

METHODEN FÜR DAS RANKING IN DISCOVERY- SYSTEMEN

Manuel Hora

Universitätsbibliothek Stuttgart / Bibliotheksakademie Bayern

manuel.hora@tum.de

1. Relevanzranking steuert die Nutzung

Wissenschaftliche Bibliotheken setzen vermehrt Discovery-Systeme ein, mit denen Nutzer das Medienangebot durchsuchen können. Eine offensichtliche Neuerung gegenüber traditionellen OPACs ist die Ausweitung des durchsuchbaren Bestands, sodass neben Monographien z. B. auch Aufsätze gefunden werden können. Um die so drastisch vergrößerte Titelauswahl beherrschbar zu machen, bieten Discovery-Systeme eine Sortierung nach Relevanz an. Die technische Grundlage dafür ist der Schritt von den relationalen Datenbanken des OPACs hin zur Verwendung von Suchmaschinentechnologie.¹

Dieser Wandel verändert nicht nur, wie wir suchen, sondern natürlich auch, was wir finden. Eine Studie ergab, dass Nutzer zur Beantwortung der gleichen Fragen bei Verwendung verschiedener Discovery-Systeme unterschiedliche Medientypen verwendeten.² Nutzer des EBSCO Discovery Service (EDS) rezipierten überwiegend Aufsätze, Nutzer von Summon zitierten auffällig viele Zeitungsbeiträge. Nutzer von Google Scholar verwendeten mehr Monographien als EDS- oder Summon-Nutzer. In dieser Studie spielten sicherlich auch der lokale Bestand und die Konfiguration der Discovery-Systeme eine Rolle. Trotzdem deutet sich der Einfluss des Rankings auf wissenschaftliche Arbeiten an.

¹ Jansen (2014, S. 774).

² Asher, Duke & Wilson (2013, S. 471).

Vor diesem Hintergrund ist es wichtig, die Kriterien zu kennen, anhand derer Relevanz ermittelt wird. Dies ist auch unter dem Gesichtspunkt der Informationskompetenz wichtig.³ Die kommerziellen Anbieter machen durchaus einige Angaben zu den Methoden ihres Rankings.⁴ Weitere Informationen können mittels Reverse Engineering gewonnen werden.⁵ Dazu werden Trefferlisten gezielt auf bestimmte Parameter untersucht, um deren Einfluss auf das Ranking zu bestimmen. Technische Details lassen sich vor allem aus den Open Source Lösungen lernen. Wichtig sind hier die weit verbreitete Suchmaschinenprogramm-bibliothek Lucene⁶ und ihre Derivate Solr und ElasticSearch. Diese sind nicht nur Basis für das kommerzielle Discovery-System Summon⁷ oder das nichtkommerzielle BASE⁸, sondern auch für nichtbibliothekarische Anwendungen wie Wikipedia⁹.

In dieser Studie soll ein Überblick über technische Grundlagen des Rankings in Discovery-Systemen gegeben werden. Sofern Informationen verfügbar sind, wird auch auf Methoden in den kommerziellen Angeboten EDS, Primo und Summon eingegangen.

2. Komponenten eines Discovery-Systems

Zunächst wird ein Überblick über die Komponenten einer Suchmaschine bzw. eines Discovery-Systems gegeben.¹⁰ Eine grobe schematische Darstellung findet sich in Abbildung 1. Alle Dokumente müssen indexiert werden. Dazu gehört, dass Metadaten, z. B. Autor und Titel, im Index erfasst werden. Falls vorhanden, kann auch der Volltext indexiert werden. Dabei wird für jedes Dokument erfasst, welche Wörter, auch als Terme bezeichnet, vorkommen, wie oft sie enthalten sind und oft auch ihre Positionen im Text.

Oft werden die Terme für die Indexierung prozessiert. So können beispielsweise Komposita in einzelne Terme zerlegt werden.¹¹ Flexionen können

³ Niedermaier (2014, S. 123).

⁴ EBSCO, ExLibris (2015), ProQuest (2013).

⁵ Beel & Gipp (2009).

⁶ Apache Software Foundation: Lucene.

⁷ Natarajan (2016, S. 128).

⁸ BASE.

⁹ Horohoe (2014).

¹⁰ Langville & Meyer (2006, S. 9–24).

¹¹ Maylein & Langenstein (2013, S. 193–194).

durch Stemming abgeschnitten werden oder die Terme werden durch Lemmatisierung in eine unflektierte Form überführt. Stemming und Lemmatisierung erhöhen die Treffermenge, allerdings auf Kosten der Präzision. Insbesondere in der englischen Sprache ist der Nutzen in Relation zu den Kosten oft überraschend gering, während das deutschsprachige Retrieval von beiden Verfahren deutlich profitiert.¹²

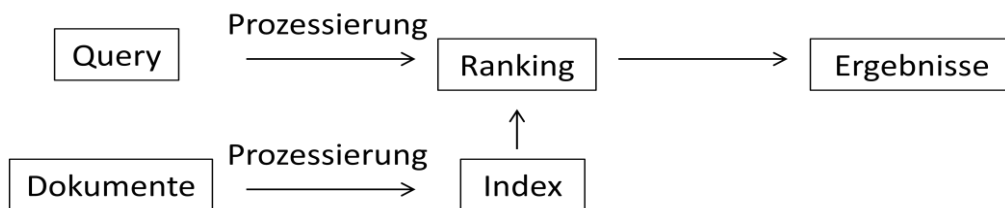


Abb. 1: Komponenten einer Suchmaschine

Wenn ein Nutzer eine Anfrage (oder Query) abschickt, wird diese auf ähnliche Art prozessiert, auch hier findet eine Zerlegung in Terme statt. Die Query kann in diesem Schritt auch erweitert werden, z. B. um Synonyme oder Übersetzungen. Die prozessierte Query wird schließlich mit dem Index abgeglichen. Je nach Umfang des Indexes werden entweder alle Dokumente nach ihrer Ähnlichkeit zur Query gerankt oder es findet erst eine Vorauswahl statt und die in Frage kommende Teilmenge wird gerankt. Das Ergebnis ist eine Liste an Treffern, die nach Ranking Scores geordnet dem Nutzer angezeigt wird.

3. Modelle für das Text-Matching

Das Text-Matching stellt die Grundlage des Rankings dar. Hier wird überprüft, wie hoch die inhaltliche Übereinstimmung zwischen Query und Indexeinträgen ist. Dafür können unterschiedliche Bewertungsmodelle verwendet werden. Im Folgenden werden das Boolesche Modell als einfache Grundlage sowie die beiden weit verbreiteten Methoden des Vektormodells und BM25 vorgestellt.

¹² Manning, Ragahvan & Schütze (2009, S. 33-34, 46).

3.1 Boolesches Modell

Beim Booleschen Modell wird in der Query mit den Operatoren AND, OR und NOT definiert, welche Terme vorkommen müssen oder nicht vorkommen dürfen.¹³ Ein Ranking findet nicht statt, ein Dokument kann die Anfrage nur erfüllen oder nicht erfüllen. Somit kann das Boolesche Modell allenfalls als Vorauswahl verwendet werden, um die Treffermenge vor dem Berechnen eines Rankings mit einem anderen Modell einzugrenzen.

3.2 Vektormodell

Das Vektormodell ist ein klassisches, aber immer noch verwendetes Verfahren für das Text-Matching. Lucene hat bis 2016 das Vektormodell als Standardeinstellung verwendet.¹⁴ Grundlage dieses Modells ist, dass jedes Dokument als Vektor beschrieben wird, wobei alle Terme im gesamten Index in dem Vektor erfasst werden.¹⁵ Der Wert für jedes Element ergibt sich oft aus dem Produkt von Term Frequency (TF) und Inverse Document Frequency (IDF), zusammen oft abgekürzt als TFIDF.¹⁶ Die TF gibt an, wie oft ein Term im Dokument enthalten ist, häufigere Nennungen führen also zu einem besseren Platz im Ranking. Die IDF misst, in wie vielen Dokumenten des gesamten Suchraumes der Term vorkommt. Insgesamt selten verwendeten Wörtern wird somit ein höheres Gewicht für das Ranking gegeben.

Analog zu den Dokumenten kann auch die Query als Vektor aufgefasst werden. Jeder Term, der in der Query enthalten ist, bekommt den Wert 1, jeder nicht enthaltene Term eine 0. Um einen Score für ein Dokument in Bezug auf die Query zu ermitteln, wird eine Metrik benötigt, die den Abstand zwischen Query-Vektor und Dokument-Vektor misst. Üblich ist hier, das Skalarprodukt beider Vektoren zu berechnen, d. h. die Summe über die elementweisen Produkte. Nachdem das Skalarprodukt anschaulich dem Cosinus des Winkels zwischen zwei Vektoren entspricht, ist die Bezeichnung „Cosinus-Ähnlichkeit“ zwischen Query und Dokument gebräuchlich.

¹³ Langville & Meyer (2006, S. 5–6).

¹⁴ Apache Software Foundation: Lucene.

¹⁵ Turnbull & Berryman (2016, S. 61–64).

¹⁶ Turnbull & Berryman (2016, S. 67).

Zusammenfassend bewertet dieses Modell Dokumente als relevant, wenn insgesamt seltene Suchterme in dem Dokument häufig vorkommen. Beachtenswert ist, dass bei dieser Wertung Dokumente auch dann einen hohen Score bekommen können, wenn sie einzelne Suchterme gar nicht enthalten.

3.3 BM25

BM25 ist ein Vertreter der Familie der probabilistischen Relevanzmodelle und seit Lucene Version 6.0 die Standardeinstellung für die Relevanzbewertung.¹⁷ Eine detaillierte Herleitung der Formel für BM25 findet sich bei Robertson¹⁸, hier soll auf einige Eigenschaften der Formel eingegangen werden.

$$\text{Score} = \sum_{T_i \in \text{Query}} \frac{\text{TF}_i}{k \left((1 - b) + b \frac{\text{DL}}{\text{AVGDL}} \right) + \text{TF}_i} \lg \frac{N - n_i + 0,5}{n_i + 0,5}$$

mit T_i als alle Terme in der Query, TF_i als Term Frequency, k und b als empirischen Parametern, DL als Document Length, AVGDL als Average Document Length, N als Anzahl aller Dokumente im Index und n_i als Anzahl aller Dokumente, die T_i enthalten.

Der letzte Faktor, der Logarithmus, ist ähnlich zur IDF, es werden also wieder seltene Terme höher gewichtet. DL und AVGDL werden verwendet, um zu kompensieren, dass längere Dokumente mehr verschiedene Terme enthalten und die einzelnen Terme häufiger vorkommen. Ohne so eine Normierung würden längere Dokumente pauschal bevorzugt werden. Der Parameter b justiert, wie stark die Dokumentlänge normalisiert werden soll und liegt meistens zwischen 0,5 und 0,8. Der Vollständigkeit halber sei erwähnt, dass auch TFIDF -Werte für das Vektormodell in Bezug auf die Länge normiert werden können,¹⁹ der Parameter b in BM25 erlaubt allerdings eine feinere Justierung.

Der große Quotient im ersten Faktor gibt dem Score als Funktion von TF einen sigmoidalen Verlauf. Dies ist wünschenswert, um den Einfluss der TF abzuschwächen (siehe Abbildung 2). Beim klassischen TFIDF -Modell skaliert der Score linear mit der TF , dies kann zu Verzerrungen führen. Ob ein Suchterm 20 mal

¹⁷ Apache Software Foundation: Lucene.

¹⁸ Robertson (2010, S. 340–345).

¹⁹ Manning et al. (2009, S. 121).

oder 50 mal im Volltext vorkommt, sollte für die Relevanz keine Rolle spielen. Aus diesem Grund verwendet z. B. Lucene auch im Vektormodell nicht direkt die TF, sondern deren Wurzel.²⁰ Die Wurzelfunktion ist allerdings immer noch nach oben unbeschränkt, während die sigmoidale Funktion von BM25 einen Grenzwert nicht überschreiten kann. Der Parameter k kontrolliert, wie schnell die Sättigung erreicht wird und liegt meistens zwischen 1,2 und 2,0.

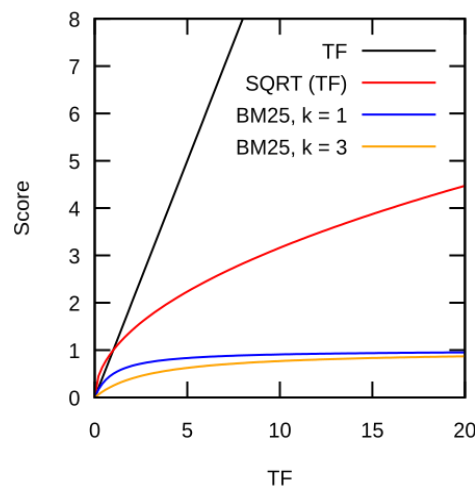


Abb. 2: Einfluss der TF auf den Score in verschiedenen Modellen

TF zeigt die lineare Abhängigkeit nach klassischem TFIDF Modell, SQRT (TF) ist eine abgeschwächte Variante auf Basis der Wurzelfunktion. Bei BM25 gibt es einen Sättigungswert, der nicht überschritten werden kann. Der Parameter k kontrolliert, wie schnell die Sättigung erreicht wird.

4. Text-Matching von Metadaten und Volltexten

Die bisherige Diskussion von Ranking-Modellen ist vom Text-Matching mit Volltexten ausgegangen. Diese liegen bei bibliothekarischen Beständen allerdings oft nicht in digitaler Form vor. Stattdessen werden die Medien mit strukturierten Metadaten beschrieben. Diese Metadaten haben auf das Ranking tatsächlich größere Auswirkungen als die Volltexte.

²⁰ Turnbull & Berryman (2016, S. 68).

4.1 MARC-Daten und Suche in verschiedenen Feldern

Die National Library of Australia hat Empfehlungen für das Ranking auf Basis von MARC Daten veröffentlicht.²¹ Höchste Priorität haben demnach Titel, Autor und Schlagwörter. Für den EBSCO Discovery Service ist bekannt, dass folgende Felder in absteigender Reihenfolge gewichtet werden: normierte Schlagwörter, Titel, vom Autor vergebene Schlagwörter, Abstract, Autor, Volltext.²²

Für das Text-Matching in den einzelnen Metadatenfeldern können jeweils unterschiedliche Kriterien definiert werden. Der Abgleich mit dem Autorenfeld ist eine ja/nein-Frage, bei der Bewertung des Titels ist die IDF hilfreich, im Abstract oder Inhaltsverzeichnis auch die TF und im Volltext ist eine Längennormalisierung sinnvoll. Außerdem ist es beim Autor nicht sinnvoll, Stemming oder Lemmatisierung zu verwenden, sondern höchstens die Edit Distance²³, um potentielle Rechtschreibfehler in der Query zu kompensieren. Bei allen Modifikationen der Query (z. B. Stemming) sollten Treffer auf die unmodifizierten Terme höher gewichtet werden als andere.²⁴

Bei einem Vergleich von Metadateneinträgen und Volltextverfügbarkeit bei Dokumenten aus unterschiedlichsten Quellen gibt es naturgemäß Heterogenitäten in der Qualität und Vollständigkeit.²⁵ So treffen z. B. Katalogisate deutscher Print-Monographien in Discovery-Systemen auf fremdsprachige elektronische Aufsätze, die nicht mit der GND erschlossen wurden.²⁶ Diese Heterogenität ist eine Herausforderung für ein Ranking. Für unterschiedliche kontrollierte Vokabularien ist ein Mapping sinnvoll, wie es in EDS verwendet wird.²⁷ Ein Lösungsweg für die unterschiedliche Qualität und Vollständigkeit der Daten besteht darin, getrennte Ergebnislisten zu verwenden. Häufig gibt es auf der Ergebnisseite einen Reiter „Bücher & mehr“, in welchem Treffer lokaler Katalogisate angezeigt werden. Daneben gibt es dann einen weiteren Reiter „Artikel & mehr“ für Treffer aus dem zentralen Index des Anbieters des Discovery-Systems. Der erste Reiter zeigt somit

²¹ Dellit & Boston (2007, S. 6).

²² EBSCO.

²³ Manning et al. (2009, S. 58–60).

²⁴ Dellit & Boston (2007, S. 6).

²⁵ Sunckel, Reh & Nienerza (2014, S. 787).

²⁶ Wiesenmüller (2012, S. 10).

²⁷ EBSCO.

Ergebnisse mit homogeneren Metadaten, der zweite zeigt den größeren Suchraum. Ob solch eine Aufteilung auf getrennte Listen sinnvoll ist und den Erwartungen der Nutzer entspricht, ist noch Gegenstand der Diskussion.²⁸

4.2 Abstand zwischen Suchtermen, Phrasensuche und exakte Treffer

Wenn ein Nutzer nach mehreren Termen in einer Query sucht, erwartet er Treffer, die diese Terme im Kontext behandeln und nicht isoliert voneinander. Ein Indiz dafür ist, wenn die gesuchten Terme nahe beieinander liegen. Um dies zu bestimmen, kann ein zweistufiges Verfahren verwendet werden: Zuerst werden klassisch alle Dokumente gesucht, welche die Suchterme enthalten. Anschließend werden die Positionsangaben im Index verglichen, die dafür natürlich beim Indexieren mit erfasst werden müssen.²⁹ Wenn nach einer Phrase gesucht werden soll, wird überprüft, ob die Positionsangaben aufeinanderfolgende Nummern sind. Wenn nur die Nähe zwischen zwei Termen bewertet werden soll, kann der Relevanzwert als Funktion des Abstands verbessert werden.³⁰ Der Ansatz, Positionsangaben im Index für Abstands- und Phrasensuchen zu verwenden, ist im Übrigen der Grund, warum Listen von Stopp-Wörtern inzwischen eher selten verwendet werden.³¹ Während früher Stopp-Wörter wie Artikel oder Präpositionen oft nicht indexiert wurden, um Speicherplatz zu sparen, hat sich dies inzwischen als zu problematisch für die Phrasensuche herausgestellt.

Die Phrasensuche wird in Suchmaschinen und Discovery-Systemen durch Anführungszeichen gekennzeichnet. Es kommt allerdings vor, dass Nutzer eine Phrase suchen, ohne sie zu kennzeichnen. In Lucene gibt es dafür die Möglichkeit, jede Query unaufgefordert auch als Phrase zu suchen.³² Das Vorhandensein aller einzelnen Terme hat hier zwar schon einen großen Einfluss auf das Ranking, ein nicht geforderter Phrasentreffer kann aber noch einen Boost für die Relevanz geben.

Eine weitere Steigerung ist es, wenn die Query ein Metadatenfeld vollständig abdeckt. So gibt es z. B. den gewöhnlichen Boost für Phrasentreffer, wenn mit der

²⁸ Jansen (2014, S. 779), Maylein & Langenstein (2013, S. 190).

²⁹ Turnbull & Berryman (2016, S. 35).

³⁰ Langenstein & Maylein (2009, S. 412).

³¹ Manning et al. (2009, S. 27).

³² Smiley & Pugh (2009, S. 132–133).

Query „eine kurze Geschichte“ der Titel „Eine kurze Geschichte der Zeit“ gerankt werden soll. Wird hingegen mit dem kompletten Titel gesucht, so dass es eine exakte Übereinstimmung zwischen Query und Titelfeld gibt, wird dies deutlich höher bewertet. Auch hier ist der konkrete Boost wieder feldabhängig. Dellit und Boston empfehlen einen hohen Boost für exakte Treffer oder Phrasen im Titel. Im Autorenfeld soll ein Phrasentreffer mit Vor- und Nachname immer noch belohnt werden, aber nicht so stark wie im Titel.³³

Das Argument für eine hohe Gewichtung von exakten Treffern und Phrasen ist, dass hier oft nach bereits bekannten Medien gesucht wird, im Gegensatz z. B. zur explorativen Suche. Exakte Treffer können hier auch helfen, um Titel, die nur aus einem Wort bestehen, im Ranking nach oben zu bringen, z. B. Goethes „Faust“.³⁴ Phrasentreffer können aber auch für das Matching von Schlagwort-Komposita relevant sein.³⁵

4.3 Zuordnung der Terme zu Feldern

Bei der einfachen Suche ist es eine große Herausforderung, die einzelnen Terme richtig den verschiedenen Metadatenfeldern zuzuordnen. Wird nach einem Namen gesucht, kann dieser Teil des Titels sein, Autor oder Herausgeber. Wenn in allen Feldern Treffer möglich sind, müssen diese sinnvoll zu einem Score zusammengefasst werden. Dafür gibt es zwei grundlegende Ansätze, die beide bestimmte Schwächen aufweisen.

Die feldzentrierte Suche testet die komplette prozessierte Query gegen jedes einzelne Feld.³⁶ Für jedes Feld wird dann unabhängig von den anderen Feldern ermittelt, welcher Term oder welche Terme den höchsten Feld-Score erzielen. Anschließend werden die Scores aller Felder aufaddiert. Ein Problem mit diesem Ansatz ist, dass bestimmte Terme für mehrere Felder gut passen könnten, während andere Terme schlechtere Scores erzielen. In diesem Fall könnten einzelne Suchterme komplett ignoriert werden, was nicht der Erwartung des Nutzers

³³ Dellit & Boston (2007, S. 6).

³⁴ Fricke (2012, S. 270).

³⁵ Plassmeier (2016a, S. 16)

³⁶ Turnbull & Berryman (2016, S. 118–136).

entspricht. Dieses Problem wird als „Albino-Elephant“ bezeichnet, in Anspielung auf seine erste Beschreibung.³⁷

Eine Lösung ist die termzentrierte Suche. Diese testet jeden einzelnen Term gegen jedes Feld.³⁸ Anschließend wird für jeden Term ein Score ermittelt und der Gesamtscore ergibt sich aus der Summe der Term-Scores, im Gegensatz zur Summe der Feld-Scores. Ein Problem der termzentrierten Suche ist, dass diese nicht für Phrasentreffer geeignet ist, da jeder Term für sich bewertet wird.

Eine einfache Lösung, um die Probleme beider Ansätze anzugehen, gibt es nicht.³⁹ Lösungen ergeben sich aus Kombinationen beider Verfahren, z. B. termzentrierter Suche mit zusätzlicher Phrasensuchfunktion.

5. Weitere Kriterien für das Relevanzranking

Bislang wurde nur behandelt, wie die Query verwendet werden kann, um Treffer zu ranken. Neben dem Ranking auf Basis des Text-Matchings gegen Metadaten und Volltext gibt es noch weitere Kriterien, die berücksichtigt werden können. Diese hängen oft nicht oder nicht direkt von der vom Nutzer formulierten Query ab, sondern vom jeweiligen Titel oder den Rahmenbedingungen der Recherche.

5.1 Aktualität

Dass die Aktualität eines Mediums einen Einfluss auf den Score hat, sollte nicht überraschen. Schließlich war die chronologische Sortierung in OPACs meist die Standardeinstellung.⁴⁰ In allen gängigen Discovery-Systemen werden neue Titel höher gerankt. Interessant wäre, fachspezifische Unterschiede zu berücksichtigen, ab wann ein Titel als veraltet gilt.⁴¹

5.2 Wissenschaftliche Seriosität

Ein weiteres weit verbreitetes Kriterium ist, wie gut die Art eines Dokuments wissenschaftlichen Anforderungen entspricht. EDS, Primo und Summon gewichten

³⁷ Turnbull & Berryman (2016, S. 140–141).

³⁸ Turnbull & Berryman (2016, S. 138–147).

³⁹ Turnbull & Berryman (2016, S. 162–169).

⁴⁰ Oberhauser (2010, S. 27).

⁴¹ Oberhauser (2010, S. 30).

Aufsätze, die ein Peer-Review durchlaufen haben oder Monographien höher als Zeitungsartikel und Rezensionen.⁴²

5.3 Art der Suche

Bei Anfragen, die nur aus wenigen Termen bestehen, ist es eine Herausforderung, zu erkennen, welche Art von Antwort der Nutzer erwartet. Oben wurde bereits die Bedeutung exakter Treffer für die Suche nach bekannten Titeln erläutert. Solche Suchen machen etwa 20 % der Anfragen an Discovery-Systeme aus.⁴³ Ein Spezialfall dieser Suche ist, wenn eine Quellenangabe unverändert aus einem Literaturverzeichnis kopiert wird. Primo und Summon verwenden beide spezielle Verfahren, um Angaben aus den üblichen Zitationsstilen richtig in Phrasen zu zerlegen und sie Suchfeldern zuzuordnen.⁴⁴

Die Mehrzahl aller Suchanfragen ist explorativ, d. h. der Nutzer sucht Informationen zu einem Thema, hat dabei aber noch keine konkrete Quelle im Sinn. Solche Anfragen können breit angelegt oder spezialisiert sein. Primo versucht, diese Anfragen durch die Auswertung von Log-Dateien bisheriger Nutzungen richtig einzuordnen. Bei unspezifischen Suchen werden dann Titel mit Überblicksinformationen höher gerankt.⁴⁵

5.4 Personalisierung

Die gerade beschriebene Auswertung von Log-Dateien zur Klassifikation von Queries geschieht anonym. Eine personalisierte Auswertung mit einem Nutzerprofil kann relevantere Ergebnisse präsentieren. Dies ist ein zweischneidiges Schwert. Dem potentiellen Nutzen stehen Bedenken wegen des Datenschutzes gegenüber. Auch der Nutzen selbst kann ins Gegenteil verkehrt werden, wenn es zum Filterblasen-Effekt kommt.⁴⁶ Bei diesem Effekt wird postuliert, dass Nutzer durch Personalisierung bevorzugt Suchergebnisse angezeigt bekommen, die ihre bisherige Meinung bestätigen. Ein gezieltes Experiment legt nahe, dass der Filterblasen-Effekt z. B. bei

⁴² EBSCO, ExLibris (2015, S. 4), ProQuest (2013, S. 4).

⁴³ Meadow & Meadow (2012, S. 171).

⁴⁴ ExLibris (2015, S. 2), ProQuest (2013, S. 3).

⁴⁵ ExLibris (2015, S. 4).

⁴⁶ Pariser (2011).

Google niedriger ist, als allgemein angenommen.⁴⁷ Trotzdem erstellen die Anbieter von Discovery-Systemen keine Profile auf Basis von individuellen Nutzungsdaten. Wenn man bedenkt, wie wichtig gerade in der Wissenschaft ein unvoreingenommener Blick und auch das Finden von fachfremden Zufallstreffern ist,⁴⁸ ist diese Entscheidung gut nachvollziehbar.

Primo bietet dem Nutzer an, Fachpräferenzen freiwillig im Account anzugeben, um passende Treffer höher zu ranken.⁴⁹ Der Nutzer kann auch seine Erfahrung in Form eines akademischen Abschlusses eintragen.⁵⁰ Im Nutzeraccount der UB Heidelberg ist es auch möglich, Fachpräferenzen zu nennen.⁵¹ Hier werden Treffer, deren klassifikatorische Erschließung zu den angegebenen Fachpräferenzen passen, mit einem Faktor 2 geboostet. Vorbildlich ist hier, dass Nutzer in der Trefferanzeige darauf aufmerksam gemacht werden, dass die Angaben in ihrem Account das Ranking beeinflusst haben und dass dies temporär deaktiviert werden kann. Eine andere Möglichkeit zur Personalisierung ist, den Nutzer Sprachpräferenzen im Account abspeichern zu lassen.⁵² Die bevorzugte Sprache aus den Einstellungen des verwendeten Webbrowsers auszulesen⁵³, beurteile ich allerdings kritisch. Bevorzugte Sprachen können beim allgemeinen Browsen und bei der wissenschaftlichen Recherche durchaus unterschiedlich sein. Die Möglichkeit zur sprachlichen Filterung durch Facetten sollte hier ausreichend sein.

5.5 Verfügbarkeit

Verfügbarkeitsbezogene Kriterien für das Ranking sind ein kontroverses Thema. Hier gilt es, abzuwägen zwischen dem Bedürfnis nach schnellem Zugriff und dem wissenschaftlichen Anspruch auf Vollständigkeit. Außerdem gilt es hier besonders, Präferenzen verschiedener Disziplinen Rechnung zu tragen.⁵⁴

Es ist möglich, elektronische Medien höher zu ranken. Dies geschieht im Katalog der UB Heidelberg mit einem Boostfaktor von 1,5. So sollen elektronische

⁴⁷ Jürgens, Stark & Magin (2014, S. 128).

⁴⁸ Quan-Haase & McCay-Peet (2014, S. 138).

⁴⁹ ExLibris (2015, S. 4–5).

⁵⁰ MeetPrimo: Personalized ranking.

⁵¹ Maylein & Langenstein (2013, S. 195–196).

⁵² Behnert & Lewandowski (2015, S. 19).

⁵³ Plassmeier (2016a, S. 24).

⁵⁴ Roscher (2014, S. 71–72).

Parallelausgaben vor den entsprechenden Printexemplaren gezeigt werden. Dies wird bewusst als Werbemaßnahme für E-Medien eingesetzt.⁵⁵ Denkbar ist auch, aus politischen Gründen Open Access Titel im Ranking zu bevorzugen. Die kommerziellen Discovery-Systeme bieten an, Titel aus lokalen Beständen oder Repositorien zu boosten.⁵⁶

Der Suchraum bei Monographien beschränkt sich häufig auf den lokalen Bestand. Dies ist keine Frage des Rankings, sondern des zu Grunde liegenden Indexes. Wird der Suchraum z. B. auf den Verbundkatalog ausgeweitet, könnten Titel niedriger gerankt werden, die nur per Fernleihe verfügbar sind.⁵⁷ Analog kann bei E-Ressourcen entweder der Suchraum auf lizenzierte und Open Access Dokumente beschränkt werden oder nicht lizenzierte Titel werden niedriger gerankt.

Im DFG-Projekt LibRank⁵⁸ wurden weitere Kriterien diskutiert, wie die Verfügbarkeit das Ranking beeinflussen könnte. So könnte kontextabhängig ein Titel niedriger bewertet werden, wenn er gerade ausgeliehen und entsprechend nicht verfügbar ist. Über IP-Adressen könnte des Weiteren ermittelt werden, ob der Nutzer in der Bibliothek oder von einem anderen Ort aus recherchiert. Entsprechend würden Print- oder E-Medien höher bewertet werden.⁵⁹

Diese Vorschläge sind interessant für den Fall, dass der Nutzer Informationen sofort benötigt. Allerdings muss betrachtet werden, an welche Zielgruppe sich das Discovery-System richtet. Für eine hoffentlich relevant große Gruppe von Studenten und vor allem Wissenschaftlern sollte wichtiger sein, zu wissen, was existiert, als ob es sofort und direkt verfügbar ist. Eine zu diesem Thema befragte Umwelthistorikerin formulierte treffend: „Für richtige Forschung hat die Verfügbarkeit keine Relevanz!“⁶⁰ Konsequenter weitergedacht, mit dem Ziel, dass Discovery-Systeme ein One-Stop-Shop für die Recherche sein sollen,⁶¹ müsste der Suchraum auf nicht lizenzierte Titel ausgedehnt werden. Dem gegenüber steht die Beobachtung, dass sich Nutzer schon jetzt von großen Treffermengen überfordert

⁵⁵ Langenstein & Maylein (2009, S. 410).

⁵⁶ EBSCO, ExLibris (2015, S. 5), ProQuest (2013, S. 4).

⁵⁷ Fricke (2012, S. 276).

⁵⁸ LibRank.

⁵⁹ Behnert & Lewandowski (2015, S. 17).

⁶⁰ Anonym, zitiert in Roscher (2014, S. 55).

⁶¹ Christine Baron (2015, S. 4).

fühlen.⁶² Meiner Meinung nach sollte das Thema Verfügbarkeit nicht in das Ranking einfließen. Stattdessen könnte es über verschiedene Reiter in der Trefferanzeige dargestellt werden. Eine ähnliche Möglichkeit bietet sich mit Facetten: Im Katalog der UB Heidelberg werden z. B. als Standardeinstellung nur verfügbare Artikel angezeigt. In der Facettierung gibt es dann eine Option, auch über den Heidelberger Bestand hinaus zu suchen.

5.6 Popularität

Für Popularität als Rankingfaktor gibt es ein prominentes Beispiel: Google. Der PageRank Algorithmus revolutionierte die Branche der Internetsuchmaschinen, indem er Links zwischen Websites berücksichtigte.⁶³ Die Analogie zu Links in der wissenschaftlichen Literatur sind Zitationen. Entsprechend sind Zitationszahlen auch tatsächlich ein zentraler Faktor im Ranking von Google Scholar.⁶⁴ Dies kann allerdings den Matthäus-Effekt verstärken, dem zu Folge vor allem die Autoren zitiert werden, die schon in der Vergangenheit oft zitiert wurden.⁶⁵

Von den großen kommerziellen Discovery-Systemen verwenden Primo und Summon Zitationszahlen für das Ranking.⁶⁶ Ex Libris bietet Bibliotheken des Weiteren an, auf Basis von Zugriffsstatistiken mit dem unternehmenseigenen Link Resolver SFX Vorschläge in Primo anzuzeigen.⁶⁷ D. h., Nutzer, die einen Titel anschauen, bekommen Vorschläge, welche Titel andere Nutzer angeschaut haben, die sich für denselben Titel interessierten.

Das LibRank-Projekt diskutierte, dieses Konzept auszuweiten. Für E-Medien könnten Zugriffszahlen berücksichtigt werden, die COUNTER-konform standardisiert wurden. Für Print-Bestände, zumindest für ausleihbare, könnten Ausleihzahlen in das Ranking einfließen. Behnert und Lewandowski erklären, dass die Berücksichtigung von Zitationen und Nutzung zwar viele theoretische Schwächen haben, in der Praxis aber gut funktionieren.⁶⁸

⁶² Roscher (2014, S. 50).

⁶³ Brin & Page (2012, S. 3827).

⁶⁴ Beel & Gipp (2009, S. 3–4).

⁶⁵ Merton (1988).

⁶⁶ ExLibris (2015, S. 4), ProQuest (2013, S. 4).

⁶⁷ Thompson (2014, S. 20).

⁶⁸ Behnert & Lewandowski (2015, S. 8).

Es gibt auch andere Wege, die Popularität zu messen, die weniger direkt von der Nutzung abhängen. Das Discovery-System der SuUB Bremen vergibt Boosts von jeweils 3 % für Titel, die in höheren Auflagen erschienen sind oder von denen mehrere Exemplare vorhanden sind.⁶⁹ Im ersten Fall wird davon ausgegangen, dass nur erfolgreiche Titel neu aufgelegt werden. Somit wird die Auflagenzahl als Qualitätskriterium interpretiert. Der zweite Fall, die Anzahl der Exemplare, berücksichtigt das Urteil der erwerbenden Fachreferenten. Im Fall von Universitätsbibliotheken mag dies noch nicht sehr aussagekräftig sein, da je nach Erwerbungsprofil häufig nur von Lehrbüchern Mehrfachexemplare erworben werden. Der Ansatz könnte aber auch überregional ausgedehnt werden, z. B. über die Anzahl der Bibliotheken in einem Verbund, die einen Titel erworben haben.⁷⁰

6. Kombination mehrerer Kriterien zu einem Score

Wenn klar ist, welche der dargestellten Kriterien in welcher Form für das Ranking berücksichtigt werden sollen, dann müssen alle Kriterien noch zu einem Score zusammengefasst werden. Dass die Kriterien mit direktem Bezug zur Query eine große Rolle spielen sollten, liegt auf der Hand. Das Auffinden von Suchtermen in den Metadaten und im Volltext stellt sicher, dass die Treffer einen Bezug zur Query haben. Wie hoch der Einfluss anderer Parameter sein soll und auf welche Art sie das Ranking beeinflussen, soll jetzt diskutiert werden.

Bei Lucene ist vorgesehen, Kriterien als additive oder multiplikative Boosts zu implementieren.⁷¹ Diese modifizieren dann den Score, der durch die grundlegende Rankingfunktion bestimmt wurde. In diesem Fall muss erfasst werden, in welchem Wertebereich sich ein Kriterium befinden kann. Häufig ist eine Glättung und Normalisierung der Werte sinnvoll. So könnten z. B. Zitationszahlen und Downloadzahlen auf einen ähnlichen Wertebereich normiert werden, um anschließend die relative Gewichtung der Faktoren transparenter zu machen.⁷²

Die Gewichtung wird dann iterativ im Rahmen ausführlicher Tests justiert. Dazu kann eine repräsentative Auswahl an Queries getestet werden. Die Reihenfolge

⁶⁹ Blenkle, Ellis, Haake & Zillmann (2015, S. 39).

⁷⁰ Behnert & Lewandowski (2015, S. 10).

⁷¹ Turnbull & Berryman (2016, S. 174).

⁷² Plassmeier, Borst, Behnert & Lewandowski (2015, S. 2).

der Treffer wird im einfachsten Fall intellektuell bewertet. Realistisch betrachtet ist für ein Discovery-System eine Automatisierung der Tests notwendig. Dafür können intellektuell vergebene Bewertungen für Paare aus Query und Position in der Trefferliste gespeichert werden, um damit verschiedene Gewichtungen zu testen.⁷³ Eine weitere Automatisierung dieses Prozesses verwendet Log-Dateien, um zu erkennen, welche Titel tatsächlich genutzt wurden. Auf dieser Basis können Machine Learning Techniken angewendet werden, um die Gewichtungen zu optimieren. Die Verwendung von Machine Learning zur Optimierung von Rankingverfahren wird als „Learning to Rank“ bezeichnet.⁷⁴

Bei dem gerade vorgestellten Verfahren mit optimierten, aber statischen Boosts ist es möglich, dass sich verschiedene Faktoren gegenseitig kompensieren.⁷⁵ Beispielsweise könnte ein Treffer mit schlechter Bewertung der Nutzung trotzdem hoch gerankt werden, wenn er einen hohen Boost für ein aktuelles Erscheinungsjahr bekommt.

Es sind andere Modelle denkbar, in denen so eine Kompensation nicht möglich ist. Bei der priorisierten Aggregation werden die Kriterien nach Wichtigkeit geordnet.⁷⁶ Dies könnte wie folgt aussehen:

$$\text{Priorität}_{\text{Text-Matching}} > \text{Priorität}_{\text{Aktualität}} > \text{Priorität}_{\text{Popularität}}$$

Die Gewichtungsfaktoren ergeben sich dann aus dem gewichteten Score des jeweils nächstwichtigeren Kriteriums. Für die Gewichtungsfaktoren γ ergibt sich dann:

$$\begin{aligned} \gamma_{\text{Text-Matching}} &= 1 \\ \gamma_{\text{Aktualität}} &= \gamma_{\text{Text-Matching}} \cdot \text{Score}_{\text{Text-Matching}} \\ \gamma_{\text{Popularität}} &= \gamma_{\text{Aktualität}} \cdot \text{Score}_{\text{Aktualität}} \end{aligned}$$

Ein weniger wichtiges Kriterium hat also weniger Einfluss, wenn das wichtigere Kriterium schlecht erfüllt wurde. Der Reiz dieses Ansatzes besteht einerseits darin, dass die willkürliche Gewichtung verschiedener Kriterien umgangen wird. Andererseits bietet dieses bislang wenig verbreitete, aber von LibRank

⁷³ Turnbull & Berryman (2016, S. 251–252).

⁷⁴ Li (2011), Turnbull & Berryman (2016, S. 252).

⁷⁵ Plassmeier (2016b, S. 5).

⁷⁶ da Costa Pereira, Dragoni & Pasi (2012, S. 9–11).

diskutierte Modell⁷⁷ die Möglichkeit, den Nutzer ad hoc sortieren zu lassen, welche Kriterien ihm wichtig sind.

7. Eignung weiterer Arten von Algorithmen

In diesem Abschnitt soll ein Überblick über eine kleine Auswahl an weiteren Arten oder Erweiterungen von Algorithmen gegeben werden. Diese werden unter anderem im Web-Retrieval eingesetzt.

7.1 Pseudo-Relevanzfeedback

Es wurde bereits angesprochen, dass die Query um Synonyme oder Übersetzungen erweitert werden kann. Das Pseudo-Relevanzfeedback greift dieses Konzept auf.⁷⁸ Klassisches Feedback basiert darauf, dass ein Nutzer in der Trefferliste relevante Treffer auswählt. Diese Informationen werden dann in einer neuen Anfrage verwendet, um eine optimierte Liste zu erhalten. Beim Pseudo-Relevanzfeedback werden diese beiden Suchen zumindest äußerlich zu einer zusammengefasst. Dazu wird eine erste Suche ausgeführt, mit der vom Nutzer formulierten Query. Die Trefferliste wird dem Nutzer nicht gezeigt. Es wird angenommen, dass z. B. die ersten zehn Treffer relevant sind. Häufige Terme in diesen Treffern können jetzt der Query hinzugefügt werden, damit wird die Suche wiederholt. Dieses Ergebnis wird dem Nutzer angezeigt. Im Großen und Ganzen ist diese Methode oft hilfreich, auch wenn sie manchmal die Anfrage in eine falsche Richtung driften lassen kann. Dem Autor sind bislang keine bibliothekarischen Anwendungen bekannt, die diese Methode verwenden. Interessant könnte es sein, mit Pseudo-Relevanzfeedback eine Query um passende normierte Schlagwörter zu ergänzen.

7.2 Semantische Methoden

Das Versprechen semantischer Ansätze ist, inhaltlich passendere Treffer zu liefern, indem nicht nur auf Stichwörter zurückgegriffen wird, die mitunter mehrdeutig sind. Dafür gibt es einerseits strukturierte Verknüpfungen von Konzepten, z. B. mit dem

⁷⁷ Plassmeier (2016a, S. 4–5).

⁷⁸ Manning et al. (2009, S. 187–188).

Resource Description Framework, kurz RDF.⁷⁹ Andererseits gibt es ausgefeilte Techniken, um ohne Rückgriff auf explizites Wissen passende Volltexte zu finden. Beispielsweise mit latenter semantischer Indexierung (LSI) können Volltexte gefunden werden, selbst wenn sie die Suchbegriffe nicht enthalten.⁸⁰ Die dafür nötigen semantischen Beziehungen werden hergeleitet, indem untersucht wird, welche Wörter im Textcorpus häufig nahe beieinander stehen. Die Methode ist rechenaufwändig, kann aber vollautomatisch erfolgen, d. h. es ist keine Form von intellektueller Erschließung nötig.

Diese und verwandte Techniken des Text Mining haben allerdings die offensichtliche Einschränkung, dass ein digital auswertbarer Volltext vorhanden sein muss. Um Dokumente, die dafür nicht in Frage kommen, nicht zu diskriminieren, sollten solche Methoden auf Anfragen beschränkt werden, bei denen der überwiegende Teil der zu erwartenden Treffer digital vorliegt. Denkbar wären semantische Ansätze somit z. B. für Anfragen, die auf moderne Aufsätze aus dem MINT-Bereich zielen.

7.3 Analyse der Query mit künstlicher Intelligenz

Auf Seiten der Dokumente sind Text Mining Ansätze aktuell problematisch, weil sie viele Printdokumente ausschließen würden. Stattdessen könnte allerdings mit semantischen Analysen der Query gearbeitet werden, um die Interessen der Nutzer besser zu erkennen. Hier kann von Bedeutung sein, in welcher Reihenfolge Terme eingegeben werden oder wie sie verknüpft sind, bis hin zur natürlichsprachlichen Suche. Google führte 2015 RankBrain ein, eine künstliche Intelligenz, um bislang unbekannte Suchanfragen zu verstehen.⁸¹ RankBrain scheint komplexe Queries in einfachere Anfragen umzuformulieren, die dann vom Ranking-Algorithmus bearbeitet werden können. Umso mehr sich Nutzer daran gewöhnen, auch natürlichsprachliche Anfragen an Suchmaschinen zu stellen, desto mehr sollten auch Bibliotheken ihre Discovery-Systeme auf solche Fragen vorbereiten.

⁷⁹ Horch, Kett & Weisbecker (2013, S. 17).

⁸⁰ Manning et al. (2009, S. 413–417).

⁸¹ Capala (2016).

8. Der Kontext des Rankings in Discovery-Systemen

Abschließend soll betrachtet werden, in welchen Kontext die Ranking-Verfahren innerhalb von Discovery-Umgebungen eingebettet sind. Das Ranking ist nur einer der Faktoren, die darüber bestimmen, welche Treffer die Nutzer finden können. Der Prozess beginnt bereits damit, dass die Nutzer eine Suchanfrage formulieren. Hierbei können Discovery-Systeme schon mit Vorschlägen zur Auto-Vervollständigung unterstützen und lenken.⁸² Die Auswahl von Vorschlägen basiert dabei selbst auf einem Ranking-Algorithmus.

Nachdem die Query abgeschickt und ausgewertet wurde, stellt die Präsentation der Treffer ein weiteres bedeutendes Element dar. Die Gruppierung in verschiedene Trefferlisten ist weit verbreitet. Innerhalb der Listen können Facetten nicht nur die Treffermenge eingrenzen, sondern den Nutzern auch einen schnellen Überblick geben, welche Themenbereiche gefunden wurden. Ein weiterer Schritt ist, sich von der linearen Darstellung in Trefferlisten zu lösen und Ergebnisse z. B. wie in Yewno als Mindmaps zu präsentieren.⁸³

Letztendlich braucht es die Kombination dieser Methoden, um die immer größer werdende Menge an Literatur für Nutzer beherrschbar zu machen. Wo sich aufgrund überbordender Trefferzahlen Überforderung breit macht,⁸⁴ braucht es technisch und inhaltlich angemessene Lösungen, um die Suche zu erleichtern. Dann können wir darauf hoffen, dass sich Nutzer nicht mit dem ersten Treffer zufrieden geben, sondern im Geiste guter wissenschaftlicher Praxis alle relevanten Quellen rezipieren.

⁸² Krallinger, Rabal, Lourenço, Oyarzabal & Valencia (2017, S. 7692).

⁸³ Yewno: Yewno Discover.

⁸⁴ Roscher (2014, S. 50).

Literatur

- Apache Software Foundation: Lucene.
<https://lucene.apache.org/>
(abgerufen am 25. 02. 2018).
- Asher, A. D., Duke, L. M. & Wilson, S. (2013). Paths of discovery: Comparing the search effectiveness of EBSCO Discovery Service, Summon, Google Scholar, and conventional library resources. *College & Research Libraries*, 74(5), 464–488. doi:10.5860/crl-374
- BASE. <https://www.base-search.net/about/de/faq.php>
(abgerufen am 25. 02. 2018).
- Beel, J. & Gipp, B. (2009). Google Scholar`s ranking algorithm: An introductory overview. In *Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISST'09)*. Rio de Janeiro.
- Behnert, C. & Lewandowski, D. (2015). Ranking search results in library information systems. *The Journal of Academic Librarianship*, 41(6), 725–735. doi:10.1016/j.acalib.2015.07.010
- Blenkle, M. et al. (2015). Nur die ersten Drei zählen! Optimierung der Rankingverfahren über Popularitätsfaktoren bei der Elektronischen Bibliothek Bremen (E-LIB). *o-bib*, 2, 33–42. doi:10.5282/o-bib/2015H2S33-42
- Brin, S. & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18), 3825–3833. doi:10.1016/j.comnet.2012.10.007
- Capala, M. (2016). Machine learning just got more human with Google's RankBrain.
<https://thenextweb.com/artificial-intelligence/2016/09/02/machine-learning-just-got-more-human-with-googles-rankbrain/>
(abgerufen am 25. 02. 2018).
- Christine Baron. (2015). DigiBib IntrOX. *Bibliotheksdienst*, 49(1), 4–13. doi:10.1515/bd-2015-0003
- da Costa Pereira, C., Dragoni, M. & Pasi, G. (2012). Multidimensional relevance: Prioritized aggregation in a personalized information retrieval setting. *Information Processing and Management*, 48(2), 340–357. doi:10.1016/j.ipm.2011.07.001
- 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*, (February), 1–14.
- EBSCO: How is relevance ranking determined in EBSCO Discovery Service (EDS)?
https://help.ebsco.com/interfaces/EBSCO_Discovery_Service/EDS_FAQs/relevance_ranking_determined_in_EDS
(abgerufen am 25. 02. 2018).
- ExLibris. (2015). Primo Discovery: Search, ranking, and beyond.
https://knowledge.exlibrisgroup.com/@api/deki/files/26778/Primo_Search_and_Ranking.pdf?revision=1
(abgerufen am 25. 02. 2018).
- Fricke, F. (2012). Relevance Ranking in elektronischen Bibliothekskatalogen (OPACs). *Informationswissenschaft*, 263–284.
- Horch, A., Kett, H. & Weisbecker, A. (2013). *Semantische Suchsysteme für das Internet*. Stuttgart: Fraunhofer Verlag.

- Horohoe, C. (2014). Wikimedia moving to Elasticsearch.
<https://blog.wikimedia.org/2014/01/06/wikimedia-moving-to-elasticsearch/>
(abgerufen am 25. 02. 2018).
- Jansen, H. (2014). Discovery-Services – Einführung, Marktübersicht und Trends. *Bibliotheksdienst*, 48(10), 773–783. doi:10.1515/bd-2014-0095
- Jürgens, P., Stark, B. & Magin, M. (2014). Gefangen in der Filter Bubble? In B. Stark, D. Dörr & S. Aufenanger (Hrsg.), *Die Googleisierung der Informationssuche* (S. 98–135). Berlin, Boston: de Gruyter.
- Krallinger, M. et al. (2017). Information retrieval and text mining technologies for chemistry. *Chemical Reviews*, 117(12), 7673–7761. doi:10.1021/acs.chemrev.6b00851
- Langenstein, A. & Maylein, L. (2009). Relevanz-Ranking im OPAC der Universitätsbibliothek Heidelberg. *B.I.T. Online*, 12(4), 408–413.
- Langville, A. N. & Meyer, C. D. (2006). *Google's PageRank and beyond: The science of search engine rankings*. Princeton: Princeton University Press.
- Li, H. (2011). A short introduction to learning to rank. *IEICI Trans. Inf. & Syst.*, 94(10), 1854–1862.
- LibRank. <http://www.librank.info/>
(abgerufen am 25. 02. 2018).
- Manning, C. D., Ragahvan, P. & Schütze, H. (2009). *An introduction to information retrieval*. Cambridge: Cambridge University Press.
- Maylein, L. & Langenstein, A. (2013). Neues vom Relevanz-Ranking im HEIDI-Katalog der Universitätsbibliothek Heidelberg. *B.I.T. Online*, 16(3), 190–200.
- Meadow, K. & Meadow, J. (2012). Search query quality and web-scale discovery: A qualitative and quantitative analysis. *College and Undergraduate Libraries*, 19(2–4), 163–175.
doi:10.1080/10691316.2012.693434
- MeetPrimo: Personalized ranking.
<http://meetprimo.com/features/schollarrank/>
(abgerufen am 25. 02. 2018).
- Merton, R. K. (1988). The matthew effect in science, II. *ISIS*, 79, 606–623.
- Natarajan, M. (2016). Exploring the E-discovery tools on the use of library collections by users. In E. De Smeg & S. Dhamdhare (Hrsg.), *E-Discovery tools and applications in modern libraries* (S. 122–137). Hershey: Information Science Reference. doi:10.4018/978-1-5225-0474-0.ch007
- Niedermaier, K. (2014). Gefährden Suchmaschinen und Discovery-Systeme die informationelle Autonomie? *Mitteilungen der VÖB*, 67(1), 109–125.
- Oberhauser, O. (2010). Relevance Ranking in den Online-Katalogen der “nächsten Generation.” *Mitteilungen der VÖB*, 63(1–2), 25–37.
- Pariser, E. (2011). *The filter bubble: How the new personalized web is changing what we read and how we think*. New York: Penguin.
- Plassmeier, K. et al. (2015). Evaluating popularity data for relevance ranking in library information systems. In *Proceedings of the Association for Information Science and Technology* (S. 1–4).
doi:10.1002/pr2.2015.1450520100125

- Plassmeier, K. (2016a). Role and relevance model.
http://www.librank.info/wp-content/uploads/2016/10/AP3_Relevanzmodell.pdf
(abgerufen am 25. 02. 2018).
- Plassmeier, K. (2016b). Übertragbarkeit auf EconBiz.
<http://www.librank.info/wp-content/uploads/2016/03/Übertragbarkeit-auf-EconBiz.pdf>
(abgerufen am 25. 02. 2018).
- ProQuest. (2013). Relevance Ranking in the Summon Service.
<http://media2.proquest.com/documents/Summon-RelevanceRanking-Datasheet.pdf>
(abgerufen am 25. 02. 2018).
- Quan-Haase, A. & McCay-Peet, L. (2014). The new boundaries of search. In B. Stark, D. Dörr & S. Aufenanger (Hrsg.), *Die Googleisierung der Informationssuche* (S. 136–159). Berlin, Boston: de Gruyter.
- Robertson, S. (2010). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4), 333–389.
- Roscher, M. (2014). *Fachdisziplinäre Bedürfnisse in der Gestaltung von Discovery-Lösungen: Wirklich ein Katalog für Alle? Berliner Handreichungen zur Bibliotheks- und Informationswissenschaft*. Humboldt-Universität Berlin.
- Smiley, D. & Pugh, E. (2009). *Solr 1.4 Enterprise Search Server*. Birmingham: Packt Publishing.
- Sunckel, B., Reh, U. & Nienerza, H. (2014). Das HeBIS Discovery System. *Bibliotheksdienst*, 48(10), 784–794.
- Thompson, J. (2014). *Implementing web-scale discovery services*. London: Rowman & Littlefield.
- Turnbull, D. & Berryman, J. (2016). *Relevant search*. Shelter Island: Manning.
- Wiesenmüller, H. (2012). Resource Discovery Systeme. *Jahrestagung der Gesellschaft für Klassifikation*.
urn:nbn:de:swb:90-290811
- Yewno: Yewno Discover.
<https://www.yewno.com/discover>
(abgerufen am 25. 02. 2018).