Andreas Henrich, Robin Jegan und Tobias Gradl

5.2 Data Retrieval

Abstract: Die Ermöglichung einer effektiven und effizienten Suche nach und in Forschungsdaten ist eine der wesentlichen Zielsetzungen des Forschungsdatenmanagements. Um Suchfunktionalitäten passend nutzen und bereitstellen zu können, sind verschiedene Aspekte des Data Retrieval relevant. Diese reichen vom Verständnis des einer Suche zugrundeliegenden Informationsbedarfs über Modelle zum inhaltsbasierten Ranking von Datensätzen bis hin zu Frameworks und Beispielsystemen für die Suche. Der vorliegende Beitrag gibt hierzu einen Überblick und eine Einführung.

1 Motivation

Für das Forschungsdatenmanagement (FDM) sind die bereits in vorangegangenen Beiträgen[1] dargelegten FAIR-Prinzipien von großer Bedeutung.[2] Ein wichtiger Teilaspekt unter dem Schlagwort „Findable" ist dabei „(Meta)data are registered or indexed in a searchable resource". Es geht hier also um die Bereitstellung von Suchfunktionalitäten, die es einer potenziellen Nutzerin bzw. einem potenziellen Nutzer ermöglicht, einen Forschungsdatensatz zu finden, von dem sie oder er zuvor in der Regel noch keine explizite Kenntnis hatte. Die Suche ist hier abzugrenzen vom direkten Zugriff auf einen Datensatz, der über eindeutige Identifikatoren wie einen Digital Object Identifier (DOI) erfolgen kann.

Von der Begrifflichkeit her ist festzuhalten, dass sich Data Retrieval grundsätzlich auf die Suche in (semi-)strukturierten Daten, und in diesem Kontext auf die Suche nach Forschungsdaten bezieht. Ein erster und primärer Zugang ist die Suche über entsprechend gepflegte Metadaten zu den einzelnen Forschungsdaten. Neben sehr allgemeinen Metadatenschemata existieren fachspezifische Schemata, die spezifische Charakteristika von Forschungsobjekten für das jeweilige Fach einbeziehen. In Abhängigkeit von dem konkret verwendeten Schema umfassen Metadaten typischerweise technische Daten wie das Format, inhaltsbeschreibende Daten wie den Titel oder eine Kurzbeschreibung, Informationen zu Zugriffsrechten, Identifikatoren, das Jahr und den Ort der Publikation und vieles mehr. Auch Schlagwörter und ggf. Referenzen auf Publikationen sind üblich.

1 S. z. B. Beitrag von Linne et al., Kap. 3.2 in diesem Praxishandbuch.
2 Vgl. Wilkinson 2016.

https://doi.org/10.1515/9783110657807-024

Eine Suche auf Basis der Metadaten kann sehr unterschiedlich erfolgen. Eine Stichwortsuche im Titel und der Beschreibung bildet einen üblichen Weg. Dies kann durch Filterbedingungen z. B. zum Jahr der Publikation oder zu anderen Feldern ergänzt werden. Während die inhaltsbasierte Suche in beschreibenden Texten neben klassischen Verfahren des booleschen Retrieval auch Techniken aus dem Kontext von (Web-)Suchmaschinen nutzen kann, können einfache Attribute, wie das Jahr der Publikation, als Filterbedingungen zur Eingrenzung der Ergebnisse dienen.

Neben einer Suche über die Metadaten ist bei Forschungsdaten, die z. B. umfangreiche Text- oder Bildteile umfassen, auch eine Suche auf den Inhalten selbst denkbar. Dazu können Verfahren der Textsuche, wie sie im vierten Kapitel dieses Beitrags beschrieben werden, eingesetzt werden, um so die Möglichkeiten des Data Retrieval zu ergänzen. Für andere Medientypen wie etwa Bilder, Videos oder Audiodateien ist man auf entsprechende spezialisierte Verfahren angewiesen.[3] In diesem Beitrag wird primär auf die Suche über Metadaten eingegangen.

Der Beitrag gliedert sich wie folgt: Zunächst werden wir zum besseren Verständnis der Suche unterschiedliche Arten von Suchsituationen darlegen und die Suche als oft iterativen Prozess beschreiben. Im Anschluss wird eine tiefere Betrachtung zum Charakter der Metadaten in Relation zu den Forschungsdaten selbst gegeben. Da zur Implementierung von Suchlösungen Techniken aus dem Information Retrieval angewendet werden, folgt eine Betrachtung exemplarischer Modelle des Information Retrieval, wobei wir auch die Besonderheiten bei der Verarbeitung anderer Medientypen skizzieren. Darauf aufbauend betrachten wir die Umsetzung von Suchsystemen, mögliche Architekturen für Suchlösungen und Beispiele für zugrundeliegende Softwaresysteme sowie exemplarische Systemumsetzungen. Eine Zusammenfassung rundet den Beitrag ab.

2 Arten von Suchsituationen und der Suchprozess

Sowohl die Suche nach als auch die Nutzung von Informationen sind Teile eines Prozesses, der insbesondere in der digitalen Welt von fundamentaler Bedeutung ist. Jedoch unterscheiden sich Situationen und Kontexte der Suche auf mehreren Ebenen.

Eine Unterscheidung kann im Typ der Anfrage oder in der Art des Informationsbedarfs gesehen werden.[4] So kann ein *konkreter Informationsbedarf* vorliegen, in dem nach Fakten bzw. mit klar abgesteckten thematischen Grenzen gesucht wird,

3 Vgl. die Ausführungen in Abschnitt 4.5 in diesem Beitrag sowie z. B. Raieli 2016, 9–42 oder Ponceleón 2012, 587–639.

4 Vgl. Frants 1997, 34–40.

beispielsweise nach der Einwohnerzahl einer bestimmten Stadt. Diese Anfrage kann gewöhnlich mit einer einzelnen Fakteninformation beantwortet werden und ebenso kann mit präzisen Anfragen danach gesucht werden. Weiterhin ist dieser konkrete Informationsbedarf nach der Übermittlung der erforderlichen Fakten befriedigt.

Ein zweiter Typ kann als *problemorientierter Informationsbedarf* charakterisiert werden. Hier kann die Suchanfrage nicht durch eine einzelne Fakteninformation vollständig beantwortet werden. Stattdessen ist meist die Analyse mehrerer durch die Suche zurückgegebener Datensätze notwendig, um die Anfrage beantworten zu können. Außerdem werden die thematischen Grenzen hier nicht klar gesetzt, weswegen auch die Suchanfrage durch mehrere und unterschiedliche Suchterme ausgedrückt werden kann. Die Rückgabe der Suchergebnisse und deren Betrachtung durch die Nutzerin bzw. den Nutzer kann zu einer Modifizierung der Suchanfrage führen, falls der Informationsbedarf noch nicht gedeckt wurde.

Eine weitere Differenzierung hinsichtlich der Suche wurde von Marchionini durch die Aufteilung in *Lookup*, *Learn* und *Investigate* getroffen.[5] Die Suche im Sinne eines Lookup ist am stärksten mit dem oben beschriebenen konkreten Informationsbedarf zu vergleichen. Hier steht die Suche nach Fakten im Vordergrund, die mithilfe eines einzelnen konkreten Suchergebnisses für die Anfrage abgeschlossen werden kann. Interessanter ist die Abgrenzung zu Learn und Investigate, die Eigenschaften des problemorientierten Informationsbedarfs aufweisen, und deren Unterschiede. Die Suche im Sinne von Learn umfasst mehrere Suchanfragen, welche verglichen mit Lookup ausführlichere Antworten liefen und außerdem meist eine Interpretation oder weitere Analyse erfordern. Das Ziel dieser Suche ist ein tieferes Verständnis zu der gewünschten Anfrage, welche durch das Verarbeiten und Vergleichen der Suchergebnisse herausgearbeitet werden kann. Investigate umfasst ebenso wie Learn die Analyse von mehreren Suchanfragen, auf die jedoch eine noch ausführlichere Interpretation und Evaluation folgt. Diese tiefe Analyse erfordert allerdings bereits existierendes Fachwissen, um Vergleiche und Bewertungen für die Suchergebnisse durchführen zu können.

Sowohl Learn als auch Investigate können in eine gemeinsame Kategorie der Exploratory Search eingegliedert werden.[6] Durch die umfangreiche Beteiligung der Nutzerin bzw. des Nutzers, durch zahlreiche Anfragen sowie durch die nötige manuelle Evaluierung der Ergebnisse liegt der Fokus in beiden Sucharten auf einer länger andauernden Untersuchung der Suchergebnisse. Natürlich kann intuitiv argumentiert werden, dass Suchen im Zusammenhang mit Forschungsdaten primär der Kategorie der Exploratory Search zuzuordnen sind, weil es um die Suche nach potenziell relevanten Forschungsdaten für eine bestimmte Forschungsfrage geht. Es ist aber in Analogie zu fast allen Suchlösungen davon auszugehen, dass in vielen Fällen die

5 Vgl. Marchionini 2006, 42–43.
6 Vgl. Marchionini 2006, 43–44.

Suchenden bereits recht genau wissen, wonach sie suchen. Die Suchform Lookup sollte daher bei der Konzeption einer Suchlösung für Forschungsdaten mit bedacht werden, weil die Nutzerin bzw. der Nutzer hier in der Regel davon ausgeht, dass lediglich eine Suchanfrage gestellt werden muss, die unmittelbar zur Rückgabe der relevanten Information führt. Während bei einer explorativen Suche eine iterative Verfeinerung der Anfrage akzeptiert wird, ist dies beim Lookup keineswegs der Fall.

Insbesondere der explorative Suchprozess besteht aus einer Reihe von Aktivitäten, die iterativ ausgeführt werden und aufeinander aufbauen.[7] Zu Beginn der Suche muss ein Bedarf nach einer bestimmten Information erkannt werden (recognize), woraufhin der zweite Schritt erfolgt, nämlich, dass akzeptiert wird, dem Bedarf nach Information nachzugehen (accept). Anschließend wird das Problem formuliert (formulate). Es wird identifiziert, welche Information den Bedarf decken kann und welche Quellen herangezogen werden können. Mithilfe der Suchanfrage soll daraufhin ausgedrückt werden, wie die Suchlösung den Informationsbedarf decken soll (express). Durch die Suchlösung gelieferte Ergebnisse werden im Anschluss geprüft (examine), möglicherweise auch mehrmals. Da in diesem Prozess nicht immer sofort die passenden Treffer gefunden werden können, folgt oft eine Umformulierung der Anfrage (reformulate). Das positive Ende des Suchprozesses ist erreicht, sobald die Nutzerin bzw. der Nutzer die Suche beendet und die erhaltenen Informationen verwendet (use).

Der Aufwand, der von Seiten der Nutzerin bzw. des Nutzers einerseits und der Suchlösung andererseits aufgewendet werden muss, ist je nach Aktivität im Suchprozess unterschiedlich verteilt. So liegt ein großer Teil des Aufwandes zur Formulierung des Problems und zur Prüfung der Ergebnisse bei der Nutzerin bzw. dem Nutzer. Im Gegensatz dazu kann die Suchlösung insbesondere bei den Aktivitäten Hilfe bieten, welche sich mit dem Formulieren der Suchanfrage und deren Verfeinerung beschäftigen. Für das Formulieren der Suchanfrage sind Hilfestellungen wie die Autovervollständigung, eine Rechtschreibprüfung oder vorgeschlagene Suchbegriffe zu nennen und für die Umformulierung etwa die Ergänzung vorheriger Suchanfragen, um die Rangliste der Ergebnisse zu verbessern.

Anfragen können aber nicht nur in textueller Form gestellt werden, sondern auch durch andere, an einen bestimmten Anwendungszweck angepasste Suchparameter. Ein Beispiel ist die Suche über chemische Strukturen, etwa in der Crystallography Open Database (COD)[8], in der mittels zweidimensionaler Skizzen Kristallstrukturen aus wissenschaftlichen Veröffentlichungen durchsucht werden können. Dazu kann man in einem kleinen, speziellen Editor Strukturen oder Fragmente von Strukturen als Anfrage skizzieren. Die Möglichkeiten der COD sind damit ein Beispiel für die Anfrageformen Query by Sketch oder Query by Example.

7 Vgl. Marchionini 2007, 207–228.

8 S. http://www.crystallography.net/cod/jsme_search.html. Letztes Abrufdatum der Internet-Dokumente ist der 15.11.2020.

In der Darstellung der Ergebnisse kann eine Suchlösung durch passende Ergebnisbeschreibungen eine erste Hilfestellung zur unmittelbaren Relevanzbeurteilung geben. Eine Suchlösung muss dabei der Heterogenität der Informationsbedürfnisse ebenso Rechnung tragen wie dem oft iterativen Suchprozess, in dem die Nutzerin bzw. der Nutzer entsprechende Unterstützung und Orientierung erwartet. Weiterhin muss bei Daten, die auf unterschiedlichen Ebenen organisiert sind – etwa Forschungsdaten die nicht nur als einzelner Datensatz vorliegen, sondern auch als Teil von Sammlungen – der variable Aggregationsgrad beachtet und in den Suchergebnissen entsprechend präsentiert werden.

3 Metadaten vs. Forschungsdaten

Data Retrieval für Forschungsdaten setzt einen entsprechenden Bestand an Forschungsdatensätzen voraus. Die Suche in diesem Bestand soll als Ergebnis eine Menge oder ein Ranking relevanter Datensätze für die gegebene Anfrage erzeugen. Jedoch kann hier zwischen verschiedenen Typen von Daten unterschieden werden, die für die Suche ausgewertet werden können, nämlich den Metadaten und den Forschungsdaten selbst.

Metadaten sind Daten über Daten, also zusätzlich zu den eigentlichen Forschungsdaten gespeicherte Informationen, die etwa den Titel, eine Beschreibung, den Verfasser oder das Erstelldatum des Dokuments umfassen.[9] Dabei können verschiedene Arten von Metadaten unterschieden werden:[10] *Deskriptive Metadaten* beschreiben inhaltliche Felder, wie etwa die Themen, welche in einem Dokument vorkommen, oder formale Aspekte, wie die Anzahl der Wörter. *Strukturelle Metadaten* umfassen die Aufteilung eines Dokuments bzw. Forschungsdatenbestandes, das heißt in welchen Einheiten das Dokument aufgebaut ist, etwa Kapitel und Unterkapitel oder Teilmengen. *Administrative Metadaten* beschreiben organisationsbezogene Gegebenheiten wie die Lizenzen, welche die Forschungsdaten betreffen, oder auch die zugehörige Institution, die bei der Entstehung der Forschungsdaten mitgewirkt hat. Zuletzt präsentieren *technische Metadaten* beispielsweise Details zum Dateiformat und der Dateigröße.

Die Definition und Beschreibung von Metadaten erfolgt insbesondere in kleineren Projekten und Sammlungen initial häufig nicht anhand von Standards. Stattdessen werden oft eigene Schemata entwickelt, welche die Anforderungen der vorliegenden Domäne möglichst exakt widerspiegeln sollen. Aspekte der FAIR-Prinzipien – wie die langfristige Nachnutzbarkeit von Metadaten aus vielen Systemen –

9 Vgl. Ferber 2003, 267–284.
10 Vgl. Schöch 2017, 228–229.

werden allerdings erst durch die Verwendung von Standards ermöglicht, da diese eine übergreifende Interpretierbarkeit der Datenfelder gestatten. Die Dublin Core Metadata Initiative hat mit den fünfzehn Hauptelementen von Dublin Core einen frühen Metadatenstandard geschaffen,[11] welcher bis heute im Gebrauch ist. Eine Weiterentwicklung von Dublin Core – ebenso wie des im Bibliothekswesen gebräuchlichen MARC-Standards[12] – wurde mit dem Metadata Object Description Schema (MODS) geschaffen,[13] um zum einen Kompatibilität zu MARC zu gewährleisten und zum anderen den beschränkten Dublin Core Standard zu erweitern.

Für das kulturelle Erbe im Allgemeinen ist das CIDOC Conceptual Reference Model (CRM) relevant,[14] welches eine semantische Datenmodellierung ermöglicht, die dadurch die Erstellung von Ontologien gestattet. Die Auswahl verfügbarer Standards ist dabei annähernd so divers, wie das disziplinäre Spektrum der Wissenschaften selbst. Für einen Überblick über fachspezifische Standards und weiterführende Betrachtungen kann an dieser Stelle lediglich auf einschlägige Literatur verwiesen werden – für den Bereich des kulturellen Erbes beispielsweise auf Neuroth und Flanders.[15]

Der Begriff der digitalen Daten umfasst in diesem Kontext neben den Metadaten auch die inhaltliche Ebene, also digitalisierte bzw. digital erstellte (born-digital) Artefakte. Forschungsdaten sind als Begriff für die Gesamtheit an Daten aufzufassen, welche einen Datensatz ausmachen, das heißt Inhalte ebenso wie andere Formate von Daten und zugehörige Metadaten.[16] Ein Beispiel für ein in dieser Hinsicht übergreifendes Format wird von der Text Encoding Initiative (TEI)[17] betreut und weiterentwickelt. Das gleichnamige Dokumentenformat ist ein Standard für Textdaten, -kodierung und -transfer und hat sich in den Geisteswissenschaften (u. a. Editionswissenschaft, Linguistik) etabliert. Neben den encodierten, annotierten Inhalten bietet das TEI Format tiefe Möglichkeiten zur Beschreibung von Metadaten.

Standardisierte Formate nicht nur von Metadaten, sondern auch von Forschungsdaten allgemein und den Schnittstellen, über die diese abgerufen werden können, erlauben den Austausch durch Institutionen und Wissenschaftlerinnen bzw. Wissenschaftler. An dieser Stelle bleibt festzuhalten, dass Suchlösungen für Forschungsdaten verschiedene Standards zu Metadaten unterstützen und nach Möglichkeit auch Werkzeuge zur Integration bereitstellen sollten.

11 S. https://www.dublincore.org/specifications/dublin-core/dces.

12 S. https://www.loc.gov/marc.

13 S. http://www.loc.gov/standards/mods.

14 S. http://www.cidoc-crm.org Letztes Abrufdatum der Internet-Dokumente ist der 15.11.2020.

15 Vgl. Neuroth 2017, 213–22; Flanders 2015, 229–237.

16 Forschungsdaten umfassen dabei zum Teil auch Referenzen auf sogenannte „Sekundärdaten", die aus der Verarbeitung der Primärdaten etwa durch Interpretation oder Datenaggregation entstehen können. Vgl. Rixen 2018.

17 S. https://tei-c.org.

4 Modelle des Information Retrieval

Die Aufgabe eines Suchsystems für Forschungsdaten ist es, primär zu einem Informationsbedarf relevante Datensätze im Ergebnis für eine Suche zu präsentieren. Klassisch wurde dabei häufig eine Ergebnismenge berechnet. Dies ist allerdings in den letzten Jahrzehnten durch Rankings von relevanten Datensätzen bzw. Dokumenten abgelöst worden, weil der Nutzerin bzw. dem Nutzer so ein differenzierteres Bild präsentiert werden kann. In diesem Kapitel soll es nun um die Modelle gehen, die der Bestimmung einer Ergebnismenge bzw. eines Rankings zugrunde liegen. Im Forschungsgebiet des Information Retrieval (IR) ging und geht es unter anderem darum, (mathematische) Modelle für die Ermittlung relevanter Dokumente zu einer Anfrage zu definieren.[18] Einige exemplarische Modelle werden wir im Weiteren beschreiben, um dann in späteren Abschnitten auf die technische Umsetzung solcher Modelle einzugehen. Die Modelle des IR gehen überwiegend von Situationen aus, in denen mit einer entsprechenden Anfrage in einer Kollektion von Dokumenten – bzw. Datensätzen – gesucht wird, die in der Regel durch einen Text repräsentiert werden.[19] In unserem Fall könnte es sich bei diesem Text z. B. um den Titel eines Forschungsdatensatzes oder einen kurzen Beschreibungstext aus den Metadaten zu diesem Datensatz handeln.

Um die Modelle besser verstehen zu können, ist es wichtig, sich nochmals klarzumachen, dass die Aufgabe eines Suchsystems in der Bereitstellung relevanter Forschungsdaten bzw. Informationen liegt. Die Qualität des Ergebnisses hat hier zwei Perspektiven: Zum einen sollte das Ergebnis möglichst viele relevante Forschungsdaten zum Informationsbedarf enthalten. Auf der anderen Seite ist es aber auch wichtig, dass das Ergebnis möglichst wenige irrelevante Datensätze enthält. Zur Einschätzung des ersten Aspekts verwendet man den Recall als Kennzahl. Dieser errechnet sich aus der Anzahl relevanter Datensätze im Ergebnis der Suchmaschine im Verhältnis zur Anzahl der insgesamt in der Kollektion enthaltenen relevanten Datensätze. Der Recall misst damit die Vollständigkeit des Ergebnisses. Dem steht als zweite Kennzahl die Precision gegenüber. Sie misst, wie gut es dem System gelingt, nicht relevante Datensätze aus dem Ergebnis fernzuhalten. Die Precision errechnet sich aus der Anzahl relevanter Datensätze im Ergebnis im Verhältnis zur Gesamtzahl der Datensätze im Ergebnis (relevante und irrelevante).

Ziel eines Suchsystems muss es nun sein, einen geeigneten Kompromiss zwischen diesen Zielgrößen zu erzielen. Um zu messen, wie gut dieser Kompromiss ge-

18 Vgl. Croft 2010, 1–12.

19 Bei Bilddaten – als Beispiel für multimediale Daten – kann man einerseits versuchen durch eine (ggf. automatische) Analyse des Bildinhalts eine Verschlagwortung oder Klassifikation durchzuführen und so den Inhalt des Bildes ebenfalls durch Text zu repräsentieren. Andererseits kann man aber auch Vergleiche auf den Bilddaten selbst durchführen. Vgl. die Ausführungen in Abschnitt 4.5 in diesem Beitrag sowie z. B. Bullin 2020, 1–22.

lingt, wird bisweilen das sogenannte F-Maß eingesetzt, das dem harmonischen Mittel aus Recall (R) und Precision (P) entspricht: $\frac{2RP}{(R+P)}$. Durch die Verwendung des harmonischen Mittelwertes statt des arithmetischen Mittelwertes wird erreicht, dass das F-Maß niedrig ist, sobald eine der beiden Eingangsgrößen niedrig ist.

Die skizzierten Maße können zum einen genutzt werden, um die Leistungsfähigkeit entsprechender IR-Modelle einzuschätzen. Zum anderen helfen sie auch bei der Charakterisierung von Anfragen. So gibt es Anfragen, für die ein hoher Recall wichtig ist – z. B. bei der Recherche nach Forschungsarbeiten in einem Promotionsvorhaben. Bei anderen Anfragen spielt der Recall eine geringere Rolle, weil man zur Beantwortung der Anfrage nur ein oder zwei relevante Datensätze benötigt – wenn man z. B. nach dem Geburtsdatum einer Person sucht.

4.1 Boolesches Retrieval

Ein erstes einfaches Retrievalmodell findet sich im klassischen booleschen Retrieval. Hier können zunächst Texte gesucht werden, die einzelne Anfragebegriffe enthalten. Durch die Zusammensetzung von einzelnen Begriffen oder Anfrageteilen mithilfe boolescher Operatoren können Anfragen weiter ausspezifiziert werden. So sucht eine Anfrage „Novelle AND Mittelalter“ z. B. nach Datensätzen, die beide Begriffe enthalten. Neben den üblichen booleschen Operatoren können zum Teil auch komplexere Operatoren wie NEAR[n] angewendet werden, wobei in diesem Fall die beiden Begriffe links und rechts des Operators in einem Wortfenster von maximal n Wörtern gemeinsam vorkommen müssen. Vorteile des booleschen Retrieval sind, dass die Ergebnisse vorhersehbar und relativ einfach zu erklären sind. Viele verschiedene Eigenschaften (wie z. B. auch das Publikationsdatum) können in einen Anfrageausdruck einbezogen werden. Nachteilig ist allerdings, dass die Effektivität davon abhängt, ob es der Nutzerin bzw. dem Nutzer gelingt, einen passenden booleschen Ausdruck zu formulieren. Hinzu kommt, dass das Modell eine nicht weiter strukturierte Ergebnismenge liefert, was insbesondere bei großen Ergebnismengen unmittelbar den Bedarf zur Verfeinerung der Anfrage nach sich zieht.

4.2 Vektorraummodell

Ein weiteres weit verbreitetes Retrievalmodell ist das Vektorraummodell. Die Anfragen werden hier, verglichen mit dem booleschen Retrieval, nicht mithilfe von Operatoren erstellt, sondern können als Schlüsselwortanfragen gestellt werden. Sowohl die Forschungsdaten als auch die Anfragen werden als Vektoren dargestellt, wobei die Werte in den einzelnen Dimensionen der Vektoren die Bedeutung einzelner Wörter (Terme) für die jeweiligen Forschungsdaten bzw. die Anfrage repräsentieren. Die Anzahl der Dimensionen der Vektoren entspricht damit der Größe des Vokabu-

lars, das alle Begriffe enthält, die in den betrachteten Dokumenten vorkommen. In dem so gebildeten Vektorraum kann mit einem entsprechenden Ähnlichkeitsmaß nach den zu einem Anfragevektor ähnlichsten Dokumentvektoren gesucht werden. Das Maß der Kosinus-Ähnlichkeit etwa drückt über den Kosinus des Winkels zwischen den Vektoren die Ähnlichkeit zwischen Anfrage und Dokument aus und ermöglicht dadurch ein Ranking der Forschungsdaten basierend auf der Anfrage.

Für die Verbesserung der Ergebnisse können in diesen Modellen unterschiedliche Ähnlichkeitsmaße und Verfahren zur Bestimmung der Termgewichtungen eingesetzt werden.[20] Während bei den Ähnlichkeitsmaßen häufig die oben erwähnte Kosinus-Ähnlichkeit genutzt wird sind für die Termgewichtungen verschiedene Formeln nach dem sogenannten *tf-idf*-Muster im Einsatz. *tf* steht dabei für die Frequenz eines Terms in einem Dokument. Dem liegt die Annahme zugrunde, dass ein Term umso besser geeignet ist ein bestimmtes Dokument zu beschreiben, je häufiger er in diesem Dokument vorkommt. Allerdings ist auch zu berücksichtigen, dass das häufige Vorkommen eines Terms in einem Dokument nur dann bedeutsam für dieses Dokument selbst ist, wenn der Term nicht auch in anderen Dokumenten relativ häufig vorkommt. Dieser Tatsache wird durch die *idf*-Komponente Rechnung getragen, die als inverse Dokumentfrequenz (*idf*) umso höher ist, je seltener der Begriff im gesamten Korpus vorkommt. Durch *tf-idf*-Formeln werden also Terme hoch gewichtet, die im betreffenden Dokument häufig, im gesamten Korpus dagegen eher selten sind. Die für Indexierung und Suchanwendungen bekannte Bibliothek Lucene ermöglicht sowohl das oben erwähnte boolesche Retrieval als auch das Vektorraummodell für das Bewerten von Dokumenten.[21] Beginnend mit der Version 6.0 von Lucene wurde 2016 aber das bisher als Standardscoring verwendete *tf-idf*-Modell durch ein anderes Modell ersetzt, nämlich BM25, welches im folgenden Kapitel näher beleuchtet wird.[22]

4.3 Probabilistisch motivierte Modelle

Während das Vektorraummodell weitgehend pragmatisch motiviert ist, gab es im IR immer auch die Bestrebung, probabilistisch fundierte Modelle für das Ranking von Dokumenten zu entwickeln. Ein wichtiger Meilenstein in diesem Zusammenhang war das BIR-Modell.[23] BIR steht für Binary Independence Retrieval und macht verschiedene Annahmen deutlich. Eine erste wesentliche Annahme ist, dass das Vor-

20 Vgl. Manning 2008, 289–292.

21 S. https://lucene.apache.org/core/8_4_1/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html.

22 S. https://lucene.apache.org/core/6_0_0/changes/Changes.html.

23 Vgl. Robertson 1976, 129–146.

kommen eines Wortes in einem Dokument eine binäre, nicht weiter gewichtete Eigenschaft darstellt. Das bedeutet, es geht nur darum, ob ein Wort in einem Dokument vorkommt, und nicht z. B. darum, wie oft es in diesem Dokument vorkommt. Ein zweiter wichtiger Punkt ist die Unabhängigkeitsannahme (Independence). Diese Annahme liegt dabei auch dem Vektorraummodell zugrunde, in dem die einzelnen Dimensionen des Vektorraumes repräsentieren, wie gut ein bestimmtes Wort geeignet ist, um ein Dokument – bzw. einen Datensatz – zu beschreiben. Im Hinblick auf ein probabilistisches Modell erlaubt die Unabhängigkeitsannahme die Wahrscheinlichkeiten für das Vorkommen einzelner Terme/Wörter in Dokumenten unabhängig voneinander zu betrachten und zu verrechnen. Die Unabhängigkeitsannahme ist dabei natürlich stark vereinfachend, denn einzelne Begriffe wie z. B. „Dichter" und „Poet" werden keineswegs statistisch unabhängig in Dokumenten vorkommen. Die Unabhängigkeitsannahme ist aber wichtig, um effiziente mathematische und auch algorithmische Verfahren für das Ranking einsetzen zu können.

Während das reine BIR-Modell in der praktischen Anwendung keine nennenswerte Rolle spielt, hat es doch viele weitergehende Verfahren beeinflusst und Eingang in das sehr oft genutzte Modell Okapi BM25 gefunden.[24] Im Folgenden werden wir dieses Modell in Anlehnung an Croft etwas genauer betrachten, da es den Charakter von IR-Modellen exemplarisch verdeutlicht:[25]

$$Score(D,Q) = \sum_{(i \in Q)} log \frac{(N - n_i + 0,5)}{(n_i + 0,5)} \cdot \frac{(k_1 + 1) \cdot f_i(D)}{k_1 \left((1-b) + b \cdot \frac{dl(D)}{avdl} \right) + f_i(D)} \cdot \frac{(k_2 + 1) \cdot f_i(Q)}{k_2 + f_i(Q)}$$

Die Formel berechnet mit *Score(D,Q)* ein Maß für die Passung des Dokumentes *D* zur Anfrage *Q*. Die Anfrage *Q* besteht dabei aus einer Reihe von Anfragebegriffen $i \in Q$. In der Summe der obigen Formel werden Werte für die einzelnen Anfragebegriffe addiert. Dabei wird aus technischen Gründen der Logarithmus verwendet. Durch die Eigenschaften des Logarithmus ist gewährleistet, dass sich die Rangordnung der Dokumente auf Basis der Score-Werte durch die Anwendung des Logarithmus nicht ändert. Das folgende Produkt besteht aus drei Faktoren. Der erste Faktor bestimmt auf Basis der Kennzahlen *N* (= Anzahl der Dokumente in der Kollektion) und n_i (= Anzahl der Dokumente, in denen der Begriff *i* vorkommt) eine aus dem BIR-Modell stammende Variante der *idf*-Komponente. Hier wird ausgedrückt, wie selten der Begriff in der Kollektion ist. Je seltener der Begriff, umso höher ist seine „Erklärungskraft" für Dokumente, in denen er vorkommt.

24 Vgl. Sparck 2000, 795–802.
25 Vgl. Croft 2010, 243–252.

Für die folgenden beiden Faktoren werden Parameter k_1, k_2 und b berücksichtigt, die für bestimmte Anwendungen optimiert werden können. Als Standardwerte[26] werden z. B. $k_1 = 1,2$, $k_2 \in [0;1000]$ und $b = 0,75$ genutzt.

Im mittleren Faktor wird die Vorkommenshäufigkeit $f_i(D)$ des Begriffs i im Dokument D berücksichtigt (*tf*-Komponente). Im Nenner wird dabei mit der Relation zwischen der Länge $dl(D)$ des Dokumentes D und der durchschnittlichen Länge eines Dokumentes in der Kollektion *avdl* und den gegebenen Parametern gerechnet, um zielgerichtet die Vor- und Nachteile kürzerer und längerer Dokumente durch eine angepasste Dokumentlängennormierung zu berücksichtigen. Dies ist wichtig, weil in kurzen Dokumenten einzelne Begriffe fast zwangsweise relativ häufig sind. Würde man also mit relativen Häufigkeiten arbeiten, würden kurze Dokumente im Ranking stark bevorteilt, während bei absoluten Häufigkeiten lange Dokumente im Ranking bevorzugt würden. Die angepasste Dokumentlängennormierung schafft hier einen Ausgleich.

Der letzte Faktor spielt bei Stichwortanfragen keine Rolle, da dort die Vorkommenshäufigkeit der Stichworte in der Anfrage $f_i(Q)$ in der Regel jeweils 1 sein wird. Verwendet man die Formel jedoch um Anfragetypen wie „Suche ähnliche Dokumente“ zu unterstützen, sollte die Vorkommenshäufigkeit im Anfragedokument berücksichtigt werden. Auch hier kann die Auswirkung wieder über einen Parameter (k_2) gesteuert werden.

Die betrachtete Formel verdeutlicht den typischen Aufbau von Ranking-Funktionen, die dazu dienen, die Dokumente anhand der errechneten Werte im Ergebnis zu einer Anfrage zu sortieren.

4.4 Sprachmodelle

Neben dem Vektorraummodell oder BM25 existieren weitere für Suchsituationen, Dokumentvergleiche und andere Aufgabenfelder vielversprechende Ansätze. Einer dieser Ansätze basiert auf statistischen Sprachmodellen. Einzelne Dokumente werden hier mithilfe der Vorkommenswahrscheinlichkeiten der in ihnen enthaltenen Wörter charakterisiert (relative Vorkömmenshäufigkeiten). Mathematisch gesehen liegt dabei eine Multinomialverteilung über Wörtern vor (Urnenmodell mit Zurücklegen). Diese Sprachmodelle können entweder genutzt werden, um z. B. zu ermitteln, wie wahrscheinlich die Generierung einer bestimmten Anfrage auf Basis des Sprachmodells eines Dokumentes wäre. Das Dokument mit der höchsten Wahrscheinlichkeit, eine bestimmte Anfrage zu generieren, wird dann als relevantestes Dokument für diese Anfrage eingestuft. Auf der anderen Seite erlauben diese Modelle aber auch, Dokumente miteinander zu vergleichen und ähnliche Dokumente auf

26 Vgl. Robertson 1999, 3.

Basis der Sprachmodelle zu bestimmen. Für den erfolgreichen Einsatz von Sprachmodellen spielen dabei Glättungstechniken eine wichtige Rolle, bei denen die aus den Wortvorkommen im Dokument gewonnenen Wahrscheinlichkeiten mit entsprechenden Hintergrundwahrscheinlichkeiten geglättet werden.[27] Dadurch wird insbesondere das Problem behoben, dass Begriffe, die nicht in einem Dokument vorkommen, eine Erzeugungswahrscheinlichkeit von 0 als Anfragebegriffe erhalten würden. Sprachmodelle werden beispielsweise in der Websuche verwendet, um die Relevanz der Elemente auf der Suchergebnisseite zu verbessern.[28]

4.5 Multimedia Information Retrieval

Multimediale Daten erfordern auf den Datentyp angepasste Verarbeitungsschritte, um Retrieval ähnlich zu den vorgestellten Modellen für Text zu ermöglichen. Nicht nur Bilder, sondern auch weitere Medien wie Video und Audio sind im Bereich der Forschungsdaten von großer Bedeutung und müssen entsprechend behandelt werden. Exemplarisch soll hier auf einige Besonderheiten bei der Analyse von Bildern eingegangen werden.

Metadaten spielen bei der Bildsuche – wie auch im Text Retrieval – eine große Rolle, da je nach Umfang und Qualität der Metadaten die Anfragen ohne Rückgriff auf die eigentlichen Bildinhalte beantwortet werden können. Auch hier sind Standards wie Dublin Core oder CIDOC CRM weit verbreitet. Falls die Anfragen jedoch mittels Metadaten nicht ausreichend beantwortet werden können, muss eine inhaltsbasierte Analyse hinzugezogen werden. Das sogenannte „Content Based Image Retrieval“ bezieht hierzu Farb- oder Helligkeitswerte der Pixel in die Analyse ein, wobei wie im Text Retrieval zahlreiche Modelle und Techniken verfügbar sind.[29]

Die klassische Bildanalyse analysiert Eigenschaften des gesamten Bildes, insbesondere Farben, Texturen und Formen, und kann somit bei spezifischen Suchanfragen, etwa zur Suche nach Ähnlichkeiten zwischen Markenzeichen und Firmenlogos erfolgreich eingesetzt werden.[30] Sollen jedoch nur Teile eines Bildes betrachtet werden, etwa um andere Bilder zu finden, in denen das Anfragebild als Teilbild auftaucht, werden die Schwächen der klassischen Bildanalyse deutlich. In Anwendungen, in denen nicht mithilfe von Stichwörtern, sondern mit einem Anfragebild gesucht werden soll – auch als „Query by Example“ bezeichnet – wird daher oft eine Segmentierung vorgenommen. So können lokale, charakteristische Stellen im Bild identifiziert werden. Mittels dieser Bildregionen sollen in der Suche auch Bilder ge-

27 Vgl. Zhai 2004, 183–185.
28 Vgl. Ogilvie 2003, 143–150.
29 Vgl. Ponceleon 2011, 592–597.
30 Vgl. Bullin 2020, 8–10.

funden werden, die nur in einem kleinen Teilbereich übereinstimmen bzw. diesem ähneln.[31] Der Gedanke der Segmentierung wird dabei von Verfahren fortgeführt, die markante Punkte im Bild identifizieren und deren lokale Umgebung durch ihre Eigenschaften (z. B. die Orientierung von Kanten oder Farbverläufen) repräsentieren.[32] Für große Datenbestände wurden hierzu effiziente Methoden eingeführt. „Bag of Visual Words" etwa stellt einen Vektorraum auf, in Analogie zum im Text Retrieval lange eingesetzten „Bag of Words"-Modell, um darin charakteristische „Visual Words" abzubilden, die das Bild beschreiben. Diese Visual Words können z. B. durch Clustern der markanten Bildpunkte aus einer Beispielkollektion auf Basis ihrer Beschreibungsvektoren gewonnen werden.

In neuerer Zeit haben in der Bildsuche auf neuronalen Netzen basierende Ansätze große Fortschritte erzielt. Dies gilt insbesondere für die Klassifikation von Bildinhalten und damit für die (semi-)automatische Annotation von Bilddaten. Hierzu werden in der Regel sogenannte Deep Convolutional Neural Networks genutzt.[33]

4.6 Weitere Einflussfaktoren zum Ranking

Während die bisher betrachteten Modelle ihren Fokus auf der inhaltlichen Passung von Dokumenten haben, sind für die Relevanz von Dokumenten im Hinblick auf eine konkrete Anfrage häufig auch noch andere Kriterien ausschlaggebend. Man denke hier z. B. an das Veröffentlichungsjahr, an die veröffentlichende Institution oder gegebenenfalls auch an die Popularität einzelner Dokumente bzw. Datensätze. Im Bereich der Websuchmaschinen hat hier z. B. der PageRank-Algorithmus[34] große Bedeutung gewonnen. Das Ranking von Dokumenten wird daher bei vielen Suchmaschinen nicht allein auf Basis der inhaltlichen Passung zu einer Anfrage ermittelt. Stattdessen werden verschiedene Kriterien miteinander in Bezug gesetzt. Einzelne Kriterien können als Filter genutzt oder mit einer bestimmten Gewichtung in das Ranking eingerechnet werden. Ein Beispiel für einen Filter wäre, dass man den Suchraum auf Datensätze beschränkt, die in einem bestimmten technischen Format vorliegen. Eine andere Variante wäre, dass man sich im Ranking zu einem gewissen Prozentsatz auf die inhaltliche Passung und zu einem anderen Prozentsatz auf ein entsprechend zu definierendes Popularitätskriterium beziehen könnte. Verfahren, die Suchsysteme mit mehreren Kriterien betrachten, sind unter Begriffen wie multikriterielles Matching oder Polyrepräsentation bekannt geworden. Die Gewichtung

31 Vgl. Bullin 2020, 10–12.
32 Vgl. Tuytelaars 2007, 177–280.
33 Vgl. Goodfellow 2016, 326–366.
34 Vgl. Brin 1998, 109–111.

und Verrechnung der einzelnen Kriterien ist dabei ein wichtiges Forschungsgebiet, das im Forschungsfeld „Learning to Rank“ Gegenstand intensiver Forschung war und ist. Eine andere Umsetzung der Suche auf Basis verschiedener Kriterien ist die facettierte Suche mit verschiedenen Kriterien zum Filtern und Sortieren, wie wir sie z. B. aus Online-Shops oder Gebrauchtwagenbörsen kennen[35].

Bei all diesen Überlegungen zu Retrievalmodellen darf nicht übersehen werden, dass wichtige Entscheidungen bereits in der Vorbereitung der Dokumente bzw. Datensätze und in einer gegebenenfalls durchzuführenden Aufbereitung der Anfrage liegen. Bei der Vorbereitung der Dokumente sind wichtige Schritte im Bereich des Tokenizing zu sehen (was wird als Wort betrachtet?), aber auch im Bereich der Stoppworteliminierung oder der Stamm- bzw. Grundformreduktion (Lemmatisierung). Bei der Stoppworteliminierung werden gezielt Begriffe aus der Betrachtung ausgeschlossen, die grammatikalische oder syntaktische Funktionen im Text übernehmen und daher als Begriff keine Rückschlüsse auf den Inhalt des Dokumentes erlauben. Stoppwortlisten für das Englische beinhalten in der Regel einige hundert Wörter. Heute wählen Suchmaschinen oft den Ansatz, Stoppworte mit zu indexieren und dann im Rahmen der Anfragebearbeitung entsprechend gering zu gewichten – was z. B. durch BM25 praktisch automatisch erfolgt. Für die Stamm- und Grundformreduktion werden in der Literatur viele Algorithmen vorgeschlagen.[36] Eine solche Reduktion vereinfacht die Suche und vereinheitlicht die Begriffswelt. Sie führt aber auch dazu, dass bestimmte Wortformen nicht mehr ohne Weiteres gezielt recherchiert werden können. Daher stellt sich insbesondere bei einer Suche nach Forschungsdaten die Frage, wie hier konkret vorgegangen werden sollte. Ein weiterer Ansatzpunkt für Optimierungen ist die Anfrage selbst. Hier kommen häufig sogenannte Erweiterungstechniken zum Einsatz, bei denen Anfragen auf Basis eines kontrollierten Vokabulars oder auf Basis statistischer Modelle mit bedeutungsähnlichen Begriffen erweitert werden. Dadurch kann in der Regel der Recall verbessert werden, die Precision leidet aber häufig unter derartigen Ansätzen.

Erst das zielgerichtete Zusammenspiel von Retrievalmodellen, entsprechenden Vorverarbeitungsschritten für die Dokumente sowie geeigneten Erweiterungstechniken für die Anfragen schafft in der Regel die Basis für eine leistungsfähige, dem konkreten Anwendungsfeld angemessene Suchlösung.

35 Vgl. Tunkelang 2009, 39–43.

36 S. z. B. https://snowballstem.org.

5 Umsetzung von Suchsystemen

Die Umsetzung der im vierten Kapitel beschriebenen Modelle benötigt speziell auf die Suche ausgerichtete Implementierungstechniken, welche die Suche in großen Datenmengen effizient ermöglichen. Eine in vielen Implementierungen eingesetzte Datenstruktur ist die *invertierte Liste*.

Die Invertierung beruht auf der Überlegung, dass normalerweise die Dokumente oder Datensätze, die als Forschungsdaten bereitstehen, zusammen mit den in ihnen enthaltenen Wörtern abgespeichert werden. Ein Dokument ist dabei praktisch eine Liste von Worten. Eine invertierte Liste hingegen legt für jedes Wort eine Liste der Dokumente an, in denen dieses Wort enthalten ist. Der Hintergrund dieser Datenstruktur ist, dass gewöhnliche Suchanfragen einige wenige Worte umfassen. Bei den im vorherigen Kapitel betrachteten Modellen müssen bei der Bearbeitung einer Anfrage nun „nur" die invertierten Listen zu den wenigen Anfragebegriffen durchlaufen werden. Dies erlaubt eine effiziente Bearbeitung, da eine gezielte Konzentration auf die potenziell relevanten Dokumente möglich ist. Für eine Anfrage „Goethe Weimar Brief" müssen so nur die drei Listen zu den Anfragebegriffen durchlaufen werden, wobei z. B. die Werte für die BM25-Formel zu den einzelnen Dokumenten, die in den Listen enthalten sind, berechnet werden. Eine Strategie zur weiteren Optimierung ist dann, die Listen nach Dokument-IDs zu sortieren und so die Berechnungen in einem parallelen Durchlauf vornehmen zu können. Es existieren aber noch zahlreiche weitere Verfahren zur Optimierung invertierter Listen, die z. B. bei Witten oder Büttcher beschrieben werden.[37]

Zwei problematische Aspekte für invertierte Listen sind sehr große Datenmengen und hohe Änderungsraten. Im Bereich der Websuche müssen die invertierten Listen z. B. verteilt verwaltet werden, um die Datenmenge bewältigen zu können. Grundsätzlich wäre es dabei möglich, eine invertierte Listenstruktur zu verteilen, indem alle beteiligten Rechner jeweils für eine bestimmte Teilmenge von Wörtern zuständig wären und die entsprechenden Listen verwalten würden. Problematisch wäre dabei allerdings, dass Rechner, die für populäre Anfragebegriffe zuständig wären, schnell überlastet werden könnten. Üblicher ist daher eine Aufteilung der Gesamtmenge der Dokumente auf verschiedene Rechencluster. Auf diesen Rechenclustern werden dann jeweils eigene invertierte Listen (z. B. für bestimmte Regionen) verwaltet. Anfragen werden dann ggf. parallel auf mehreren Clustern bearbeitet und die erzielten Ergebnisse kombiniert.

Gerade im Bereich der Websuche stellen Aktualisierungen einzelner Dokumente ein weiteres Problem dar, da hier ggf. viele invertierte Listen zu modifizieren wären. Man arbeitet daher oft mit größeren stabilen invertierten Listen und Differenzlisten für aktuelle Änderungen und Löschungen. Damit werden Anfragen und Änderun-

37 Vgl. Witten 1999, 114–127; Büttcher 2010, 174–227.

gen zu komplexen Abläufen, welche in der Regel mehrere Datenstrukturen und Rechner betreffen und von Verteilerknoten orchestriert werden.[38]

6 Architekturen von Suchlösungen

Während bei Websuchmaschinen allein die Menge der zu indexierenden Dokumente eine verteilte Lösung erzwingt, ergeben sich bei Forschungsdaten verteilte Architekturen oft auf Basis der technisch, rechtlich und auch historisch bedingten dezentralen Verwaltung der Daten. Für die Umsetzung der Suche in und nach Forschungsdaten stehen dabei unterschiedliche Ansätze und Architekturen zur Verfügung.

Ausgangspunkt der Überlegung ist, dass eine Nutzerin bzw. ein Nutzer einen Informationsbedarf hat. Die Forschungsdatenbestände, auf die sich die entsprechende Suche beziehen sollte, sind über mehrere Systeme zur Verwaltung von Forschungsdaten verteilt. Die Systeme verfügen in der Regel über eigene Such- und Exportschnittstellen, wobei die Leistungsfähigkeit dieser Schnittstellen von System zu System stark variieren kann.

Ein erster Ansatz wäre nun die „direkte Suche", bei der die Nutzerin bzw. der Nutzer die relevanten Systeme selbst recherchiert und dann deren jeweilige Suchschnittstelle zur individuellen Abfrage nutzt. Hier muss die Nutzerin bzw. der Nutzer die Recherche nach potenziell relevanten Beständen selbst durchführen, sich selbst mit den verschiedenen Suchschnittstellen auseinandersetzen und selbst die Kombination der Ergebnisse vornehmen.

Der Aufwand, der für die Nutzerin bzw. den Nutzer bei der direkten Suche durch die manuelle Arbeit mit den diversen Suchanwendungen anfällt, kann durch andere Suchkonzepte verringert werden. *Metasuchmaschinen* binden z. B. mithilfe von Suchschnittstellen anderer Suchlösungen deren Datenbestände in eine Suchanfrage mit ein.[39] Dadurch ist mit einer einzelnen Suchanfrage, die von der Metasuchlösung mithilfe von transformierten Anfragen an die anderen Suchlösungen weitergeleitet wird, die Durchsuchung vieler Datenbestände möglich. Somit können Suchergebnisse aus vielen Datenbeständen zurückgegeben und nachgewiesen werden. Probleme treten hier jedoch ggf. bei der Umsetzung der Metasuchlösung auf, da sich die Transformation der Suchanfragen für die anderen Suchlösungen zeit- und kostenaufwendig gestalten kann und außerdem ein übergreifendes Ranking der Suchergebnisse kaum möglich ist – zu einzelnen Suchergebnissen ist zwar der Rang, zumeist jedoch nicht die detaillierte Bewertung oder deren Berechnung ver-

38 Vgl. Cambazoglu 2015, 10–20.
39 Vgl. Lewandowski 2005, 25–26.

fügbar. Auf der anderen Seite ist bei einer Metasuchlösung die bei Forschungsdaten oft bedeutsame Problematik von Zugriffsrechten in der Regel gut handhabbar, da die Daten, in denen gesucht wird, weiterhin in den originalen Datenbeständen vorliegen und die zentrale Suchlösung nur die Suchanfrage und die Zugriffsrechte der Nutzerin bzw. des Nutzers weiterleitet. Praktische Anwendung findet das Konzept der Metasuchlösung z. B. in der Suchmaschine Metager[40] oder der Federated Content Search von CLARIN.[41]

Eine weitere Alternative ist das Konzept des *Gathering*. Hierbei leitet die primäre Suchlösung nicht die Anfragen weiter, sondern nutzt die Exportschnittstellen der anderen Systeme, um einen gesammelten Index aufzubauen, auf dem die Anfragen ausgeführt werden können. Die Daten werden somit zentral gesammelt, weshalb eine einheitliche Suchansicht und ein übergreifendes Ranking angeboten werden kann. Andererseits tritt hier das Problem der Zugriffsrechte verstärkt auf, da die Rechte der Nutzerin bzw. des Nutzers für die jeweiligen Datenbestände einzeln geprüft werden müssen. Weiterhin fällt für die Suchlösung hier in der Regel ein deutlich höherer Speicherplatzbedarf an, der für die Verwaltung des Index aller Datenbestände notwendig wird. Außerdem ist die Synchronisierung der Daten problematisch, da Neuerungen oder Änderungen in den Datenbeständen unter Umständen nicht an die zentrale Suchlösung weitergegeben werden und somit die Anfrage auf veralteten Daten ausgeführt wird. Neben der Suchlösung der „Generischen Suche“ von DARIAH-DE[42] findet sich das Gathering-Konzept unter anderem in dem „B2Find“-Discovery-Service von EUDAT.[43]

Die Entwicklung hin zu verteilten Systemen, die auf Basis von Architekturen wie Metasuchmaschinen oder Gathering umgesetzt werden, erfordert eine gemeinsame Sprache und ein einheitliches Protokoll, um den Austausch der Daten über die Schnittstellen zu ermöglichen. Frühe Systeme, die über derartige Schnittstellen Daten und Suchanfragen austauschen, waren oft im Bereich von Bibliotheken angesiedelt. Einheitliche Metadatenformate (vgl. Abschnitt 3) bildeten die Basis für die Entwicklung hin zu einer gemeinsamen Suchschnittstelle – im Bibliothekswesen beispielsweise durch den MARC-Standard vertreten, welcher bereits in den 1960er Jahren in den USA entwickelt wurde.

Ein frühes Protokoll war der Z39.50-Standard, ebenso in den USA von der Library of Congress initiiert. Z39.50 definiert ein Client-Server-System, worin der Server an mehrere Datenbanken gekoppelt ist und über das Protokoll Anfragen vom Client an den Server gesendet werden können. Da Z39.50 vor dem Durchbruch von Web-Technologien entwickelt wurde, werden Anfragen und Antworten zwischen Client

40 S. https://metager.de/.
41 S. https://www.clarin.eu/content/federated-content-search-clarin-fcs.
42 S. https://search.de.dariah.eu/search/.
43 S. https://eudat.eu/services/b2find.

und Server über ein eigenes Protokoll gesendet, welches nicht kompatibel mit gegenwärtig gebräuchlichen Web-Standards ist. Diese Einschränkung wurde in einer Weiterentwicklung des Z39.50-Standards behandelt, genannt Z39.50 International Next Generation, in welcher weit verbreitete Standards wie HTTP, URI und XML eingesetzt werden.

Innerhalb dieser neuen Version wird insbesondere die explizite Trennung in ein Protokoll Search/Retrieve via URL (SRU) und eine Anfragesprache Contextual Query Language (CQL) vorgenommen. Anfragen werden mithilfe dieser beiden Technologien via HTTP versendet, in einer standardisierten Syntax in CQL ausgedrückt und mittels XML übermittelt. Die Öffnung hin zu gängigen Technologien erlaubt den Einsatz von SRU und CQL nicht nur im Bibliothekswesen, sondern auch in anderen Einsatzgebieten wie etwa Museen oder – noch generischer – in der Internetsuche allgemein. Anfragen in SRU und CQL basieren im Übrigen auf weiteren, inhaltlichen Standards wie z. B. Dublin Core.

Konkret bieten z. B. sowohl der Bayerische Bibliotheksverbund als auch die Deutsche Nationalbibliothek eine SRU Schnittstelle für ihre Datenbestände an. Selbst der ältere Standard Z39.50 kann in modernen Systemen benutzt werden, beispielsweise sieht die Literaturverwaltungssoftware Citavi[44] diese Schnittstelle weiterhin vor, um direkt in den Bibliothekskatalogen unterschiedlicher Institutionen zu suchen.

7 Beispiele für Suchlösungen

Nachdem sowohl die grundlegenden Protokolle und Schnittstellen als auch die algorithmischen Hintergründe der Umsetzung von Suchsystemen präsentiert wurden, sollen nun sowohl Frameworks und Programmbibliotheken vorgestellt werden, die in der Praxis eingesetzt werden, als auch konkrete Suchsysteme, welche aktuell im Einsatz sind.

7.1 Frameworks und Bibliotheken

Für die praktische Umsetzung der Suche in Daten stehen Softwarelösungen auf unterschiedlichen Ebenen zur Verfügung. Die Bibliothek Lucene[45] ermöglicht die Suche in Daten mit Fokus auf Text. Sie ist in Java geschrieben und als Projekt der Apache Software Foundation entwickelt worden und bildet gleichermaßen die Basis für

44 S. https://www.citavi.com/de.
45 S. https://lucene.apache.org/core.

andere Bibliotheken und Frameworks. In Lucene sind verschiedene IR-Modelle wie BM25 umgesetzt, so dass sie einfach zur Berechnung der Suchergebnisse verwendet werden können.

Eng verknüpft mit Lucene ist Solr,[46] ebenfalls als Apache-Projekt entwickelt. Diese Plattform ist auf den Einsatz im Unternehmensbereich ausgerichtet, mit größeren Netzwerken und auf verteilten Systemen. Solr verwendet Lucene als Basis und erlaubt dessen Anwendung mittels Administrationsoberflächen, Analysetools und weiteren Funktionen. Das Projekt Blacklight[47] bietet eine Schnittstelle, um die Suchfunktionalitäten von Solr für eine Vielzahl von Anwendungsfällen mittels Ruby on Rails als Webapplikation bereitzustellen, beispielsweise für raumbezogene Daten oder im Kontext von Bibliotheken und Museen.

ElasticSearch[48] nutzt ebenso wie Solr die Funktionalitäten von Lucene im Hintergrund, hebt sich jedoch durch einen web-basierten Workflow mittels REST APIs von Solr ab. Die Abfragen sind in ElasticSearch in JSON verfasst und können durch verschiedene Programmiersprachen gestellt werden, da eine Vielzahl an Clients verfügbar ist.

Funktional deutlich weitreichender als die angesprochenen Bibliotheken wurde das Suchsystem vufind[49] entworfen, um im Bereich von Bibliotheken den traditionell genutzten Online Public Access Catalogue (OPAC) zu ersetzen. Mittels vufind kann in einem System nicht nur der Bestand einer Bibliothek zur Suche verfügbar gemacht werden, sondern es können die Bestände vieler Institutionen und Bibliotheken eingebunden sowie mittels diverser Schnittstellen zugänglich gemacht werden, etwa OAI oder das bereits erwähnte Solr.

Diese Technologien unterscheiden sich im Umfang der angebotenen Funktionalitäten und dadurch auch durch ihre Komplexität. Große Suchlösungen wie vufind benötigen eine Vielzahl an zugehöriger Software und sind dadurch nicht nur in ihrer Installation zeitaufwendig, sondern verursachen durch die Wartung und gegebenenfalls Anpassung an die jeweiligen Anforderungen weitere Kosten. Schlankere Suchlösungen, die durch Bibliotheken wie Lucene ohne große Frameworks implementiert sind und etwa nur die Anbindung über SRU und CQL an die Daten anbieten, können für die Datenbestände von kleineren Institutionen eine lohnenswerte Alternative sein, falls die Anwendungszwecke hier auch verhältnismäßig schmal gehalten werden.

46 S. https://lucene.apache.org/solr.
47 S. https://projectblacklight.org/.
48 S. https://www.elastic.co/de/products/elasticsearch.
49 S. https://vufind.org/vufind.

7.2 Exemplarische Systeme

In der Praxis eingesetzte Systeme gibt es sowohl auf nationaler als auch auf internationaler Ebene. Dementsprechend sollen hier exemplarisch mehrere Projekte vorgestellt werden, um die Ziele der jeweiligen Suchsysteme aufzuzeigen.

Die Generische Suche der digitalen Forschungsinfrastruktur DARIAH-DE[50] ist ein Suchsystem, welches auf nationaler Ebene die Suche in Sammlungen verschiedener Institutionen ermöglicht. Da DARIAH-DE als Föderationsinfrastruktur heterogene Daten aus unterschiedlichen Quellen in einem Suchsystem zugänglich macht, kann das System als Gathering-Architektur charakterisiert werden. Von besonderer Bedeutung sind für die Generische Suche Forschungsdaten aus dem geisteswissenschaftlichen Bereich, die als Sammlungen in das Suchsystem integriert werden. Die Heterogenität dieser Sammlungen ist eines der Merkmale geisteswissenschaftlicher Forschungsdaten, das im Rahmen der Digital Humanities derzeit häufig adressiert wird. In der Generischen Suche wird dieses Problem mit Komponenten wie der Collection Registry, für die Eintragung und Beschreibung von Sammlungen, und dem Data Modeling Environment, zur Modellierung und Abbildung von Daten und deren Metadatenschemata, behandelt.[51]

Der Einsatz von SRU und CQL in einer Metasuchlösung wird beispielsweise in der Federated Content Search (FCS) von CLARIN, einer Forschungsinfrastruktur für die text-bezogenen Geistes- und Sozialwissenschaften, umgesetzt. In dieser Applikation wird mittels des SRU Protokolls in der Anfragesprache CQL eine Anfrage vom Client zu einem Endpoint weitergeleitet, wo die CQL-Anfrage so übersetzt wird, dass die lokale Suchlösung diese Anfrage weiterverarbeiten kann.

Auf internationaler Ebene wird der B2Find-Service durch die EUDAT Initiative bereitgestellt. B2Find ermöglicht die explorative Suche und das Entdecken von Daten über die Suche in Metadaten aus Forschungssammlungen. Die Forschungsdaten werden über einen Katalog aus in EUDAT verzeichneten Services und Metadaten für die Suche vorbereitet. B2Find stellt somit nicht nur die Suche in den Volltexten der Forschungsdaten zur Verfügung, sondern auch die Suche mittels der Metadaten nach facettierten, raumbezogenen und zeitlichen Eigenschaften und erlaubt damit die Filterung nach diesen Kategorien.

Die European Open Science Cloud (EOSC)[52], und insbesondere deren EOSC-hub, umfasst neben einer Schnittstelle zu B2Find eine Vielzahl von Services und anderen Ressourcen für die Forschung mit dem Ziel des Zugriffs, der Verarbeitung und der Analyse von Daten. Ein Fokus der EOSC ist die Betonung von Open Science – wel-

50 Vgl. Gradl 2017, 25–26.
51 Vgl. Gradl 2016, 123–126.
52 S. https://www.eosc-portal.eu.

che durch die wesentliche Rolle von Konzepten wie Open Data, Open Source oder Open Access geprägt ist.

Im Bereich spezifischer Suchlösungen kann erneut exemplarisch auf das Projekt Blacklight, und insbesondere dessen Ausprägung GeoBlacklight[53] verwiesen werden, welches Suchanwendungen zu raumbezogenen Daten ermöglicht, etwas das Big Ten Academic Alliance Geoportal.[54] Mithilfe dieses Geoportals können Institutionen aus den Vereinigten Staaten Zugang zu Tausenden von Karten-Datensätzen bereitstellen, inklusive Webservices und Zugangsmechanismen zu den Daten.

8 Zusammenfassung

Im vorliegenden Beitrag wurde ein Überblick über diverse Konzepte und Ansätze zum Data Retrieval für Forschungsdaten gegeben. Erfolgreiche Ansätze müssen die Nutzerin bzw. den Nutzer mit ihrem bzw. seinem Informationsbedarf im Blick haben. Die bestehenden Modelle zur inhaltsbasierten Suche und insbesondere Ansätze zur Kombination verschiedener Kriterien bilden dabei eine gute formale Basis. Für die Nutzung und Umsetzung konkreter Suchlösungen existieren Programmbibliotheken, Frameworks und Systeme, die nachgenutzt werden können. Eine Herausforderung ist der Zwiespalt zwischen der fachlich bedingten Heterogenität von Forschungsdaten und den zugehörigen Metadaten sowie dem Wunsch nach einer übergreifenden Recherchierbarkeit, die es erlaubt, interdisziplinäre Zusammenhänge und Perspektiven zu adressieren.

Literatur

Letztes Abrufdatum der Internet-Dokumente ist der 15.11.2020.

Brin, Sergey und Lawrence Page. 1998. „The anatomy of a large-scale hypertextual Web search engine." *Computer Networks and ISDN Systems* 30 (1–7): 107–117. doi:10.1016/S0169-7552 (98)00110-X.

Bullin, Martin und Andreas Henrich. 2020. „Die inhaltsbasierte Bildsuche und Bilderschließung: Ansätze und Problemfelder." In *Bilddaten in den digitalen Geisteswissenschaften* (in Erscheinen), hg. v. Canan Hastik und Philipp Hegel. Wiesbaden: Harrassowitz.

Büttcher, Stefan, Charles L. A. Clarke und Gordon V. Cormack. 2010. *Information retrieval: Implementing and evaluating search engines*. Cambridge, Mass., London: MIT Press.

53 S. https://geoblacklight.org/.

54 S. https://geo.btaa.org/.

Cambazoglu, B. B. und Ricardo Baeza-Yates. 2015. „Scalability Challenges in Web Search Engines." *Synthesis Lectures on Information Concepts, Retrieval, and Services* 7 (6): 1–138. doi:10.2200/S00662ED1V01Y201508ICR045.

Croft, W. B., Donald Metzler und Trevor Strohman. 2010. *Search engines: Information retrieval in practice*. Boston, Mass.: Pearson.

Ferber, Reginald. 2003. *Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. 1. Aufl. Heidelberg: dpunkt-Verl.

Flanders, Julia und Fotis Jannidis. 2015. „Data Modeling." In *A New Companion to Digital Humanities*. Bd. 33, hg. v. Susan Schreibman, Ray Siemens und John Unsworth, 229–237. Chichester: John Wiley & Sons, Ltd.

Frants, V., Jacob Shapiro und Vladimir G. Voiskunskii. 1997. *Automated information retrieval: Theory and methods*. Library and information science. San Diego, Calif., London: Academic Press.

Goodfellow, Ian, Yoshua Bengio und Aaron Courville. 2016. *Deep learning. Adaptive computation and machine learning.* Cambridge, Massachusetts: The MIT Press.

Gradl, Tobias, Anna Aschauer, Swantje Dogunke, Lisa Klaffki, Stefan Schmunk und Timo Steyer. 2017. „Daten sammeln, modellieren und durchsuchen mit DARIAH – DE." In *Konferenzabstracts DHd 2017 Digitale Nachhaltigkeit*, 22–27. Bern.

Gradl, Tobias und Andreas Henrich. 2016. „Extending Data Models by Declaratively Specifying Contextual Knowledge." In *Proceedings of the 2016 ACM Symposium on Document Engineering – DocEng '16*, hg. v. Robert Sablatnig und Tamir Hassan, 123–126. New York: ACM Press.

Lewandowski, Dirk. 2005. *Web Information Retrieval: Technologien zur Informationssuche im Internet.* Frankfurt am Main: Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis.

Manning, Christopher D., Prabhakar Raghavan und Hinrich Schütze. 2008. *An introduction to information retrieval*. Cambridge: Cambridge University Press.

Marchionini, Gary. 2006. „Exploratory search." *Commun. ACM* 49 (4): 41. doi:10.1145/1121949.1121979.

Marchionini, Gary und Ryen White. 2007. „Find What You Need, Understand What You Find." *International Journal of Human-Computer Interaction* 23 (3): 205–237. doi:10.1080/10447310701702352.

Neuroth, Heike. 2017. „Bibliothek, Archiv, Museum." In *Digital Humanities* 64/3, hg. v. Fotis Jannidis, Hubertus Kohle und Malte Rehbein, 213–222. Stuttgart: J. B. Metzler.

Ogilvie, Paul und Jamie Callan. 2003. „Combining document representations for known-item search." In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval – SIGIR '03*, hg. v. Charles Clarke, Gordon Cormack, Jamie Callan, David Hawking und Alan Smeaton, 143. New York: ACM Press.

Ponceleon, Dulce B. und Malcolm Slaney. 2011. „Multimedia Information Retrieval." In *Modern Information Retrieval – The Concepts and Technology behind Search*, hg. v. R. Baeza-Yates und Berthier Ribeiro-Neto. 2nd ed., 587–639. Harlow: Addison Wesley.

Raieli, Roberto. 2016. „Introducing Multimedia Information Retrieval to libraries." *JLIS.it* 7 (3): 9–42. doi:10.4403/jlis.it-11530.

Rixen, Stephan. 2018. „Zukunftsthema: Zum Umgang mit Forschungsdaten." In Forschung & Lehre. 2/18, hg. v. Deutscher Hochschulverband.

Robertson, S. E. und K. S. Jones. 1976. „Relevance weighting of search terms." *J. Am. Soc. Inf. Sci.* 27 (3): 129–146. doi:10.1002/asi.4630270302.

Robertson, S. E., S. Walker und M. M. Beaulieu. 1999. „Okapi at TREC-7: automatic ad hoc, filtering, VCL and interactive track." In *The Seventh Text REtrieval Conference (TREC-7)*, 253–264. Gaithersburg: National Institute of Standards and Technology.

Schöch, Christof. 2017. „Aufbau von Datensammlungen.“ In *Digital Humanities* 28/2, hg. v. Fotis Jannidis, Hubertus Kohle und Malte Rehbein, 223–233. Stuttgart: J. B. Metzler.

Sparck Jones, K., S. Walker und S. E. Robertson. 2000. „A probabilistic model of information retrieval: development and comparative experiments.“ *Information Processing & Management* 36 (6): 809–840. doi:10.1016/S0306-4573(00)00016-9.

Tunkelang, Daniel. 2009. „Faceted Search.“ *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1 (1): 1–80. doi:10.2200/S00190ED1V01Y200904ICR005.

Tuytelaars, Tinne und Krystian Mikolajczyk. 2007. „Local Invariant Feature Detectors: A Survey.“ *FNT in Computer Graphics and Vision* 3 (3): 177–280. doi:10.1561/0600000017.

Wilkinson, Mark D., Michel Dumontier, I. J. J. Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg et al. 2016. „The FAIR Guiding Principles for Scientific Data Management and Stewardship.“ *Scientific data* 3: 160018. doi:10.1038/sdata.2016.18.

Witten, I. H., Alistair Moffat und Timothy C. Bell. 1999. *Managing gigabytes: Compressing and indexing documents and images*. 2nd ed. (Morgan Kaufmann series in multimedia information and systems.) San Francisco, Calif. Morgan Kaufmann Publishers.

Zhai, Chengxiang und John Lafferty. 2004. „A study of smoothing methods for language models applied to information retrieval.“ *ACM Trans. Inf. Syst.* 22 (2): 179–214. doi:10.1145/984321.984322.