Relevanz-Ranking im OPAC der Universitätsbibliothek Heidelberg

Annette Langenstein und Leonhard Maylein

In Bibliothekskatalogen kommt der „Treffersortierung nach Relevanz" immer größere Bedeutung zu. Der Aufsatz beschreibt verschiedene Möglichkeiten zur Optimierung des Trefferrankings am Beispiel des Lucene-basierten OPACs der UB Heidelberg. Zur Bestimmung der Relevanz können die Inhalte einzelner Datenfelder analysiert und gewichtet, es können Kriterien der Popularität, der Verfügbarkeit oder der Bewertung eines Titels, oder auch Nutzerprofile berücksichtigt werden. Im Beitrag werden verschiedene Gewichtungsmöglichkeiten und Lösungsansätze für weitere Kriterien aufgezeigt.

Heutige Nutzergenerationen sind mit Internetwerkzeugen wie Suchmaschinen, Web-Blogs, Foren und sozialen Netzwerkdiensten bestens vertraut. Die Erwartung der Nutzer, einen Bibliothekskatalog ebenso einfach und intuitiv wie diese Dienste bedienen zu können, ist nachvollziehbar. Das exzellente Relevanz-Ranking von Google liefert auch bei einfachen Stichwortsuchen relevante Treffer, ohne dass man sich mit kategorisierten Suchfeldern und Boole'schen Operatoren herumschlagen muss. Herkömmliche elektronische Bibliothekskataloge bieten zwar eine Fülle an untereinander verlinkten formalen und sachlichen Metadaten, aber die Aufbereitung für Recherche und Präsentation ist oft ungenügend. Die wenigsten Nutzer haben die Erfahrung und Kompetenz, ihre Recherche gemäß den komplizierten Regelwerken der formalen und inhaltlichen Erschließung zu strukturieren. Bei der Trefferanzeige erwarten die Nutzer, relevante Ergebnisse am Anfang der Liste vorzufinden. Passende Treffer auf den Folgeseiten werden häufig übersehen.[1]

Eine alphabetische Sortierung nach Verfasser- und Sachtitelschriften ist bestenfalls für spezielle Fragestellungen verwendbar, für größere Treffermengen ist sie ungeeignet. Auch die chronologische Sortierung nach Publikationsjahr kann nur Retrievalspezialisten einen zuverlässigen Überblick über die neueste Literatur eines Themengebietes bieten.

Das sogenannte Relevanz-Ranking (folgend Ranking genannt) basiert auf der bestmöglichen Analyse der Treffersätze nach festgelegten Kriterien, die sich nicht nur auf die reinen Metadaten des Titels beziehen müssen: Denkbar ist beispielsweise eine Einbeziehung der „Popularität" (Anzahl besitzender Bibliotheken, Auswertung der Ausleihstatistik), der „Verfügbarkeit" (Medium online verfügbar, entleihbar, per Fernleihe entleihbar) oder der „Bewertung" (Empfehlungen aus Bewertungsdiensten von Amazon, LibraryThing u.a.). Auch die Berücksichtigung eines vom Nutzer vorgegebenen Profils zu Fachgebiet, Themenschwerpunkt, Erscheinungszeitraum, Materialart oder Sprache ist sinnvoll.

Ziel des Rankings ist es, mit objektiven Kriterien möglichst nahe an die subjektive Relevanz-Bewertung des Nutzers zu gelangen.

Elektronische Bibliothekskataloge (OPACs) verwenden zunehmend Suchmaschinentechnologien, um neue Techniken der Datenhaltung, -indexierung und -aufbereitung zu nutzen und den Ansprüchen der Anwender gerecht zu werden.

Im Jahr 2007 wurde der Katalog der Universitätsbibliothek Heidelberg[2] auf Basis des Suchmaschinenframeworks Lucene entwickelt. Das Standard-Ranking von Lucene bedurfte jedoch der Optimierung und Anpassung an die Besonderheiten eines Bibliothekskataloges und an die Gegebenheiten des Heidelberger Bibliothekssystems. Die in diesem Beitrag dargestellten Maßnahmen zur Verbesserung des Relevanz-Rankings wurden im Rahmen eines studentischen Praktikums als Ergebnis der

1 „They will assume that if there is nothing relevant on the first page, there are no relevant results." (7)

2 http://katalog.ub.uni-heidelberg.de. Der Katalog beinhaltet die Bestände des Bibliothekssystems der Universität Heidelberg.

Beobachtungen von Nutzerrecherchen und -feedback an der UB Heidelberg entwickelt. Eine empirische Untersuchung der Rankingergebnisse fand bislang noch nicht statt. Eine solche Auswertung, die alles andere als trivial ist (3), soll das Ziel einer weiteren studentischen Arbeit sein. Die Motivation, diesen Beitrag auch ohne diese empirische Analyse zu veröffentlichen, resultiert aus der Tatsache, dass bislang zu kaum einem OPAC, der Relevanz-Ranking einsetzt, die konkreten Rankingmaßnahmen und -möglichkeiten publiziert wurden. Dies trifft nicht nur auf kommerzielle Systeme zu (10), auch zu offenen und/oder an den Bibliotheken selbst entwickelten Systemen findet sich wenig. Der vorliegende Artikel will daher die in der Regel recht einfach zu realisierenden möglichen „Ranking-Stellschrauben" für suchmaschinenbasierte OPACs beschreiben. Die genannten konkreten Einstellungen sind als Beispiele anzusehen.

1 Das Trefferranking von Lucene

Das von Lucene mitgelieferte Trefferranking orientiert sich am Vector Space Model (VSM)[3]. Der Fokus liegt dabei auf großen Sammlungen von eher weniger strukturierten Dokumenten. Wie es sich für ein Framework gehört, kann das Ranking angepasst oder durch einen eigenen Algorithmus ersetzt werden.

Die wichtigsten Aspekte des Lucene-Standardrankings sind:

- Je seltener ein Suchwort im Index vorhanden ist, desto höher ist sein Beitrag zum Ranking.
- Je länger der Feldinhalt ist, in dem ein Suchwort gefunden wurde, desto geringer ist sein Beitrag zum Ranking.
- Dokumente, in denen die Suchworte häufiger vorkommen, werden höher bewertet.
- Dokumente, in denen mehr Suchworte vorkommen, werden höher bewertet.[4]
- Einzelne Dokumente können beim Indexieren mit einem sogenannten Boostfaktor versehen und so insgesamt höher oder niedriger gewichtet werden.
- Einzelne Felder können beim Indexieren mit einem Boostfaktor versehen werden. Die Gewichtung der Felder beim Ranking kann hierüber beeinflusst werden.
- Einzelne Suchworte oder Phrasen können in der Suchanfrage mit einem Boostfaktor versehen und so mehr oder weniger stark gewichtet werden.

3 Eine Beschreibung der Lucene-Rankingformel findet sich in http://lucene.apache.org/java/2_4_1/scoring.html [Zugriff am 15.07.2009]

4 Dieses Kriterium spielt nur dann eine Rolle, wenn nicht alle Suchworte – wie dies in den meisten OPACs der Fall ist – mit dem boole'schen Und-Operator verknüpft werden.

2 Aufbau des Heidelberger OPACs

Der OPAC der Universitätsbibliothek Heidelberg besteht im Wesentlichen aus zwei Teilen, einem in Perl programmierten Frontend und einem JAVA-Backend, welches auf dem Lucene-Framework basiert. Das Frontend sorgt für die gesamte Webdarstellung, die Kommunikation mit dem Nutzer und das Sitzungsmanagement. Die Anfragen an den Lucene-Index erfolgen per RPC-Request an das Backend, welches verschiedene Funktionen für Recherche, Drill-Down und Suche von ähnlichen Titeln zur Verfügung stellt.

Der Heidelberger OPAC bietet sowohl eine feldübergreifende Freitextsuche mit einem einzelnen Suchfeld (im Folgenden auch „Einfache Suche" genannt) als auch ein Suchformular mit differenzierten Suchfeldern („Feldsuche"). Die nachfolgend beschriebenen Rankinganpassungen beziehen sich sowohl auf die Einfache Suche, auf die Feldsuche sowie auf die nachträgliche Treffereinschränkung (Drill-Down).

3 Statische Maßnahmen zur Anpassung des Rankings

Die in diesem Abschnitt beschriebenen Maßnahmen greifen bereits während des Indexaufbaus, sind also von der Suchanfrage selbst unabhängig. Sie können nur im Rahmen eines Neuaufbaus des Index verändert werden.[5]

```
public class UBHDSimilarity extends DefaultSimilarity {

        public float lengthNorm(String fieldName, int numTerms) {
                return (float)1;
        }
}
```

Abbildung 1

3.1 Feldlänge

Die von Lucene mitgelieferte Rankingformel berücksichtigt die Länge der Felder, in denen Suchbegriffe gefunden werden. Dies ist für wenig strukturierte Dokumente sinnvoll. Beim Ranking von Katalogisaten kann dies jedoch zu Verzerrungen führen. Die Länge der Katalogisate und deren einzelner Felder spielt für die Relevanzbewertung häufig keine Rolle (7). Zudem verwenden OPACs, die auf Suchmaschinentechnologie basieren, meist keine Normdatenindices. So werden im Heidelberger OPAC beispielsweise die Schlagworte eines Katalogisates jeweils mit Ansetzungs- und allen Verweisungsformen in einem Suchfeld „Schlagwort" indexiert. Selbstverständlich sollen Schlagworte mit einer großen Anzahl Verweisungsformen nicht zu einem ungünstigeren Ranking führen als Schlagworte mit weniger Verweisungsformen. Auch Anreicherungen der bibliografischen Angaben um Inhaltsverzeichnisse, Rezensionen bis hin zu elektronischen Volltexten sind geeignet zu berücksichtigen.[6]

Sicher finden sich hierzu auch Gegenbeispiele: Besteht der Hauptsachtitel nur aus den gewählten Suchbegriffen, so kann zunächst davon ausgegangen werden, dass dieser Titel relevanter ist als andere Titel, deren Hauptsachtitel zwar auch alle Suchbegriffe aber eben auch noch eine Vielzahl anderer Worte enthält. An der Universitätsbibliothek Heidelberg wurde dennoch zunächst der Weg gewählt, die Feldlänge nicht zu berücksichtigen, da bei den Titelkategorien die Schwankungen in den Feldlängen vergleichsweise klein sind und der Effekt auf das Ranking daher geringer ausfällt. Zudem werden diese Fälle teilweise im Rahmen anderer Rankingmaßnahmen aufgefangen (Feld-Boost bei Einworttiteln, Boost bei Phrasensuchen), die nachfolgend beschrieben werden.

In der DefaultSimilarity-Klasse von Lucene wurde die Funktion überschrieben, welche die sogenannte *lengthNorm* ermittelt. Sie liefert im Heidelberger OPAC unabhängig vom untersuchten Feld den Faktor 1,0 zurück (Abb. 1).

Da Lucene die Möglichkeit bietet, die *lengthNorm* abhängig vom jeweiligen Feld unterschiedlich zu ermitteln, ist zu überlegen, ob künftig bei einzelnen Feldern auf die Standardformel von Lucene zurückgegangen werden kann. Dies würde sich für die in einem separaten Feld indexierten Kataloganreicherungen (z.B. Inhaltsverzeichnisse, Klappentexte) anbieten, da diese Dokumente den klassischen Anwendungsfall von Lucene darstellen.[7]

Lucene ermittelt den *lengthNorm*-Faktor bei der Indexierung. Änderungen an dieser

5 Ein solcher Neuaufbau unter Lucene ist auch bei größeren Beständen in der Regel kurzfristig möglich. So beträgt die Indexierungszeit für die ca. 4,3 Mio. Titel des Heidelberger OPACs ca. 1 Stunde. (CPU: 2 x 3,0 GHz, 8 GByte RAM)

6 „Probleme bei den Bibliotheksinhalten ergeben sich auch dann, wenn zu unterschiedlichen Datensätzen eine unterschiedliche Informationsmenge zur Verfügung steht. So ist eine Rankingfunktion schwer auf Datensätze anzuwenden, wenn ein Datensatz nur aus bibliographischen Angaben besteht, während ein anderer Datensatz zusätzlich Text aus einem Inhaltsverzeichnis enthält." (8)

7 Zu beachten ist allerdings, dass sich dann die Gewichtung der Felder untereinander verschiebt.

Stelle der Rankingformel erfordern daher einen Neuaufbau des Indexes.

3.2 Dokumenten-Boost

Lucene erlaubt es, einzelne Dokumente (= Katalogisate) prinzipiell höher oder niedriger zu gewichten. Dazu kann bei der Indexierung eines Dokuments ein sogenannter „document boost" angegeben werden, der dann die Standardgewichtung von 1,0 überschreibt.
Diese einfachste Form der Beeinflussung des Rankings wurde beim Heidelberger OPAC für verschiedene Aspekte genutzt. Die nach den folgenden Regeln ermittelten Faktoren werden miteinander multipliziert und ergeben so den endgültigen Dokumenten-Boost.

Zeitschriften und Zeitungen

Für eine gezielte Recherche nach Zeitschriften und Zeitungen und der Übersichtlichkeit der Trefferliste wegen erlaubt der Heidelberger OPAC eine Einschränkung auf den Publikationstyp „Zeitschrift/Zeitung". Eine Recherche mit dieser Einschränkung liefert nur die Gesamtaufnahmen von Zeitschriften und Zeitungen. Die einzelnen, ebenfalls im Katalog vorhandenen Bände werden in der Trefferliste nicht angezeigt und sind nur über die Bandverknüpfung zugänglich. Da hierzu speziellere Retrievalkenntnisse erforderlich sind, sollen die Gesamtaufnahmen möglichst grundsätzlich — auch ohne die Nutzung dieser Option — vor den einzelnen Bänden präsentiert werden. Deshalb erhalten die Zeitschriften- und Zeitungsbände einen Boostfaktor von 0,7 und werden damit generell geringer als Gesamtaufnahmen, Monographien oder mehrbändige Werke gewichtet.

Auflagenwerke

Ein häufig genanntes Desiderat war es, dass neuere Auflagen eines Titels vor den Altauflagen angezeigt werden sollen. Hier stellte sich zunächst das Problem, wie die verschiedenen Auflagen erkannt werden können. Für die Lehrbuchsammlung der Universitätsbibliothek Heidelberg ist dies aufgrund der seit dem Jahr 2000 verwendeten Auflagenkennzeichnung in der Signatur recht einfach möglich.[8] Altauflagen, die hierüber erkannt werden, erhalten beim Indexieren einen Boostfaktor von 0,9.
Seit April 2009 bietet der OPAC der Universitätsbibliothek Heidelberg zudem eine Auflagenverknüpfung für den gesamten Bestand an. Über einen Link in der Trefferübersicht oder Detailanzeige können andere Auflagen oder Ausgaben (z.B. in einer anderen Sprache oder mit einem anderen Medientyp) aufgerufen werden. Die Datenbasis, die auf dem xISBN-Dienst des WorldCat beruht, soll zukünftig auch für die Gewichtung der Auflagen im Rahmen des Trefferrankings eingesetzt werden. Die eindeutige Ermittlung der Auflagenzählung ist hier schwieriger zu implementieren, da die von xISBN gelieferte freitextliche Beschreibung der Auflage/Ausgabe verschiedenen Formalismen entstammt.

Erscheinungsjahr

Ebenso häufig wie die Bevorzugung von neueren Auflagen wurde die Hervorhebung von neueren Titeln als Wunsch an das Trefferranking genannt.
Hier eine optimale Gewichtung zu finden ist problematisch, da die Zahl der möglichen Erscheinungsjahre sehr groß ist, die Gewichtung aber nicht zu stark differenziert ausfallen darf. Eine zu starke Differenzierung würde dazu führen, dass dieses Kriterium alle anderen Faktoren in den Hintergrund drängt. Dies ist schon allein deshalb nicht sinnvoll, weil der OPAC auch eine separate auf- und absteigende Treffersortierung nach Erscheinungsjahr bietet. Wer also tatsächlich an der neuesten Literatur interessiert ist, sollte diese Funktion nutzen.
Billigt man dem Erscheinungsjahr jedoch nur einen geringen Einfluss zu, dann verliert sich der Effekt allein durch die Rundung bei der Rankingberechnung in Lucene und brandneue Medien erhalten kein höheres Gewicht als drei Jahre alte Titel.
Für den Heidelberger OPAC wurden deshalb zunächst folgende Boostfaktoren gewählt:

E-Jahr	Aktuelles Jahr	Vorjahr	2 Jahre	3 Jahre	4-6 Jahre	7-11 Jahre	12-16 Jahre
Boost	1,8	1,7	1,5	1,4	1,3	1,2	1,1

Abbildung 2

Ohne Zweifel ist dies nur ein Kompromiss, der auch der Tatsache geschuldet ist, dass der OPAC der Universitätsbibliothek Heidelberg das große geistes- und naturwissenschaftliche Fächerspektrum einer Volluniversität abdeckt. Für stärker spezialisierte OPACs (z.B. von technischen Hochschulen) lassen sich hier sicher passgenauere Einstellungen finden.

Online-Ressourcen

Online verfügbare Titel erhalten einen Boostfaktor von 1,5. Hiermit soll unter anderem sichergestellt werden, dass beispielsweise E-Journals, E-Books oder Digitalisate vor der gegebenenfalls parallel vorhandenen Papierausgabe angeboten werden. Dies stellt eine einfache Form der Berücksichtigung von Verfügbarkeitskriterien beim Ranking dar. Gleichzeitig ist es eine Möglichkeit, die elektronischen Medien zu bewerben.

Anzahl der besitzenden Bibliotheken im Campus

In der Annahme, dass ein Titel, der in mehreren Bereichs- und Institutsbibliotheken vorhanden ist, in der Regel häufiger nachgefragt ist, wird die Anzahl der besitzenden Bibliotheken derzeit nach folgender Einteilung in den Boostfaktor einberechnet:

Anzahl dezentrale Bibliotheken	> = 5	4	3	2	1
Boost	1,5	1,4	1,3	1,2	1,1

Abbildung 3

3.3 Feld-Boost bei der Indexierung

Geht man von der Annahme aus, dass bestimmte Kategorien (z.B. Titel- oder Schlagwortkategorien) stärker als andere (z.B. Informationen aus Kataloganreicherungen, wie Inhaltsverzeichnisse) über die Relevanz eines Treffers entscheiden, so führt dies zu einer Gewichtung der einzelnen Suchfelder.
Eine Möglichkeit zu einer solchen Gewichtung, die Lucene bietet, sind feldbezogene Boostfaktoren, die bereits bei der Indexierung angegeben werden können. Im OPAC der Universitätsbibliothek Heidelberg wurde auf diese Möglichkeit verzichtet und – wegen der größeren Flexibilität – eine Feldgewichtung über die Suchanfrage gewählt.

4 Dynamische Maßnahmen zur Anpassung des Rankings

Die in diesem Abschnitt beschriebenen Maßnahmen beeinflussen das Ranking erst bei der Suche. Sie können bei einem bestehenden Index jederzeit verändert werden.

4.1 Feld-Boost bei der Suche

Der DefaultQueryParser von Lucene erlaubt die Definition eines Standardfeldes für den Fall, dass in der Nutzeranfrage kein Suchfeld angegeben ist. In einer ersten Fassung des Heidelberger OPACs wurden die Inhalte der Felder, die bei der Einfachen Suche gemeinsam durchsucht werden sollen (Autor, Titel, …), zusätzlich gemeinsam in einem Suchfeld „Freitext" indexiert. Dieses Suchfeld wurde als Standardfeld für den

8 Beispiel: 8.Aufl.: **LB-F 7-24829::(8)**; 9.Aufl.: **LB-F 7-24829::(9)**

DefaultQueryParser festgelegt. Nachdem der Wunsch nach einer unterschiedlichen Gewichtung der beteiligten Felder innerhalb dieses Sammelfelds aufkam, musste die Vorgehensweise angepasst werden. Mittels des MultiFieldQueryParsers von Lucene wird in der aktuellen OPAC-Version auch bei der feldübergreifenden Suche eine Aufteilung auf mehrere Indexfelder vorgenommen. Suchworte, die sich nicht auf ein konkretes Suchfeld beziehen, können mehreren Standardfeldern zugeordnet werden, wobei jedem dieser Felder ein eigener Feldboost zugeteilt werden kann. Der MultiFieldQueryParser übernimmt dabei selbst die Erstellung der korrekten boole'schen Anfrage. Die Verwendung des MultiFieldQueryParsers ermöglicht so die Änderungen der Feld-Boosts zur Laufzeit. Dies ist dann unabdingbar, wenn der Nutzer selbst über die Gewichtung der Felder entscheiden kann.[9] Auch für die Ermittlung einer geeigneten Standardfeldgewichtung ist die schnelle Anpassungsmöglichkeit von Vorteil.
Die feldübergreifende Freitextsuche ist beim Heidelberger OPAC auf folgende Suchfelder beschränkt:

- Titel (enthält sämtliche Titelkategorien)
- Autor (enthält alle Personendaten mit Ansetzungs- und Verweisungsformen)
- Körperschaft (enthält alle Körperschaften mit Ansetzungs- und Verweisungsformen)
- Schlagwort (enthält alle Schlagworte jeweils mit Ansetzungs- und Verweisungsformen)
- Erscheinungsjahr
- ISBN/ISSN
- Volltexte aus Kataloganreicherungen.

Andere Felder – wie beispielsweise der Verlagsort – wurden nicht in die feldübergreifende Suche aufgenommen, da dies bei Suchbegriffen mit Ortsnamen (z.B. Heidelberg) in der Regel zu sehr vielen falsch positiven Treffern führt.
Für die Gewichtung beim Trefferranking wurde bislang nur eine grobe Einteilung der Felder vorgenommen:

- Suchfeld ‚freitext' (enthält Titel, Autor, Körperschaft, Schlagwort, Erscheinungsjahr, ISBN/ISSN): Faktor 1,0
- Suchfeld ‚1w' (bei sogenannten Einworttiteln, wie z.B. ‚Nature', ‚¬Die¬ Zeit <Hamburg>', wird dieses Feld zusätzlich zum Titelfeld bestückt.): Faktor 2,0
- Suchfeld ‚exttext' (enthält Volltexte aus Kataloganreicherungen): Faktor 0,5.

Einworttitel werden so bei der Eingabe des entsprechenden Suchbegriffs bevorzugt. Suchbegriffe, die nur in den Kataloganreicherungen (z.B. Inhaltsverzeichnisse oder Klappentexte) gefunden werden, ergeben ein schlechteres Ranking.
Auf eine differenziertere Unterscheidung der mit Faktor 1,0 berücksichtigten Felder wurde im OPAC der Universitätsbibliothek Heidelberg zunächst verzichtet.[10]

Der folgende Code-Ausschnitt zeigt die Einbindung des MultiFieldQueryParsers im Heidelberger OPAC:

```
HashMap<String, Float> boost = new HashMap<String, Float>();

//Die einzelnen Felder werden mit Boostfaktoren belegt, die beim Start
//des Backend-RPC-Servers von der Kommandozeile gelesen werden
//HashMap<String = Feldname, Float = Boost>
boost.put(„freitext", (Float.valueOf(this.freitext_boost)));
boost.put(„exttext", (Float.valueOf(this.exttext_boost)));
boost.put(„1w", (Float.valueOf(this.EinW_boost)));
```

9 Im Prototyp des Lucene-OPACs der National Library of Australia (NLA) (7, 11) ist dies realisiert.
10 Ein Beispiel für eine solche Differenzierung findet sich im Prototyp des Lucene-OPACs der National Library of Australia (NLA) (7, 11)).

```
String[] fields = {„freitext“, „exttext“, „1w“};

//Erzeugen einer MultiFieldQueryParser-Instanz
MultiFieldQueryParser qP =
        new MultiFieldQueryParser(fields, AnalyzerWrapper, boost);

qP.setDefaultOperator(MultiFieldQueryParser.AND_OPERATOR);
```

Die vom Benutzer eingegebene Suchanfrage wird somit beim Parsen der Suchanfrage (im Java-Backend) entsprechend erweitert.

Beispiel:

Benutzeranfrage:
British history 1815-1914

Geparste Anfrage:

```
+(freitext:british exttext:british^0.5
1w:british^2.0)
+(freitext:history exttext:history^0.5
1w:history^2.0)
+(freitext: 1815 1914  exttext: 1815
1914 ^0.5 1w: 1815 1914 ^2.0)
```

Weitere Rankingmaßnahmen im Frontend (siehe Abschnitt „Phrasensuche“) sorgen allerdings dafür, dass diese Anfrage so tatsächlich nur als Teil einer komplexeren Struktur auftritt.
Etwaige vom Benutzer angegebene Boostfaktoren für einzelne Suchworte oder -phrasen bleiben erhalten und multiplizieren sich entsprechend mit den Feld-Boosts.
Sehr hilfreich für die Feinabstimmung der Gewichte ist die vom Lucene-Framework für die Errechnung des Rankings zur Verfügung gestellte Explain-Funktion. Diese gibt für jeden Treffer einer Suchanfrage die Zwischenergebnisse der Rankingformel aus.[11]

4.2 Phrasensuche

Werden bei einer Suchanfrage mehrere Suchbegriffe verwendet, so sollen die Titel bevorzugt werden, in denen diese Suchbegriffe als Phrase oder zumindest nahe beieinander vorkommen. Dazu wird im Heidelberger OPAC die Suchanfrage bereits im Frontend modifiziert. Dies erfolgt sowohl feldübergreifend bei der Einfachen Suche als auch feldspezifisch bei der Feldsuche. Die Suche nach den Einzelbegriffen wird dabei um die Suche nach der exakten Phrase und die Suche nach einer ähnlichen Phrase (Phrase mit Slop[12]-Wert 2) ergänzt. Treffer mit der exakten Phrase werden mit den Boost-Faktor von 6,0 gewertet, Treffer mit der ähnlichen Phrase mit einem Boost-Faktor von 3,0.

Benutzeranfrage (einfache, feldübergreifende Suche):
British history 1815-1914

Suchergänzung Phrasensuche im Frontend:

```
(British history 1815-1914) OR  British history 1815-1914 ^6 OR
 British history 1815-1914 ~2^3
```

Geparste Anfrage:

```
(
//        Einzelfelder
          +(freitext:british exttext:british^0.5 1w:british^2.0)
          +(freitext:history exttext:history^0.5 1w:history^2.0)
          +(freitext: 1815 1914  exttext: 1815 1914 ^0.5 1w: 1815 1914 ^2.0)
)
(
//        exakte Phrase in den verschiedenen Feldern mit Boost 6.0
          (
                    freitext: british history 1815 1914
                    exttext: british history 1815 191 4 ^0.5
                    1w: british history 1815 1914 ^2.0
          )^6.0
)
(
//        ähnliche Phrase in den verschiedenen Feldern mit Slop-Wert 2 und Boost 3.0
          (
                    freitext: british history 1815 1914 ~2
                    exttext: british history 1815 1914 ~2^0.5
                    1w: british history 1815 1914  ~2^2.0
          )^3.0
)
```

5 Aussicht

Die Arbeiten am Relevanz-Ranking des Heidelberger OPACs sind noch nicht abgeschlossen. Mit den dargestellten Maßnahmen wurden „Stellschrauben“ für das Ranking eingeführt, die es weiter zu optimieren gilt. Auch wenn mit den gewählten Einstellungen bereits beträchtliche Verbesserungen der Ergebnisse erzielt wurden, steht eine Optimierung auf Basis von detaillierten Untersuchungen noch aus.
Einige Faktoren, die beim Ranking berücksichtigt werden könnten, wurden bislang noch nicht implementiert. Dies sind hauptsächlich Faktoren, die sich auf das „popularity based ranking“ beziehen und das Nutzerverhalten abbilden:

- Anzahl der Ausleihen zu einem Titel
  Eine solche Maßnahme ist allerdings auch mit Nachteilen verbunden: Mit einer Höhergewichtung wird die ohnehin stark genutzte Literatur noch weiter in den Vordergrund gerückt, wenig genutzte noch mehr in den Schatten gestellt. Schließlich kann ein Suchergebnis, bei dem die obersten Listeneinträge immer entliehen sind, den Nutzer auch frustrieren.
- Verfügbarkeitsstatus
  Vorstellbar ist, dass verfügbare, nicht entliehene Medien höher gewichtet werden. Allerdings wäre durch die schnelle Veränderlichkeit der Faktoren eine Trefferliste häufig nicht rekonstruierbar.
- Anzahl der Exemplare eines Titels
- Nutzerbewertungen und (z.B. bei der Suche nach ähnlichen Titeln) statistisch ermittelte Empfehlungen

Ebenso ist die Einbeziehung eines persönlichen Nutzungsprofils denkbar.
Auf eventuelle Widersprüche zwischen den verschiedenen Gewichtungsmöglichkeiten gilt es zu achten. So sind beispielsweise Titel, die eine hohe Anzahl von Ausleihen haben (höhere Gewichtung) zwangsläufig häufiger nicht sofort verfügbar (niedrigere Gewichtung).
Selbstverständlich kann das Ranking nur aufgrund der bestmöglichen Interpretation der eingegebenen Suchanfrage nach den zugrunde gelegten Kriterien erfolgen. Je besser der Recherchekontext des Nutzer ermittelt werden kann, desto zutreffender wird das Relevanz-Ranking ausfallen. Zur Analyse des Recherchekontextes ist von der Berücksichtigung von Nutzerprofilen bis hin zur Auswertung von Suchanfragen mittels semantischer Modelle (4) vieles vorstellbar. Welche Treffer tatsächlich „relevant“ sind, wird letztendlich immer der menschlichen Beurteilung vorbehalten bleiben. Das Ranking soll und kann die Recher-

11 Ein Beispiel für eine solche Ausgabe der Explain-Funktion findet sich beispielsweise in (11).
12 In Lucene wird die Proximity-Suche über den sogenannten Slop gesteuert: Im Slop wird die maximal erlaubte Anzahl von Worten angegeben, die zwischen den Suchbegriffen vorhanden sein darf. Über den Slop kann auch eine gegebenenfalls abweichende Reihenfolge von Suchbegriffen in den den Trefferdokumenten abgefangen werden.

che erleichtern, zur Treffereingrenzung und thematischen Einschränkung sind weitere Methoden, wie beispielsweise der Drill-Down komplementär einzusetzen.

Literaturhinweise

1. Abel-Kops, C.P., 2008. „Just where's the damn book? or, Rediscovering the art of cataloging". URL: http://eprints.rclis.org/12940/ [Zugriff am 26.11.2009].

2. Antelman, K., Lynema, E. & ace, A.K., 2006. „Toward a 21st Century Library Catalog". URL: http://eprints.rclis.org/7332/ [Zugriff am 26.11.2009].

3. Bade, D., 2007. „Relevance ranking is not relevance ranking or, when the user is not the user, the search results are not search results". *Online Information Review*, 31(6), 831-844

4. Bai, J. & Nie, J., 2008. „Adapting information retrieval to query contexts". *Information Processing & Management*, 44(6), 1901-1922.

5. Campbell, D.G. & Fast, K.V., 2004. „Panizzi, Lubetzky, and Google: How the Modern Web Environment is Reinventing the Theory of Cataloguing". *Canadian Journal of Information & Library Sciences*, 28(3), 25-38.

6. Dellit, A., Fitch, K. 2007. „Rethinking the catalogue", a paper delivered by Alison Dellit and Kent Fitch to the NLA Innovative Ideas Forum, 19 April 2007. *National Library of Australia Staff Papers, 2007* URL: http://www.nla.gov.au/openpublish/index.php/nlasp/article/view/1047/1316. [Zugriff am 26.11.2009]

7. Dellit, A., Boston, T. 2007. „Relevance ranking of results from MARC-based catalogues: from guidelines to implementation exploiting structured metadata". *National Library of Australia Staff Papers, 2007* URL: http://www.nla.gov.au/openpublish/index.php/nlasp/article/view/1052/1321. [Zugriff am 26.11.2009]

8. Lewandowski, D., 2009. „Spezialsuchmaschinen". URL: http://eprints.rclis.org/15516/ [Zugriff am 26.11.2009].

9. Markey, K. 2007. „The Online Library Catalog: Paradise Lost and Paradise Regained?" URL: http://www.dlib.org/dlib/january07/markey/01markey.html [Zugriff am 26.11.2009].

10. Oberhauser, O. & Labner, J., 2003. „Relevance Ranking in Online-Katalogen: Informationsstand und Perspektiven". URL: http://eprints.rclis.org/7224/ [Zugriff am 26.11.2009].

11. „Set of rules for Lucene relevance ranking". URL: http://ll01.nla.gov.au/docs/LuceneRRNotes.html [Zugriff am 26.11.2009].

AUTOREN

Annette Langenstein
langenstein@ub.uni-heidelberg.de

Leonhard Maylein
maylein@ub.uni-heidelberg.de
Informationstechnologie und DV-Anwendungen
Universitätsbibliothek
Plöck 107-109
69117 Heidelberg