.1.

¹⁸¹ Vgl. Kap. 2.3.

¹⁸² Vgl. Kap. 3.2.2.

¹⁸³ Vgl. Bertelmann/Szott/Höhnnow 2012, S. 373.

¹⁸⁴ Vgl. Lohmeier/Seige 2015, S. 150 f.

¹⁸⁵ Vgl. Maylein/Langenstein 2013, S. 192.

Prozent der in dieser Arbeit als Eigenentwicklung identifizierten Katalogsuchmaschinen eingesetzt wird noch warum die übrigen Enterprise-Suchmaschinen nur in einzelnen Katalogsuchmaschinen verwendet werden.

Ein weiterer lohnenswerter Gegenstand für die weitere Forschung ist die Untersuchung weiterer Funktionen von Enterprise-Suchmaschinen. Durch welche dieser Funktionen könnte der Funktionsumfang von Katalogsuchmaschinen sinnvoll erweitert werden? Welche dieser Funktionen könnten nützlich für die Erschaffung neuer bibliothekarischer Dienstleistungen sein?

Denkbar ist etwa die Nutzung einer Enterprise-Suchmaschine zur Datenanalyse von Forschungsdaten. So könnten die Forschungsdaten nicht nur in Katalogsuchmaschinen oder Repositorien verzeichnet und aufbewahrt, sondern dank einer Enterprise-Suchmaschine als technischer Basis weiter verarbeitet und erforscht werden. Bibliotheken bietet sich dadurch die Chance, einen Mehrwert für die Wissenschaft zu erzielen.

Literaturverzeichnis

Das letzte Abrufdatum aller im Literaturverzeichnis genannten Internet-Dokumente ist in runden Klammern angegeben.

Antelman/Lynema/Pace 2006:

Antelman, Kristen; Lynema, Emily; Pace, Andrew K.: Toward a Twenty-First Century Catalog. In: Information Technology and Libraries 25 (2006) 3, S. 128-139. <http://dx.doi.org/10.6017/ital.v25i3.3342> (26.05.2016).

Apache Solr 2016a:

Apache Solr. Version vom 26.07.2016. In Wikipedia. https://en.wikipedia.org/wiki/Apache_Solr (23.08.2016).

Apache Solr 2016b:

Apache Solr. Version vom 09.11.2016. In Wikipedia. https://en.wikipedia.org/wiki/Apache_Solr (05.01.2017).

Arbeitsgemeinschaft der Verbundsysteme 2013:

Arbeitsgemeinschaft der Verbundsysteme: Service-Partner für Bibliotheken. 2013. http://www.dnb.de/SharedDocs/Downloads/DE/DNB/wir/agVerbundBroschuere2012.pdf?__blob=publicationFile (24.09.2016).

Baeza-Yates/Ribeiro-Neto/Middleton 2011:

Baeza-Yates, Ricardo; Ribeiro-Neto, Berthier; Middleton, Christian: Open Source Search Engines. In: Modern Information Retrieval. Hrsg. von Ricardo Baeza-Yates und Berthier Ribeiro-Neto. 2. Aufl. Harlow: Pearson, 2011, S. 737-754.

Bahrs 2009:

Bahrs, Julian: Enterprise Search - Suchmaschinen für Inhalte in Unternehmen. In: Nutzerorientierung in Wissenschaft und Praxis. Hrsg. von Dirk Lewandowski. Heidelberg: AKA Akad. Verl.-Ges., 2009. (Handbuch Internet-Suchmaschinen, 1), S. 329-355.

Baron 2016:

Baron, Christine: Suchmaschinentechnologie im hbz. In: Fakten und Perspektiven 2015. Jahresbericht. Hrsg. von Hochschulbibliothekszenrum des Landes Nordrhein-Westfalen. Köln, 2016, S. 9-12. https://hbz.opus.hbz-nrw.de/files/369/PDFA_Jahresbericht_hbz_2015_web.pdf (16.07.2016).

Beckmann 2016:

Beckmann, Regine: Protokoll der 65. Sitzung des Fachbeirates des GBV. Göttingen: Gemeinsamer Bibliotheksverbund, 18.04.2016. https://verbundwiki.gbv.de/download/attachments/884746/GBV-Fachbeirat_65_Protokoll_18_03_2016.pdf (11.07.2016).

Bertelmann/Szott/Höhnnow 2012:

Bertelmann, Roland; Szott, Sascha; Höhnnow, Tobias: Discovery jenseits von "all you can eat" und "one size fits all". In: Bibliothek 36 (2012) 3, S. 369-376.

[BibDIA 01]:

BibDIA. Referenzen. <http://www.bibdia.de/referenzen> (18.07.2016).

Blenkle/Ellis/Haake 2009:

Blenkle, Martin; Ellis, Rachel; Haake, Elmar: E-LIB Bremen – Automatische Empfehlungsdienste für Fachdatenbanken im Bibliothekskatalog / Metadatenpools als Wissensbasis für bestandsunabhängige Services. In: Bibliotheksdienst 43 (2009) 6, S. 618-625.

Blenkle et. al. 2015:

Blenkle, Martin; Ellis, Rachel; Haake, Elmar; Zillmann, Hartmut: Nur die ersten Drei zählen! In: o-bib 2 (2015) 2, S. 33-42. <http://dx.doi.org/10.5282/o-bib/2015H2S33-42> (01.08.2016).

Borst 2016:

Borst, Timo: Softwareentwicklung in Bibliotheken: Von Open Source bis „Not invented here“. In: ZBW MediaTalk, 03.08.2016. <https://www.zbw-mediataalk.eu/2016/08/softwareentwicklung-in-bibliotheken-von-open-source-bis-not-invented-here/> (10.08.2016).

Caplan 2012:

Caplan, Priscilla: On discovery tools, OPACs and the motion of library language. In: Library Hi Tech 30 (2012) 1, S. 108-115.

Christensen 2010:

Christensen, Anne: Katalog 2.0 im Eigenbau: Das beluga-Projekt der Hamburger Bibliotheken. In: Handbuch Bibliothek 2.0. Hrsg. von Julia Bergmann und Patrick Danowski. Berlin: De Gruyter Saur, 2010 (Bibliothekspraxis, 41), S. 317-332.

Christensen 2013:

Christensen, Anne: Next-generation catalogues: what do users think? In: Catalogue 2.0: The Future of the Library Catalogue. Hrsg. von Sally Chambers. London: Facet Publ., 2013, S. 1-15.

[DB-Engines 01]:

DB-Engines Ranking von Suchmaschinen. <http://db-engines.com/de/ranking/suchmaschine> (21.01.2017).

Deutscher Bibliotheksverband 2016:

Deutscher Bibliotheksverband: Mitglieder. Fachgruppen / Sektionen / Sektion 1 / Mitglieder. Stand: 03.07.2016. <http://www.bibliotheksverband.de/fachgruppen/sektionen/sektion-1/mitglieder.html> (03.07.2016).

Dobratz 2007:

Dobratz, Susanne: Open-Source-Software zur Realisierung von Institutionellen Repositorien – Überblick. In: Zeitschrift für Bibliothekswesen und Bibliographie 54 (2007) 4-5, S. 199-206.

Drost-Fromm 2014:

Drost-Fromm, Isabel: Die Über-Suchmaschine. In: c't 32 (2014) 10, S. 154-157.

Elasticsearch 2016:

Elasticsearch. Version vom 30.12.2016. In: Wikipedia. <https://en.wikipedia.org/wiki/Elasticsearch> (05.01.2017).

[Ex Libris 01]:

Ex Libris. Primo. Anforderungen des Nutzers erfüllen. <http://www.exlibrisgroup.com/de/category/PrimoUeberblick> (18.07.2016).

Flimm 2007:

Flimm, Oliver: Die Open-Source-Software OpenBib an der USB Köln – Überblick und Entwicklungen in Richtung OPAC 2.0. In: Bibliothek 31 (2007) 2, S. 185-192.

Flimm 2010:

Flimm, Oliver (2010): Anreicherungen, Mashups und Vernetzungen von Titeln in einem heterogenen Katalogverbund am Beispiel des Kölner UniversitätsGesamtkatalogs KUG. In: Handbuch Bibliothek 2.0. Hrsg. von Julia Bergmann und Patrick Danowski. Berlin: De Gruyter Saur, 2010 (Bibliothekspraxis, 41), S. 293-315.

Fühles-Ubach 2013:

Fühles-Ubach, Simone: Quantitative Befragungen. In: Handbuch Methoden der Bibliotheks- und Informationswissenschaft. Hrsg. von Konrad Umlauf und Simone Fühles-Ubach. Berlin: De Gruyter Saur, 2013, S. 96-113.

Fuhr 2013:

Fuhr, Norbert: Modelle im Information Retrieval. In: Grundlagen der praktischen Information und Dokumentation. Hrsg. von Rainer Kuhlen, Wolfgang Semar und Dietmar Strauch. Berlin: De Gruyter Saur, 2013, S. 322-335.

Gantert 2016:

Gantert, Klaus: Bibliothekarisches Grundwissen. Berlin: De Gruyter Saur, 2016.

Gormley/Tong 2015:

Gormley, Clinton; Tong, Zachary: Elasticsearch: The Definitive Guide. Beijing: O'Reilly Media, 2015.

Grainger/Potter 2014:

Grainger, Trey; Potter, Timothy: Solr in action. Shelter Island: Manning, 2014.

Granitzer 2009:

Granitzer, Michael: Enterprise Search. In: Lexikon der Bibliotheks- und Informationswissenschaft. Hrsg. von Konrad Umlauf und Stefan Gradmann. Bd. 1. Stuttgart: Hiersemann, 2009, S. 272.

Hagenbruch/Heise 2014:

Hagenbruch, André; Heise, Miriam: "suchen" – ein Prototyp zum Vergleich von Discovery-Diensten. In: Bibliotheksdienst 48 (2014) 10, S. 816-827.

Haller 1998:

Haller, Klaus: Katalogkunde. Eine Einführung in die Formal- und Sacherschließung. 3., erw. Aufl. Aufl. München: Saur, 1998.

Hawking 2011:

Hawking, David: Enterprise Search. In: Modern Information Retrieval. Hrsg. von Ricardo Baeza-Yates und Berthier Ribeiro-Neto. 2. Aufl. Harlow: Pearson, 2011, S. 641-683.

[hzb 01]:

hzb. DBS - Deutsche Bibliotheksstatistik. variable Auswertung.
<https://www.bibliotheksstatistik.de/> (24.09.2016).

[HeBIS 01]:

HeBIS. Willkommen beim HeBIS Discovery System (HDS).
<https://www.hebis.de/de/1kataloge/hds.php> (21.07.2016).

Heller 2015:

Heller, Lambert: GetInfo, die hauseigene Discovery-Lösung der TIB, ist letzte Woche vollständig auf Open Source umgestiegen. Beitrag an die Mailingliste open-science-de vom 21.07.2015. <https://lists.okfn.org/pipermail/open-science-de/2015-July/000305.html> (13.07.2016).

Herrlich/Ledl/Tréfas 2013:

Herrlich, Bernhard; Ledl, Andreas; Tréfas, David: Editorial: Die Farbe des Katalogs. In: 027.7 Zeitschrift für Bibliothekskultur 1 (2013) 2, S. 43.
<http://dx.doi.org/10.12685/027.7-1-2-25> (05.03.2016).

Hohlfeld/Tobschall 2014:

Hohlfeld, Michael; Tobschall, Esther: EIN Portal der TIB (nicht nur) für Ingenieure, Naturwissenschaftler, Informatiker, Mathematiker und Architekten. In: Bibliothek 38 (2014) 1, S. 62-71.

Houser 2009:

Houser, John: The VuFind implementation at Villanova University. In: Library Hi Tech 27 (2009) 1, S. 93-105.

Jansen 2014:

Jansen, Heiko: Discovery-Services – Einführung, Marktübersicht und Trends. In: Bibliotheksdienst 48 (2014) 10, S. 773-783.

Jansen/Kemner-Heek/Schweitzer 2010:

Jansen, Heiko; Kemner-Heek, Christin; Schweitzer, Roswitha: Konkurrenzanalyse ausgewählter kommerzieller Suchindizes. Köln: Hochschulbibliothekszenrum des Landes Nordrhein-Westfalen, 2010. <https://www.hbz-nrw.de/dokumentencenter/veroeffentlichungen/suchindizes.pdf> (06.03.2016).

Kinstler 2013:

Kinstler, Till: Making search work for the library user. In: Catalogue 2.0: The Future of the Library Catalogie. Hrsg. von Sally Chambers. London: Facet Publ., 2013, S. 17-36.

Kneifel 2009:

Kneifel, Fabienne: Mit Web 2.0 zum Online-Katalog der nächsten Generation. Wiesbaden: Dinges & Frick, 2009.

Kneifel 2010:

Kneifel, Fabienne: Der Katalog 2.0: Mit Web 2.0 zum Online-Katalog der nächsten Generation. In: Handbuch Bibliothek 2.0. Hrsg. von Julia Bergmann und Patrick Danowski. Berlin: De Gruyter Saur, 2010. (Bibliothekspraxis, 41), S. 37-61.

Kostädt 2008:

Kostädt, Peter: Innovative Recherchemöglichkeiten in Katalogen und Bibliotheksportalen. In: Bibliotheken gestalten Zukunft. Hrsg. von Evelinde Hutzler, Albert Schröder und Gabriele Schweikl. Göttingen: Univ.-Verl. Göttingen, 2008, S. 101-113.

Kostädt 2015:

Kostädt, Peter: Suchportale, Discovery-Systeme und Linkresolver. In: Praxishandbuch Bibliotheksmanagement. Hrsg. von Rolf Griebel, Hildegard Schäffler und Konstanze Söllner. Berlin: De Gruyter Saur, 2015, S. 513-523.

Lang 2014:

Lang, Mirco: Nehmen und geben. In: c't (2014) 15, S. 144-149.

Lepsky 2013:

Lepsky, Klaus: Automatische Indexierung. In: Grundlagen der praktischen Information und Dokumentation. Hrsg. von Rainer Kuhlen, Wolfgang Semar und Dietmar Strauch. Berlin: De Gruyter Saur, 2013, S. 272-285.

Lewandowski 2005:

Lewandowski, Dirk: Web Information Retrieval. Technologien zur Informationssuche im Internet. Frankfurt am Main: DGI, 2005. (Reihe Informationswissenschaft der DGI, 7).

Lewandowski 2006:

Lewandowski, Dirk: Suchmaschinen als Konkurrenten der Bibliothekskataloge.: In: Zeitschrift für Bibliothekswesen und Bibliographie 53 (2006) 2, S. 71-78.

Lewandowski 2009:

Lewandowski, Dirk: Ranking library materials. In: Library Hi Tech 27 (2009) 4, S. 584-593.

Lewandowski 2013:

Lewandowski, Dirk: Suchmaschinen. In: Grundlagen der praktischen Information und Dokumentation. Hrsg. von Rainer Kuhlen, Wolfgang Semar und Dietmar Strauch. Berlin: De Gruyter Saur, 2013, S. 495-508.

Lewandowski/Quirnbach 2013:

Lewandowski, Dirk; Quirnbach, Sonja: Suchvorschläge während der Eingabe. In: Suchmaschinen zwischen Technik und Gesellschaft. Hrsg. von Dirk Lewandowski. Heidelberg: AKA Akad. Verl.-Ges., 2013. (Handbuch Internet-Suchmaschinen, 3), S. 273-298.

Lohmeier/Seige 2015:

Lohmeier, Felix; Seige, Leander: Kooperative Katalogentwicklung in Sachsen. In: BIS – Das Magazin der Bibliotheken in Sachsen 8 (2015) 3, S. 148-151.

Lützenkirchen/Falkenstein-Feldhoff 2014:

Lützenkirchen, Frank; Falkenstein-Feldhoff, Katrin: Seit Ende 2012 nutzt die UB Duisburg-Essen Primo inklusive Primo Central. In: ProLibris 19 (2014) 3, S. 116-119.

Maaß 2016:

Maaß, Philipp: Free/Libre/Open-Source Software in wissenschaftlichen Bibliotheken in Deutschland. Eine explorative Studie in Form einer Triangulation qualitativer und quantitativer Methoden. Köln: TH Köln, Fakultät Informations- und Kommunikationswissenschaften, Institut für Informationswissenschaft, 2016. <http://nbn-resolving.de/urn:nbn:de:hbz:79pbc-opus-8299> (18.12.2016).

Maylein/Langenstein 2013:

Maylein, Leonhard; Langenstein, Annette (2013): Neues vom Relevanz-Ranking im HEIDI-Katalog der Universitätsbibliothek Heidelberg. In: b.i.t.online 16 (2013) 3, S. 190-200.

Meyer-Heß 2016:

Telefongespräch mit Anke Meyer-Heß am 22.07.2016.

[Mozilla 01]:

Mozilla. Vollständige Webseite Screenshots. <https://addons.mozilla.org/de/firefox/addon/fireshot/> (24.07.2016).

Nagy 2011:

Nagy, Andrew: Analyzing the Next-Generation Catalog. In: Library Technology Reports 47 (2011) 7, S. 5-27.

Nohr 2005:

Nohr, Holger: Grundlagen der automatischen Indexierung. Ein Lehrbuch. 3., überarb. Aufl. Aufl. Berlin: Logos-Verl., 2005.

Open Source Initiative 2007:

Open Source Initiative: The Open Source Definition. 2007. <https://opensource.org/osd> (16.12.2016).

Oßwald 2016:

Oßwald, Achim: Der Markt der geschäftsgangorientierten Bibliothekssysteme in Deutschland. Ausgewählte Entwicklungsschritte. Köln: TH Köln, Fakultät Informations- und Kommunikationswissenschaften, Institut für Informationswissenschaft, 2016.

Pianos 2014:

Pianos, Tamara: Hin zum Fachportal? Hin zur Kundschaft? Oder beides? In: Bibliothek 38 (2014) 1, S. 47-55.

Pieper 2011:

Pieper, Dirk: BASE Umstellung erfolgreich 2011. http://ekvv.uni-bielefeld.de/blog/base/entry/base_umstellung_erfolgreich (14.12.2016).

Pinna 2016a:

Pinna, Loredana: AW: Bibliothekarische Fachfragen zum Online-Katalog.
Persönliche E-Mail vom 26.07.2016.

Pinna 2016b:

Pinna, Loredana: AW: Bibliothekarische Fachfragen zum Online-Katalog.
Persönliche E-Mail vom 02.08.2016.

Pössel 2015:

Pössel, Jana: LIVIVO: Das neue ZB MED-Suchportal Lebenswissenschaften.
In: GMS Medizin, Bibliothek, Information 15 (2015) 3. <http://dx.doi.org/10.3205/mbi000352> (08.06.2016).

Quitzsch 2014:

Quitzsch, Nicole: Vorbei die Zeiten von Gebrauchsanweisungen für
Bibliotheksportale. In: Bibliotheksdienst 48 (2014) 10, S. 795-806.

Schnalke 2014:

Schnalke, Markus: Open Access, Freie Software und Co.: Eine Analyse der
Gemengelage. In: Perspektive Bibliothek 3 (2014) 2, S. 44-60.
<http://dx.doi.org/10.11588/pb.2014.2.16806> (06.03.2016).

[SLUB Dresden 01]:

SLUB Dresden. SLUB-Katalog beta. <http://www.slub-dresden.de/recherche/slub-katalog-beta/> (16.08.2016).

Störl 2015:

Störl, Uta: NoSQL-Datenbanksysteme. In: Taschenbuch Datenbanken. Hrsg.
von Thomas Kudraß. 2., neu bearbeitete Aufl. München: Hanser, 2015, S.
368-393.

Tannhof 2013:

Tannhof, Werner: Das deutsche wissenschaftliche Bibliothekswesen jenseits
der Bibliothek 2.0 – Zukunft jetzt gestalten. In: 027.7 Zeitschrift für
Bibliothekskultur 1 (2013) 1, S. 5-13. <http://dx.doi.org/10.12685/027.7-1-1-10> (03.04.2016).

Tiemann 2016:

Tiemann, Sarah: Vom klassischen OPAC zum modernen Rechercheportal.
Ansätze zur Einführung eines Discovery Systems an der ZHB Lübeck.
Berlin: Institut für Bibliotheks- und Informationswissenschaft der
Humboldt-Universität zu Berlin, 2016. (Berliner Handreichungen zur
Bibliotheks- und Informationswissenschaft, 406). <http://nbn-resolving.de/urn:nbn:de:kobv:11-100236890> (17.06.2016).

[UB Duisburg-Essen 01]:

UB Duisburg-Essen. Willkommen in der Universitätsbibliothek Duisburg-Essen.
<https://www.uni-due.de/ub/index.php> (24.09.2016).

Umstätter/Wagner-Döbler 2005:

Umstätter, Walther; Wagner-Döbler, Roland: Einführung in die Katalogkunde.
Vom Zettelkatalog zur Suchmaschine. 3. Aufl. des Werkes von Karl
Löffler. Aufl. Stuttgart: Hiersemann, 2005.

Vaughan 2011:

Vaughan, Jason (2011): Web Scale Discovery Services. In: Library Technology
Reports 47 (2011) 1, S. 5-61.

[VuFind 01]:

VuFind. Welcome to the VuFind Demo. <https://vufind.org/demo/> (14.01.2017).

Weinhold et. al. 2011:

Weinhold; Thomas; Bekavac, Bernard; Schneider, Gabi; Bauer, Lydia; Böller,
Nadja: Wissenschaftliche Suchmaschinen - Übersicht, Technologien,
Funktionen und Vergleich. In: Neue Entwicklungen in der Web-Suche.
Hrsg. von Dirk Lewandowski. Heidelberg: AKA Akad. Verl.-Ges., 2011.
(Handbuch Internet-Suchmaschinen, 2), S. 141-177.

Wonke-Stehle/Christof 2014:

Wonke-Stehle, Jens; Christof, Jürgen: Fachportale von Bibliotheken. In:
Bibliothek 38 (2014) 1, S. 14-19.

[Xapian 01]:

Xapian. <https://xapian.org/> (04.09.2016).

[Xapian 02]:

Xapian. History. <https://xapian.org/history> (04.09.2016).

[Xapian 03]:

Xapian. Detailed Example of Running Using Omega for the First Time.
<https://trac.xapian.org/wiki/OmegaExample> (01.08.2016).

Xapian 2017:

Xapian. Version vom 05.01.2017. In: Wikipedia. <https://en.wikipedia.org/wiki/Xapian> (05.01.2017).

Dokumentationsverzeichnis

Das Abrufdatum aller im Dokumentationsverzeichnis genannten Webseiten ist in Klammern angegeben.

Elasticsearch

Elasticsearch Guide 1.x

Elasticsearch: The Definitive Guide. Version 1.x. <https://www.elastic.co/guide/en/elasticsearch/guide/1.x/index.html> (05.01.2017).

Elasticsearch Guide 2.x

Elasticsearch: The Definitive Guide. Version 2.x. <https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html> (05.01.2017).

D. 1	You Know, for Search... Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/intro.html (28.08.2016).
D. 2	Talking to Elasticsearch. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/_talking_to_elasticsearch.html (28.08.2016).
D. 3	Installing and Running Elasticsearch. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/running-elasticsearch.html (28.08.2016).
D. 4	Document Oriented. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/_document_oriented.html (28.08.2016).
D. 5	Indexing Employee Documents. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/_indexing_employee_documents.html (28.08.2016).
D. 6	Search with Query DSL. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/_search_with_query_dsl.html (28.08.2016).
D. 7	Highlighting Our Searches. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/highlighting-intro.html (28.08.2016).

D. 8	Analytics. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/_analytics.html (28.08.2016).
D. 9	Add an Index. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/_add_an_index.html (28.08.2016).
D. 10	Updating a Whole Document. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/update-doc.html (28.08.2016).
D. 11	Analysis and Analyzers. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/analysis-intro.html (28.08.2016).
D. 12	Mapping. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/mapping-intro.html (28.08.2016).
D. 13	Sorting. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/_sorting.html (28.08.2016).
D. 14	Query-Time Search-as-You-Type. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/_query_time_search_as_you_type.html (29.08.2016).
D. 15	Lucene's Practical Scoring Function. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/practical-scoring-function.html (29.08.2016).
D. 16	Query-Time Boosting. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/query-time-boosting.html (29.08.2016).
D. 17	Boosting by Popularity. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/boosting-by-popularity.html (29.08.2016).
D. 18	Pluggable Similarity Algorithms. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/pluggable-similarities.html (29.08.2016).
D. 19	Getting Started with Languages. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/language-intro.html (29.08.2016).

D. 20	Identifying Words. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/identifying-words.html (29.08.2016).
D. 21	Algorithmic Stemmers. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/algorithmic-stemmers.html (29.08.2016).
D. 22	Dictionary Stemmers. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/dictionary-stemmers.html (29.08.2016).
D. 23	Using Stopwords. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/using-stopwords.html (29.08.2016).
D. 24	Stopwords and Performance. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/stopwords-performance.html (29.08.2016).
D. 25	Formatting Synonymns. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/synonym-formats.html (29.08.2016).
D. 26	Synonymns and The Analysis Chain. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/synonyms-analysis-chain.html (29.08.2016).
D. 27	Typoes and Mispelings. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/fuzzy-matching.html (29.08.2016).
D. 28	Scoring Fuzziness. Elasticsearch: The Definitive Guide. Version 2.x. https://www.elastic.co/guide/en/elasticsearch/guide/current/fuzzy-scoring.html (29.08.2016).

Elasticsearch Reference

Elasticsearch Reference. Version 5.1. <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html> (26.01.2017).

Elasticsearch Reference. Version 2.3. <https://www.elastic.co/guide/en/elasticsearch/reference/2.3/index.html> (26.01.2017).

D. 1	Facets. Elasticsearch Reference. Version 2.3. https://www.elastic.co/guide/en/elasticsearch/reference/2.3/search-facets.html (05.01.2017).
-------------	---

D. 2	Installation. Elasticsearch Reference. Version 2.3. https://www.elastic.co/guide/en/elasticsearch/reference/current/_installation.html (14.08.2016).
D. 3	Repositories. Elasticsearch Reference. Version 2.3. https://www.elastic.co/guide/en/elasticsearch/reference/current/setup-repositories.html (14.08.2016).

Solr

Apache Solr Documentation

Apache Solr 6.1.0 Documentation. http://lucene.apache.org/solr/6_1_0/index.html (13.08.2016).

Apache Solr Reference Guide 6.1

Apache Solr Reference Guide Covering Apache Solr 6.1. Apache Software Foundation (Hrsg.). <https://archive.apache.org/dist/lucene/solr/ref-guide/apache-solr-ref-guide-6.1.pdf> (13.08.2016).

Apache Solr Wiki

Welcome to the Apache Solr Wiki. FrontPage. Version vom 06.06.2016. In: Apache Solr Wiki. <http://wiki.apache.org/solr/> (13.08.2016).

Solr Quick Start

Solr Quick Start. <http://lucene.apache.org/solr/quickstart.html> (13.08.2016).

Xapian

Getting Started with Xapian 1.2

Xapian Documentation Team Contributors. Getting Started with Xapian 1.2. Release 1.2.19. 11.05.2016. <https://media.readthedocs.org/pdf/getting-started-with-xapian/latest/getting-started-with-xapian.pdf> (14.08.2016).

Xapian Administrator's Guide

Xapian Administrator's Guide. https://xapian.org/docs/admin_notes.html (26.08.2016).

Xapian Documentation

Xapian - the open source search engine. <https://xapian.org/docs/> (14.08.2016).

Anhang 1: Institutionen des Untersuchungsfeldes

Tabelle A1: Liste der Verbundzentralen

Verbundzentrale	Webseite	Verbundkatalog ASP-Produkt
KOBV-Zentrale, <i>Berlin</i>	http://www.kobv.de/	KOBV-Portal ALBERT
HeBIS-Verbundzentrale, <i>Frankfurt/Main</i>	http://www.hebis.de/ http://hds.hebis.de/	Verbundkatalog HeBIS Discovery System
Verbundzentrale des GBV, <i>Göttingen</i>	http://www.gbv.de/	Gemeinsamer Verbundkatalog Lukida
hbz, <i>Köln</i>	https://www.hbz-nrw.de/	hbz-Verbundkatalog DigiBib IntrOX
BSZ, <i>Konstanz</i>	https://www.bsz-bw.de/	SWB-Online-Katalog BOSS 2.2
VZ des Bibliotheksverbundes Bayern, <i>München</i>	http://www.bib-bvb.de/	Gateway Bayern OPACplus

Tabelle A2: Liste der Öffentlichen Bibliotheken in Städten mit mehr als 400.000 Einwohnern

Bibliothek	Webseite	Produkt / Link
VÖBB, <i>Berlin</i>	http://www.voebb.de/	Suche > Schnellsuche
ZLB, <i>Berlin</i>	http://www.zlb.de/	"im Katalog (VÖBB) suchen"
StB <i>Bielefeld</i>	https://www.stadtbibliothek-bielefeld.de/	"Mediensuche"
StB <i>Bochum</i>	http://www.bochum.de/stadtbuecherei	OPAC
StB <i>Bremen</i>	http://www.stabi-hb.de/	OPAC
StuLB <i>Dortmund</i>	http://www.bibliothek.dortmund.de/	Suchen und Finden
StB <i>Dresden</i>	http://www.bibo-dresden.de/	Online-Katalog
StB <i>Duisburg</i>	http://www.stadtbibliothek-duisburg.de/	Medienkatalog
StB <i>Düsseldorf</i>	http://www.duesseldorf.de/stadtbuechereien	Katalogsuche
StB <i>Essen</i>	http://www.stadtbibliothek-essen.de/	"zum Katalog & zu Ihrem Konto"
StB <i>Frankfurt/Main</i>	http://www.stadtbuecherei.frankfurt.de/	"Kataloge und Medienkonto"
Bücherhallen <i>Hamburg</i>	http://www.buecherhallen.de/	"Suche nach Buch, Film, CD"
StB <i>Hannover</i>	http://www.stadtbibliothek-hannover.de/	"Mediensuche & Benutzerkonto"
StB <i>Köln</i>	http://www.stbib-koeln.de/	Bibliothekskatalog
<i>Leipziger StB</i>	http://stadtbibliothek.leipzig.de/	"Suche starten"
<i>Münchner StB</i>	http://www.muenchner-stadtbibliothek.de/	Onlinekatalog
StB <i>Nürnberg</i>	http://www.stadtbibliothek.nuernberg.de/	"Katalog mit Benutzerkonto (OPAC)"
StB <i>Stuttgart</i>	http://www.stuttgart.de/stadtbibliothek	"Katalog durchsuchen"
StB <i>Wuppertal</i>	http://www.wuppertal.de/stadtbib/	"ONLINE-KATALOG"

Tabelle A3: Liste der Bibliotheken mit überregionaler Bedeutung

Bibliothek	Webseite	Produkt / Link
SBB-PK, <i>Berlin</i>	http://staatsbibliothek-berlin.de/	stabikat+
DNB, <i>Frankfurt/ Main</i>	https://portal.dnb.de/	Katalog der DNB
TIB, <i>Hannover</i>	https://www.tib.eu/	TIB-Portal
ZBW, <i>Kiel</i>	http://www.zbw.eu/	EconBiz
ZB MED, <i>Köln</i>	http://www.zbmed.de/	LIVIVO
BSB, <i>München</i>	https://www.bsb-muenchen.de/	BSB-Katalog

Tabelle A4: Liste der Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden

Bibliothek	Webseite	Produkt / Link
UB der RWTH <i>Aachen</i>	http://www.ub.rwth-aachen.de/	Katalog der UB
UB der FU <i>Berlin</i>	http://www.fu-berlin.de/sites/ub	Primo
UB der HU <i>Berlin</i>	https://www.ub.hu-berlin.de/	Primus
UB der TU <i>Berlin</i>	http://www.ub.tu-berlin.de/	Wissensportal Primo
UB <i>Bochum</i>	https://suchen.ub.rub.de/de/	"suchen"
ULB <i>Bonn</i>	https://www.ulb.uni-bonn.de/	bonnus
SUB <i>Bremen</i>	http://www.suub.uni-bremen.de/	E-LIB
UB <i>Dortmund</i>	https://www.ub.tu-dortmund.de/	Katalog plus
SLUB <i>Dresden</i>	http://katalogbeta.slub-dresden.de	SLUB-Katalog beta
UB <i>Duisburg-Essen</i>	https://www.uni-due.de/ub/	"Primo, der Katalog der UB"
ULB <i>Düsseldorf</i>	https://www.ulb.hhu.de/	Katalog
UB <i>Frankfurt/Main</i>	https://www.ub.uni-frankfurt.de/	Suchportal
SUB <i>Göttingen</i>	https://www.sub.uni-goettingen.de/	Göttinger Universitätskatalog
FernUB <i>Hagen</i>	https://www.ub.fernuni-hagen.de	"Schnellsuche im Katalog"
SUB <i>Hamburg</i>	https://beluga.sub.uni-hamburg.de/	beluga
UB <i>Heidelberg</i>	http://www.ub.uni-heidelberg.de/	HEIDI
KIT-Bibliothek, <i>Karlsruhe</i>	http://bibliothek.kit.edu/	KIT-Katalog
USB <i>Köln</i>	https://www.ub.uni-koeln.de/	USB-Portal, KUG
UB <i>Mainz</i>	https://www.ub.uni-mainz.de/	Rechercheportal Mainz
UB der LMU <i>München</i>	http://www.ub.uni-muenchen.de/	Online-Katalog
UB der TU <i>München</i>	http://www.ub.tum.de/	OPACplus BETA
ULB <i>Münster</i>	https://www.ulb.uni-muenster.de/	DISCO
UB <i>Erlangen-Nürnberg</i>	http://ub.fau.de/	Primo Discovery Beta

Anhang 2: Zusatzmaterial Kapitel 3 (Abbildungen)



Abbildung A1; Suchschlitz auf der Startseite von EconBiz¹⁸⁶

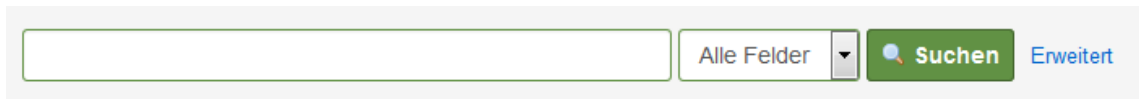


Abbildung A2: Suchschlitz auf der Startseite der VuFind Demoseite¹⁸⁷



Abbildung A3: Seitenende einer Trefferliste in EconBiz¹⁸⁸

¹⁸⁶ Der zugehörige URL ist „<http://www.econbiz.de>“ und wurde am 13.07.2016 abgerufen.

¹⁸⁷ Der zugehörige URL ist „<https://vufind.org/demo/>“ und wurde am 21.07.2016 abgerufen.

¹⁸⁸ Der zugehörige URL ist „<http://www.econbiz.de/Search/Results?lookfor=cologne&type=AllFields>“ und wurde am 13.07.2016 abgerufen.

20  **The silent angel /**
 von Böll, Heinrich, 1917-
 Veröffentlicht 1994
 Schlagworte: "...Reconstruction (1939-1951) Germany
 Cologne Fiction..."
 Signatur: U788.4790
 Standort: Mehrere Standorte
 Buch Verfügbar

Alles auswählen | Auswahl:

1 2 3 4 5 6 7 8 9 10 11 Nächster » [15]

Suchwerkzeuge:  RSS-Feed abonnieren —  Diese Suche als E-Mail versenden —  Suche speichern

Abbildung A4: Seitenende einer Trefferliste auf der VuFind Demoseite¹⁸⁹

« Zurück #10 von 2907 Weiter »

Services liberalisation in Germany : overview and the potential of deregulation ; a study by the institute for economic policy at the university of Cologne
 Oliver Arentz; Hans Manner; Leonard Münstermann; Clemens Recker; Steffen J. Roth; Clemens Recker

The structure of the German economy changed drastically over time. The decline of the proportion of gross value added of the manufacturing sector at the expense of the services sector is often cited as an indicator for this structural change. However, this shift is not necessarily an indication of a... Ausführliche Beschreibung

Erscheinungsjahr: Mrz. 2015
 Weitere Verfasser/innen: Arentz, Oliver; Manner, Hans; Münstermann, Leonard; Recker, Clemens; Roth, Steffen J.; Recker, Clemens
 Verlag: Köln : Otto-Wolff-Inst. für Wirtschaftsordnung
 Beschreibung: Online-Ressource (52 S.) graph. Darst.
 Schriftenreihe: Diskussionspapier / Otto-Wolff-Institut für Wirtschaftsordnung ; 01b/2015
 Sprache: Englisch
 Publikationsform (Subkategorien): Arbeitspapier Working Paper Graue Literatur Non-commercial literature
 Publikationsform: Buch / Working Paper
 Anmerkungen: Systemvoraussetzungen: Acrobat Reader
 Nachweis aus Datenbank: ECONIS - Online-Katalog der ZBW
 Verfügbarkeit:  zum Volltext  Weitere Zugänge

 Ähnliches Autorenspektrum

- Der Dienstleistungssektor in Deutschland : Überblick und Deregulierungspotenziale von: Oliver Arentz Veröffentlicht: (Mrz. 2015)
- Entry in German pharmacy market von: Oliver Arentz Veröffentlicht: (März 2016)
- Apothekenmarkt in Deutschland : hohe Regulierungsanforderungen als Gefahr für die Versorgungssicherheit? von: Clemens Recker Veröffentlicht: (Nov. 2014)
- Mehr

Zitieren E-Mail Exportieren In Merkliste speichern f teilen tweet +1

Exemplare Beschreibung Fehler melden

Exemplare in der ZBW
 Link: http://www.otto-wolff-institut.de/Publikationen/DiskussionPapers/OWWO_dp01b_2015.pdf

Abbildung A5: Einzeltrefferanzeige in EconBiz¹⁹⁰

¹⁸⁹ Der zugehörige URL ist „https://vufind.org/demo/Search/Results?lookfor=cologne&type=AllFields“ und wurde am 21.07.2016 abgerufen.

¹⁹⁰ Der zugehörige URL ist „http://www.econbiz.de/Record/services-liberalisation-germany-overview-and-the-potential-deregulation-study-the-institute-for-economic-policy-the-university-cologne-arentz-oliver/10011295772“ und wurde am 13.07.2016 abgerufen.



The silent angel /

1. Verfasser:	Böll, Heinrich, 1917-
Format:	Buch
Sprache:	English German
Veröffentlicht:	New York : St. Martin's Press, c1994.
Ausgabe:	1st ed.
Schlagworte:	Reconstruction (1939-1951) > Germany > Cologne > Fiction. Cologne (Germany) > Fiction.
Tags:	Keine Tags. Fügen Sie den ersten Tag hinzu! Tag hinzufügen

Ähnliche Einträge

- [The train was on time /](#)
 von: Böll, Heinrich, 1917-
 Veröffentlicht: (1994)
- [Wo warst du, Adam? Roman /](#)
 von: Böll, Heinrich, 1917-
 Veröffentlicht: (1965)
- [Adam, and, The train: two novels.](#)
 von: Böll, Heinrich, 1917-
 Veröffentlicht: (1970)
- [The stories of Heinrich Böll /](#)
 von: Böll, Heinrich, 1917-
 Veröffentlicht: (1986)
- [Stories, political writings, and autobiographical works /](#)
 von: Böll, Heinrich, 1917-1985.
 Veröffentlicht: (2006)

[Exemplare](#)
[Beschreibung](#)
[Kommentare](#)
[Ähnliche Einträge](#)
[Internformat](#)

[Login](#) für die Nachverfolgung von Bestellungen und Vormerkungen

Campus C

Signatur:	A843.9826 J174.3950
------------------	------------------------

Abbildung A6: Einzeltrefferanzeige der VuFind Demoseite¹⁹¹

¹⁹¹ Der zugehörige URL ist „https://vufind.org/demo/Record/401304“ und wurde am 21.07.2016 abgerufen.

Anhang 3: Liste der Referenzinstallationen

Tabelle A5: Liste der Referenzinstallationen

Produkt	Hersteller	Referenzinstallation	URL
aDIS/BMS	aStec	StB Herne	http://www.stadtbibliothek.herne.de/
Aleph	Ex Libris	University of Amsterdam Library	http://uba.uva.nl/en/
BIBLIOTHECAplus	OCLC	Büchereien Wien	https://www.buechereien.wien.at/
BIBLIOTHECAplus OPEN	OCLC	Öffentliche Bibliothek Ahrntal	http://biblio.bz.it/ahrntal
DigiBib	HBZ	Bibliothek der TH Köln	http://www.bibl.th-koeln.de/
EBSCO Discovery Service (EDS)	EBSCO	UC Berkeley Library	http://www.lib.berkeley.edu/
Pica Search & Index (PSI)	OCLC	Gemeinsamer Verbundkatalog	https://gso.gbv.de/DB=2.1/
Portfolio	BiblioMondo	Paris Town Hall	http://bibliotheques-specialisees.paris.fr/
Primo	Ex Libris	Bibliothek der ETH Zürich	http://www.library.ethz.ch/en
SISIS-SunRise	OCLC	StB Bremen	http://www.stabi-hb.de/
TouchPoint	OCLC	UB Bayreuth	http://www.ub.uni-bayreuth.de/
VuFind	Villanova University	VuFind Demo	http://vufind.org/demo/

Anhang 4: Zusatzmaterial Kapitel 3 (Tabellen)

Tabelle A6: Ergebnis für die Verbundkataloge

Verbundzentrale	Verbundkatalog	Art	System / Basis
KOBV-Zentrale, <i>Berlin</i>	KOBV-Portal	E	Solr
HeBIS-Verbundzentrale, <i>Frankfurt/Main</i>	Verbundkatalog	O	PSI
Verbundzentrale des GBV, <i>Göttingen</i>	Gemeinsamer Verbundkatalog	O	PSI
hbz, <i>Köln</i>	hbz-Verbundkatalog	O	Aleph
BSZ, <i>Konstanz</i>	SWB-Online-Katalog	O	PSI
VZ des Bibliotheksverbundes Bayern, <i>München</i>	Gateway Bayern	S	TouchPoint

Tabelle A7: Ergebnis für die ASP-Produkte der Verbundzentralen

Verbundzentrale	Produkt	Art	System / Basis
KOBV-Zentrale, <i>Berlin</i>	ALBERT	E	Solr
HeBIS-Verbundzentrale, <i>Frankfurt/Main</i>	HeBIS Discovery System	N	VuFind
Verbundzentrale des GBV, <i>Göttingen</i>	Lukida	E	Solr
hbz, <i>Köln</i>	DigiBib IntroOX	E	Elasticsearch
BSZ, <i>Konstanz</i>	BOSS 2.2	N	VuFind
VZ des Bibliotheksverbundes Bayern, <i>München</i>	OPACplus	S	TouchPoint

Tabelle A8: Ergebnis für die Öffentlichen Bibliotheken in Städten mit mehr als 400.000 Einwohnern

Bibliothek	Bezeichnung	Art	System / Basis
VÖBB, <i>Berlin</i>	Suche	O	aDIS/BMS
ZLB, <i>Berlin</i>	Katalog (VÖBB)	O	aDIS/BMS
StB <i>Bielefeld</i>	Mediensuche	O	BIBLIOTHECAplus OPEN
StB <i>Bochum</i>	OPAC	O	BibDIA
StB <i>Bremen</i>	OPAC	O	SISIS-SunRise
StuLB <i>Dortmund</i>	Suchen und Finden	O	DigiBib
StB <i>Dresden</i>	Online-Katalog	O	SISIS-SunRise
StB <i>Duisburg</i>	Medienkatalog	O	BibDIA
StB <i>Düsseldorf</i>	Katalogsuche	O	aDIS/BMS
StB <i>Essen</i>	Katalog	O	BibDIA
StB <i>Frankfurt/Main</i>	Katalog	O	BIBLIOTHECAplus
Bücherhallen <i>Hamburg</i>	Suche	E	Solr
StB <i>Hannover</i>	Mediensuche	O	Portfolio
StB <i>Köln</i>	Bibliothekskatalog	O	Portfolio
<i>Leipziger</i> StB	Suche	O	SISIS-SunRise
<i>Münchner</i> StB	Onlinekatalog	O	aDIS/BMS
StB <i>Nürnberg</i>	Katalog	O	aDIS/BMS
StB <i>Stuttgart</i>	Katalog	O	BIBLIOTHECAplus
StB <i>Wuppertal</i>	Online-Katalog	O	SISIS-SunRise

Tabelle A9: Ergebnis für die Bibliotheken mit überregionaler Bedeutung

Bibliothek	Katalog	Art	System / Basis
SBB-PK, <i>Berlin</i>	stabikat+	S	EDS
DNB, <i>Frankfurt/Main</i>	Katalogportal	E	Solr
TIB, <i>Hannover</i>	TIB-Portal	E	Solr
ZBW, <i>Kiel</i>	EconBiz	N	VuFind
ZB MED, <i>Köln</i>	LIVIVO	E	Solr
BSB, <i>München</i>	BSB-Katalog	S	TouchPoint

Tabelle A10: Ergebnis für die Universitätsbibliotheken an Universitäten mit mehr als 30.000 Studierenden

Bibliothek	Katalog	Art	System / Basis
UB der RWTH <i>Aachen</i>	Katalog der UB	S	TouchPoint
UB der FU <i>Berlin</i>	Primo	S	Primo
UB der HU <i>Berlin</i>	Primus	S	Primo
UB der TU <i>Berlin</i>	Wissensportal Primo	S	Primo
UB <i>Bochum</i>	"suchen"	E	Solr
ULB <i>Bonn</i>	bonnus	N	VuFind
SUB <i>Bremen</i>	E-LIB	E	CIXbase
UB <i>Dortmund</i>	Katalog plus	O	DigiBib
SLUB <i>Dresden</i>	SLUB-Katalog beta	E	Solr
UB <i>Duisburg-Essen</i>	"Primo, der Katalog der UB"	S	Primo
ULB <i>Düsseldorf</i>	Katalog	S	Primo
UB <i>Frankfurt/Main</i>	Suchportal	S	HDS
SUB <i>Göttingen</i>	Göttinger Universitätskatalog	O	PSI
FernUB <i>Hagen</i>	"Schnellsuche im Katalog"	O	DigiBib
SUB <i>Hamburg</i>	beluga	N	VuFind
UB <i>Heidelberg</i>	HEIDI	E	Solr
KIT-Bibliothek, <i>Karlsruhe</i>	KIT-Katalog	S	Primo
USB <i>Köln</i>	USB-Portal, KUG	E	Xapian
UB <i>Mainz</i>	Rechercheportal Mainz	S	HDS
UB der LMU <i>München</i>	Online-Katalog	S	TouchPoint
UB der TU <i>München</i>	OPACplus BETA	S	Primo
ULB <i>Münster</i>	DISCO	S	Primo
UB <i>Erlangen-Nürnberg</i>	Primo Discovery Beta	S	Primo

Anhang 5: Datenblatt zu Elasticsearch

Dokumentennummern ohne genannte Quelle beziehen sich auf den Elasticsearch Guide 2.x.

Basisinformationen		
Hersteller	Elastic	Elasticsearch 2016
Aktuelle Version (Stand: 05.01.2017)	5.1.1 vom 08.12.2016	Elasticsearch 2016
Untersuchte Version	2.3.5 vom 03.08.2016	Elasticsearch 2016
Markteinführung	2010	Elasticsearch 2016
Lizenz	Apache	Elasticsearch 2016
Systemarchitektur		
Typ	Webservice (REST)	D. 1
Unterbau	Lucene	D. 1
Programmiersprache	Java	D. 1
Schnittstellen	Groovy, Java, JavaScript, .NET, Perl, PHP, Python, Ruby	D. 2
Betriebssystem	plattformübergreifend	Elasticsearch 2016
Datenhaltung		
DBMS	dokumentorientiert	D. 4
invertierte Liste	ja	D. 5
Datenformat	JSON	D. 4
Feldstruktur	ja	D. 12
Aktualisierung	ja (alte Version wird gelöscht)	D. 10
Löschung	ja (wird deaktiviert)	D: 10
Größe	unendlich (Sharding)	D. 9
Indexierung		
Indexer	Analyzer (Tokenizer, Filter)	D. 11
Stemming	algorithmisch, wörterbuchbasiert	D. 21 u. D. 22
Stoppwörter	ja	D. 23
Synonyme	einfach, mehrfach	D. 25
Matching		
Query Parser	Query DSL	D. 6
Rechtschreibkorrektur	indexbasiert	D. 27
Autovervollständigung	indexbasiert	D. 14
Ergebnispräsentation		
Ranking	Relevanz, Feldwerte; mehrstufig	D. 13
IR-Modell	Lucene (Boolean, Vektorraum) BM25 (probabilistisch)	D.15 u. D. 18
Boosting	Indexierung, Suche, Popularität	D. 15 – D. 17
Highlighting	ja	D. 7
Facetten	ja	Anmerkung Nr. 1
Relevance Feedback	ja	D. 28

Anmerkung Nr. 1: Die Facetten wurden mit der Version 1.0 von Elasticsearch durch eine Funktionalität namens "Aggregations" ersetzt, vgl. Elasticsearch Reference, D. 1. Zur Funktionalität "Aggregations" vgl. Elasticsearch Guide, D. 8.

Anhang 6: Datenblatt zu Solr

Seitenzahlen ohne genannte Quelle beziehen sich auf den Apache Solr Reference Guide 6.1.

Basisinformationen		
Hersteller	Apache Software Foundation	Apache Solr 2016b
Aktuelle Version (Stand: 05.01.2017)	6.3.0 vom 08.11.2016	Apache Solr 2016b
Untersuchte Version	6.1.0 vom 17.06.2016	Apache Solr 2016a
Markteinführung	2004	Apache Solr 2016b
Lizenz	Apache	Apache Solr 2016b
Systemarchitektur		
Typ	Webservice (REST)	S. 12
Unterbau	Lucene	S. 12
Programmiersprache	Java	S. 5
Schnittstellen	AJAX, C#, ColdFusion, Forrest/ Cocoon, Java, JavaScript, .NET, Perl, PHP, Python, Rails, Ruby	S. 682
Betriebssystem	plattformübergreifend	Apache Solr 2016b
Datenhaltung		
DBMS	dokumentorientiert	S. 48
invertierte Liste	ja	S. 95
Datenformat	XML (primär), CSV, JSON	S. 12
Feldstruktur	ja	S. 48 f.
Aktualisierung	ja (alte Version wird ersetzt)	S. 184
Löschung	ja	S. 185
Größe	unendlich (Sharding)	S. 13
Indexierung		
Indexer	Analyzer (Tokenizer, Filter)	S. 104 ff.
Stemming	algorithmisch, wörterbuchbasiert	S. 108 u. S. 123
Stoppwörter	ja	S. 136
Synonyme	einfach, mehrfach	S. 138 f.
Matching		
Query Parser	Standard, DisMax, Extended DisMax	S. 246
Rechtschreibkorrektur	indexbasiert, wörterbuchbasiert	S. 326 f.
Autovervollständigung	indexbasiert, wörterbuchbasiert	S. 341 ff.
Ergebnispräsentation		
Ranking	Relevanz, Feldwerte; mehrstufig	S. 248
IR-Modell	Lucene (Boolean, Vektorraum) BM25 (probabilistisch)	Anmerkung Nr. 2 S. 71
Boosting	Indexierung, Suche, Sponsored Search	S. 184, S. 258, S. 397 ff.
Highlighting	ja	S. 317 ff.
Facetten	ja	S. 298 ff.
Relevance Feedback	ja	S. 351 ff.

Anmerkung Nr. 2: Das IR-Modell wird in Solr über eine Klasse innerhalb des Schemas bestimmt, vgl. Apache Solr Reference Guide, S. 71 f. Die ursprüngliche Klasse "DefaultSimilarityFactory" wurde mit der Solr-Version 6.0 in "ClassicSimilarityFactory" umbenannt, vgl. Apache Solr Reference Guide, S. 26. Diese Klasse definiert das IR-Modell von Lucene und besteht aus einer Kombination von Booleschem Modell und Vektorraummodell, vgl. Grainger/Potter, S. 65 ff.

Anhang 7: Datenblatt zu Xapian

Seitenzahlen ohne genannte Quelle beziehen sich auf Getting Started with Xapian 1.2.

Basisinformationen		
Hersteller	xapian.org	Xapian 2017
Aktuelle Version (Stand: 05.01.2017)	1.4.2 vom 26.12.2016	Xapian 2017
Untersuchte Version	1.4.0 vom 25.06.2016	[Xapian 01]
Markteinführung	2001	History
Lizenz	GPL	Xapian 2017
Systemarchitektur		
Typ	Softwarebibliothek	S. 3
Unterbau	N/A	
Programmiersprache	C++	[Xapian 01]
Schnittstellen	C#, Erlang, Java, JavaScript (Node.js), Lua, Perl, Python, PHP, Ruby, Tcl	[Xapian 01]
Betriebssystem	plattformübergreifend	Xapian 2017
Datenhaltung		
DBMS	dokumentorientiert	S. 10
invertierte Liste	ja	S. 10
Datenformat	proprietär (Chert, Glass)	S. 10, S. 96
Feldstruktur	ja (durch Präfixe)	S. 13
Aktualisierung	ja	S. 26
Löschung	ja	S. 26
Größe	ca. 4,3 Milliarden Datensätze	S. 15
Indexierung		
Indexer	Termgenerator	S. 14
Stemming	ja	S. 13
Stoppwörter	ja	S. 14
Synonyme	Wörterbuch, Thesaurus	S. 54 f.
Matching		
Query Parser	Xapian QueryParser	S. 18 ff.
Rechtschreibkorrektur	indexbasiert, wörterbuchbasiert	S. 51 f.
Autovervollständigung	indexbasiert	S. 20 f.
Ergebnispräsentation		
Ranking	Relevanz, Feldwerte; mehrstufig	S. 46 f.
IR-Modell	BM25 (probabilistisch)	S. 56 f.
Boosting	Suche	S. 59
Highlighting	nein	
Facetten	ja	S. 42 ff.
Relevance Feedback	nein	

Anhang 8: Installation der Enterprise-Suchmaschinen (Details)

Elasticsearch

Für Elasticsearch 2.3.5 wird Java mindestens in der Version 7 vorausgesetzt. Empfohlen wird die Installation des Oracle JDK in der Version 1.8.0_73.¹⁹² In der Virtuellen Maschine für Elasticsearch wurde das Oracle JDK in der Version 1.8.0_101 installiert.

Anschließend wurde Elasticsearch über das in Xubuntu genutzte Paketverwaltungssystem Advanced Packaging Tool (APT) installiert. Dazu wurde zunächst das Elasticsearch-Repository in die Paketverwaltung eingebunden und anschließend Elasticsearch über das Paket „elasticsearch“ installiert und dann gestartet.¹⁹³

Gemäß der Installationsanleitung soll Elasticsearch nach dem Start über das Kommando „curl 'http://localhost:9200/?pretty'“ im Terminal erreichbar sein. Als Antwort werden einige in JavaScript Object Notation (JSON) ausgegebene Informationen zum laufenden Elasticsearch-Knoten erwartet.¹⁹⁴ Stattdessen wurde die Meldung „curl: (7) Failed to connect to localhost port 9200: Verbindungsaufbau abgelehnt“ angezeigt.

Solr

Für Solr 6.1.0 wird das Java Runtime Environment (JRE) in der Version 1.8 oder höher benötigt.¹⁹⁵ Daher wurde in der Virtuellen Maschine für Solr das Oracle JRE in der Version 1.8.0_101 installiert.

Anschließend wurde Solr gemäß der Installationsanleitung installiert. Die Adminkonsole war wie angegeben über die Adresse <http://localhost:8983/solr> erreichbar.¹⁹⁶

¹⁹² Vgl. Elasticsearch Reference 2.3, D. 2.

¹⁹³ Vgl. Elasticsearch Reference 2.3, D. 3.

¹⁹⁴ Vgl. Elasticsearch Guide 2.x, D. 3. Dieser Weg wird ebenfalls bei Drost-Fromm 2016, S. 154 beschrieben.

¹⁹⁵ Vgl. Apache Solr Reference Guide 6.1, S. 5.

¹⁹⁶ Vgl. Apache Solr Reference Guide 6.1, S. 6 ff.

Xapian

Zusätzlich zu Xapian wurde die auf Xapian aufbauende Suchapplikation Omega installiert. Dazu wurde eine über das Xapian-Wiki verfügbare Installationsanleitung für die Version 1.0.5 von Xapian genutzt.¹⁹⁷ Statt dieser Version wurde die aktuelle Version 1.4.0 installiert und die Anleitung auf diese angewendet. Die Voraussetzung für die Installation sind ein Webserver und ein C++-Compiler. Mit g++ ist letzterer innerhalb von Xubuntu bereits installiert. Als Webserver wurde Apache mithilfe der APT-Paketverwaltung über das Paket „apache2“ installiert. Die Installation musste mehrfach von vorne begonnen werden, da verschiedene fehlende Systemkomponenten¹⁹⁸ jeweils zu einem Abbruch der Installation führten. Nach der Installation dieser Komponenten konnte die Installation erfolgreich abgeschlossen werden. Die in der Installationsanleitung angegebene Adresse „http://localhost/cgi-bin/omega.cgi“ war jedoch nicht über den Browser abrufbar. Gemeldet wird „Forbidden. You don't have permission to access /cgi-bin/omega.cgi on this server.“ (HTTP-Fehler 403).

¹⁹⁷ Vgl. [Xapian 03].

¹⁹⁸ Die benötigten Komponenten wurden über die folgenden Pakete mithilfe der APT-Paketverwaltung installiert: „zlib1g-dev“, „uuid-dev“, „libmagic-dev“ und „libpcre++-dev“.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Dies gilt auch für Quellen aus eigenen Arbeiten.

Ich versichere, dass ich diese Arbeit oder nicht zitierte Teile daraus vorher nicht in einem anderen Prüfungsverfahren eingereicht habe.

Mir ist bekannt, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs mittels einer Plagiatserkennungssoftware auf ungekennzeichnete Übernahme von fremdem geistigem Eigentum überprüft werden kann.

Köln, den 03.02.2016

Stefan Niesner