Die Wahl des Suchbegriffs in anfragebasierten Systemen zur Erkennung bibliographischer Dubletten

Selecting the right search term in query-based systems for deduplication
Sélection de mot de recherche dans les systèmes basés sur des requêtes en vue de détecter les doublons bibliographiques

Harald Jele

Bei der Wahl eines günstigen Suchbegriffs zur Erkennung bibliographischer Dubletten sind im Wesentlichen drei Ansätze erkennbar. Stoppwörter werden in allen dreien ausgeschlossen, anschließend wird (1) der erste Begriff eines Eintrags gewählt, der aufgefunden wird oder es wird (2) jener gewählt, der die kleinste Treffermenge hervorruft oder letztlich (3) findet der Begriff Verwendung, dessen Treffermenge unter einem definierten Schwellwert liegt.
Diese drei Vorgehensweisen werden hier miteinander verglichen. Die Ergebnisse stammen aus Messreihen, die mit Titeldaten aus dem österreichischen Bibliothekenverbund durchgeführt wurden.

Essentially three approaches could be identified when choosing a proper search term to detect bibliographic duplicates. Stop words are excluded in all of them, then (1) just the first term of an entry will be selected or (2) that term is selected, which produces the smallest number of hits or finally (3) that term will be used, which has a certain number of hits below a defined threshold.
These three procedures are compared with each other here. The results derive from series of measurements done with bibliographic data from the Austrian Central Catalog.

On a pu identifier essentiellement 3 approches dans la procédure de selection d'un terme susceptible de détecter les doublons bibliographiques. Ces trois approches excluent les mots vides (stopwords), ensuite (1) on selectionne le premier terme d'une entrée ou bien (2) on choisit celui pour lequel on obtient le plus petit nombre de résultats ou bien finalement (3) on prend le terme pour lequel le nombre de résultats se situe sous un seuil défini. Ces trois procédures sont ici comparés les unes aux autres.
Les résultats sont issus de séries de mesures effectuées sur une base de données bibliographiques du catalogue de la Bibliothèque nationale autrichienne.

1 Vorwort

Versucht man, den hier vorgelegten Beitrag anhand der aktuellen Literatur einzuordnen, mag man schnell der Versuchung unterliegen, die hier diskutierten Methoden als nicht mehr zeitgemäß abzutun. Mit diesem kurzen Vorwort, das einige Hinweise auf weitere, relevante Literatur leistet, soll darauf hingedeutet werden, dass ein solches Urteil nicht oder nicht in jedem Fall zutreffend ist.

Bei der Deduplizierung bibliographischer Daten ist man im Besonderen darauf bedacht, nicht zuviele Titeleinträge miteinander vergleichen zu müssen. Dies liegt einerseits natürlich darin begründet, dass die Dublettenberechnung großer Treffermengen aufwändig ist. Effektiver ist natürlich jeder Ansatz, der die Menge an zu prüfenden Titeln gering hält. Je kleiner diese Men-

ge wird, umso größer ist im Umkehrschluss jedoch die Wahrscheinlichkeit, dass Dubletten übersehen werden. Bei jeder getätigten Einschränkung des „Suchraums“, in dem eine Prüfung auf Dubletten stattfindet, stehen sich somit Effizienz und Zuverlässigkeit mit ihren Erfordernissen diametral gegenüber.

Der hier besprochene Ansatz, stichwortbasiert vorzugehen, ist einer, der der Realität bibliographischer Datenbanken durchaus gerecht wird. Diese Realität ist davon geprägt, dass aus sehr unterschiedlichen Datenbanken, die lokal nicht zugänglich sind, Titel in ein lokales System integriert werden. Dabei werden die entfernten Datenbanken über Schnittstellen (wie Z39.50) und APIs (=Application Programming Interfaces wie REST (=Representational State Transfer) oder SOAP (=Simple Object Access Protocol)) eingebunden. Es ist demnach ein Ansatz notwendig, der in solchen Umgebungen sinnvoll und zweckmäßig eingesetzt werden kann (vgl. Schneider 1999). Der Ansatz, über ein günstig gewähltes Stichwort jene Titeldaten automatisiert zu laden, die für eine weitere Prüfung in Frage kommen, ist ein solcher.

Alternativ zu einem Stichwort-basierten Ansatz finden sich in der aktuellen Literatur vorwiegend besprochene Methoden wie SNM (=Sorted Neighborhood Method) (Hernández & Stolfo 1995 und Yan u. a. 2007) oder die „Blocking Method“ (siehe Draisbach & Naumann 2009). Dabei werden den Daten Zeichenketten entnommen („Substrings“ aus z. B. Autor- und Titeleinträgen), diese anschließend zu einer Zeichenkette zusammengefügt und alphanummerisch sortiert. Die so gebildeten „Sorting keys“ werden in weiterer Folge dazu verwendet, in ihrer Umgebung mittels sogenannter Suchfenster nach ähnlichen Einträgen zu fahnden. Die dabei angewandten Algorithmen sind vor allem „Windowing“ (vgl. dazu ebenso Hernández & Stolfo 1995 sowie Hernández & Stolfo 1998) und „Blocking“ (Ananthakrishna u. a. 2002, Baxter u. a. 2003 sowie Bilenko u. a. 2006).

Der Zugriff auf die Sorting keys kann nur dann erfolgen, wenn diese in den entfernten Datenbanken vorhanden sind oder die Möglichkeit besteht, diese lokal aus der Gesamtmenge zu bilden. Beides ist jedoch in der Regel nicht der Fall. Zweiteres ist zudem unpraktikabel, da die angefragten Datenbanken zumeist keine statisch gleichbleibende, sondern eine stetig wachsende Menge an Titelinformationen tragen. Der von Draisbach und Naumann (vgl. 2009, S. 2) vorgeschlagene Weg, die Sortierschlüssel zur Laufzeit der Prozesse lokal im Arbeitsspeicher zu halten, missachtet den Umstand, dass die Datenmenge in durchschnittlich großen bibliographischen Datenbanken für ein solches Vorgehen deutlich zu groß ist. Die Schlüssel müssen daher in Datenbankdateien gespeichert sein. Diese Methoden, wenngleich sie State-of-the-Art sind, scheiden aus diesen Gründen im praktischen Einsatz daher aus, wenn online auf entfernte Daten zugegriffen wird. Bei der Integration bibliographischer Datenbanken in ein lokales System sind diese jedoch für den Batch-Betrieb durchaus von Relevanz.

Daneben werden in der aktuellen Literatur Ansätze diskutiert, die auf Hashing-Methoden basieren. Für textbasierte Anwendungen scheinen besonders die Implementierungen einer Indexierung mittels „Locality-sensitive hashing“ (LSH) interessant zu sein, wenngleich für die Indexierung bibliographischer Daten bislang keine entsprechend publizierten Ergebnisse vorliegen. Beim LSH kommt es zu einer Kombination (Addition) mehrerer, einfach gebildeter Hash-Werte. Jede Kollision zweier oder mehrerer Werte wird als eine Ähnlichkeit zwischen den betroffenen Datensätzen gesehen. Im günstigen Fall werden die Hash-Werte demnach so gebildet, dass sie für die Gesamtmenge nicht singuläre Werte darstellen, sofern davon ausgegangen werden kann, dass die Gesamtmenge Dubletten beinhaltet (Paulevéa u. a. 2010 sowie Stein & Potthast 2006). Für die Relevanz eines solchen Ansatzes gelten die selben Umstände wie für den Zugriff und die Auswertung von Sorting keys: Im praktischen Einsatz kann ein solcher Ansatz immer dann interessant sein, wenn dieser in einem lokalen (und nicht entfernten) System realisiert ist. Anderenfalls müssen in jeder der entfernten bibliographischen Datenbanken diese Hash-Werte gebildet und zugänglich gehalten werden.

2 Einleitung

Anfragebasierte Systeme sind im Umfeld der Erkennung bibliographischer Dubletten durch den spezifischen Umstand gekennzeichnet, dass aufgrund einer *Abfrage* eine Treffermenge zustande kommt, die als Ausgangspunkt einer weiteren *Anfrage* gesehen wird, die automatisiert durchgeführt wird. Das Ziel, das mit der Bildung einer solchen Anfrage verfolgt wird, ist die Auffindung sämtlicher Dubletten eines Datenbestandes zu jenen Titeln, die in der Abfragemenge enhalten sind.
Die eigentliche Abfrage, respektive die daraus entstehende Abfragemenge, kann dabei einerseits aus der konkreten Systemabfrage eines Benutzer entstehen. Andererseits kann diese auch im Zuge einer Zusammenführung von Datenbeständen durch ein sequentielles Abarbeiten von Titeldaten im Batch-Betrieb gebildet werden.
Bei der Abfrage durch einen Benutzer kann das Ziel einer solchen Vorgehensweise sowohl eine dublettenfreie Titelanzeige als auch eine Zusammenführung von Exemplardaten zu dubletten Titeln sein, um beispielsweise nachfolgende Bestellvorgänge der Orts- und Fernleihe zu vereinfachen.
Ein weiterer Einsatz eines solchen Verfahrens kann auch die Katalogisierung von Werken in einem Bibliothekssystem sein. Dabei wird im Zuge des Speicherns eines Ti-

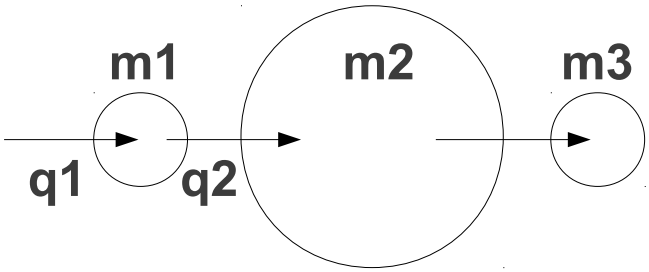

Abbildung 1: Schema eines Abfrage-Anfrage-Systems

teldatensatzes nach möglichen Dubletten gesucht. Diese werden im Fall des Auffindens solcher dem Bearbeiter angezeigt. Dieser Einsatzbereich wird hier jedoch aufgrund seiner spezifischen Implikationen nicht weiter verfolgt. Als ein Beispiel einer konkreten Anwendung zur Deduplizierung von Titeldaten im Prozess des Katalogisierens kann z. B. das von Schneider (1999) entwickelte System „ZACK“ gesehen werden.

In Abbildung 1 ist der weitere Vorgang schematisch dargestellt: Ein Benutzer generiert aufgrund seiner Abfrage *q1* eine Treffermenge {m1}, die dublettenfrei dargestellt werden soll. Die Treffermenge {m1} kann daher entweder durch geeignete Erkennungsverfahren dedupliziert oder zudem um jene Exemplare ergänzt werden, die in der Gesamtmenge des Datenbestands {m2} als Titeldubletten erkennbar sind. Dazu wird jeder Titeldatensatz der Menge {m1} gegen den Gesamtbestand {m2} ohne Zutun des Benutzers geprüft und die Menge {m3} gebildet, die um die Exemplare der mehrfach vorhandenen Titel aus {m2} angereichert ist. Ein solches Szenario ist – wie bereits angedeutet – für den Einsatz in einem Online-Katalog, der für angemeldete Benutzer Bestellfunktionen beinhaltet, vorstellbar.

Beim automatisierten Abgleich mehrerer Titelbestände ist eine solche Vorgehensweise wohl in den allermeisten Fällen durch einen Batchprozess realisiert. Dabei werden die Titeldaten der jeweils kleineren Menge {m1} sequentiell abgearbeitet und mit den Titeln aus der größeren Menge {m2} verglichen. Jedem einzelnen Titel aus {m1} werden in weiterer Folge jene Begriffe entnommen, aus denen sinnvolle Suchanfragen für die Menge {m2} gebildet werden können. Die sich durch eine nachfolgende Suche in {m2} ergebende Treffermenge {m3} wird für die Prüfung auf Dubletten verwendet.

In beiden Vorgehensweisen (im Online- sowie im Batch-Verfahren) liegt eine der wesentlichen Herausforderungen in der Wahl eines günstigen Suchbegriffs für die Durchführung der Anfrage *q2*.

3 Übliche Vorgehensweisen

Zur Bildung des Anfragebegriffs werden nicht alle verfügbaren Inhalte eines nach Kategorien strukturierten Datensatzes herangezogen. Üblicherweise gelangen ausschließlich die Kategorien zu den Verfasserangaben, den Körperschaftsnamen und jene zu den Titelangaben zur Anwendung. Hat ein Datensatz keine Angaben in zumindest einer dieser Kategorien, so wird er für die weitere Vorgehensweise ignoriert. Sind in einer dieser Kategorien jedoch Inhalte vorhanden, so werden diese im Anschluss auf weitere Verwendbarkeit geprüft. Nicht verwertbar sind dabei in erster Linie Begriffe bzw. Einträge, die als Stoppwort markiert sind.

Sowohl der einschlägigen Literatur als auch den entsprechenden, zugänglichen Routinen sind zur Wahl des Anfragebegriffs im Wesentlichen drei Ansätze zu entnehmen:

1. der erste Begriff, der kein Stoppwort ist, wird als Anfragebegriff verwendet,
2. jener Begriff, der kein Stoppwort ist und dessen Treffermenge am kleinsten ist, wird als Anfragebegriff verwendet,
3. der erste Begriff, dessen Treffermenge unter einem bestimmten Schwellwert liegt und der kein Stoppwort ist, wird als Anfragebegriff verwendet.

Diese drei Ansätze zur Ermittlung eines günstigen Anfragebegriffs werden hier miteinander verglichen und deren Vor- bzw. Nachteile ermittelt. Die Feststellung der dabei zu berücksichtigenden Aspekte geschieht im Wesentlichen durch Messung von Werten, die an konkreten und real existierenden bibliographischen Daten erhoben wurden, wie sie in einer durchschnittlich großen österreichischen Universitätsbibliothek vorkommen. Mit diesem Umstand, dass nämlich keine „künstlichen“ oder ausgewählten Daten zur Messung herangezogen werden, soll die Relevanz der Ergebnisse gegenüber eher theoretischer Werten verdeutlicht werden.

Die Ergebnisse beziehen sich auf bibliographische Daten zu Monographien und Reihenwerken. Grundsätzlich sind die Ergebnisse jedoch auch auf Daten zu Zeitschriftenartikeln und Beiträgen in Sammelwerken (d. h. auf unselbstständig erschienene Werke) anwendbar.

4 Die Messmethode

Die vollständigen bibliographischen Daten der Universitätsbibliothek Klagenfurt, die elektronische erfasst und online zugänglich im Format MAB2 sind, umfassen ca. 700.000 Titeldatensätze. Diese wurden dem Katalog vollständig entnommen.

Anschließend wurden jene Titel, die nicht monographische Werke beschreiben (das sind z. B. jene, die innerhalb einer Reihe oder mehrbändig begrenzt erschienen sind) so zusammengeführt, dass die Informationen zu Personen- und Körperschaftsnamen sowie zum Titel aus den hierarchisch übergeordneten Datensätzen in die jeweiligen „Stücke“ übernommen wurden. Dies geschah jedoch nur für den Fall, dass keine entsprechenden Angaben in den betroffenen Kategorien der untergeordneten Datensätze vorhanden waren. Dadurch sollte gewährleistet werden, dass möglichst wenige Titeldatensätze vom Umstand betroffen sind, dass diese aufgrund mangelnder Einträge ignoriert werden müssen.
Die Einträge zu den übergeordneten Datensätzen wurden anschliessend in dem Fall aus der Menge der zu prüfenden Titel entfernt, wenn deren Kategorieninhalte in den untergeordneten eingefügt werden konnten.
Die Anzahl der Kategorien, die zur weiteren Prüfung auf Dubletten herangezogen wurden, entsprechen jenen, die in Jele (2009) genannt sind:
zur Bildung des Personennamens wurden die Inhalte aus den MAB2-Kategorien `100`, `100b`, `100c`, `100f` und `359` herangezogen; für den Körperschaftsnamen die Kategorien `200`, `200b` und `200c`; für die Ausgabebezeichnung die Kategorie `403`; für die Erscheinungsorte `410` und `410a`; für die Verleger die Kategorien `412` und `412a`; für das Erscheinungsjahr `425a`, `425b` und `425c`; für die Umfangsangabe die Kategorien `433`, `433a` und `433b` sowie für die ISBN `540a`, `540b` und `540`.

Die so zusammengeführten Titeldaten wurden auf einem einfachen handelsüblichen Desktop-Computer in eine `MySQL`-Datenbank geladen. Als Betriebssystem kam ein `Ubuntu Linux` in der Version `11.04` zum Einsatz. Alle Abfrage- und Verrechnungsmethoden wurden in der Programmiersprache `Perl` realisiert. Die Auswertung der Ergebnisse erfolgte unter Einsatz der freien Statistiksoftware `R` sowie dem dazu installierten Paket `ggplot2` (siehe dazu auch Wickham (2009)). Für den Umgang mit bzw. die Darstellung von großen Datenmengen wurde auf die Empfehlungen in Unwin; Theus & Hofmann (2006) geachtet.

Zu berücksichtigen ist dabei, dass die hier angeführten Zahlen nicht als Absolutwerte, sondern ausschließlich in ihrer Relation zu verstehen sind. Die Ausstattungsmerkmale handelsüblicher Desktop-Computer ändern sich im Wesentlichen alle paar Monate, sodass davon auszugehen ist, dass kurze Zeit nach dem Erscheinen dieses Beitrags mit der jeweils aktuellen Hardware deutlich bessere Ergebnisse (also im Wesentlichen kürzere Berechnungszeiten) erzielt werden können. Das Verhältnis zwischen den Angaben sollte jedoch auch unter diesen Umständen Gültigkeit behalten.

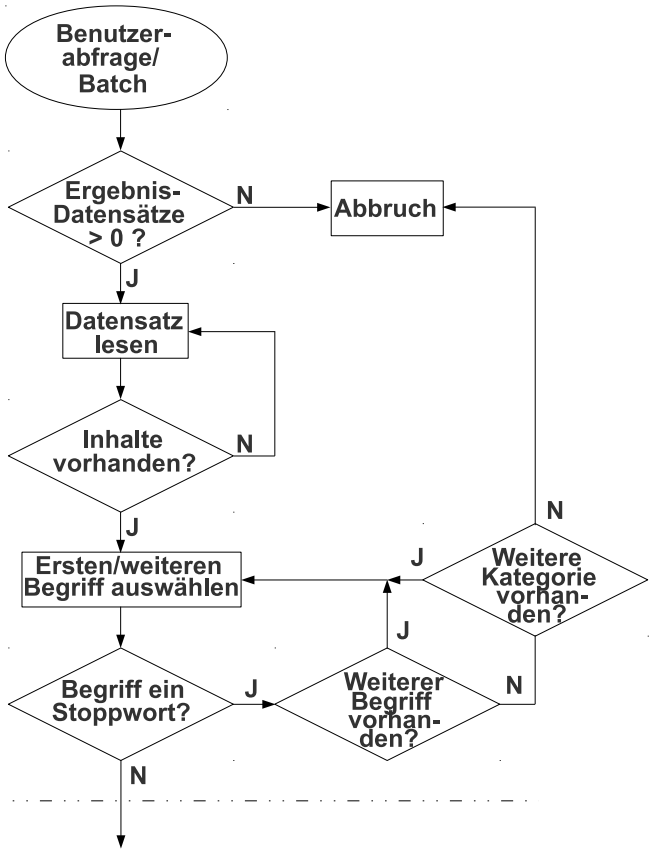

Abbildung 2: Flussdiagramm zur Logik der Begriffsfindung. Die waagrechte Linie deutet den Übergang zu den nachfolgenden Routinen an.

Gemessen wurden im Wesentlichen bei allen drei Ansätzen:

- die Zeitdauer, die für die Ermittlung der potentiellen Titeldubletten benötigt wird,
- die Zeitdauer, die verstreicht, bis der Anfragebegriff ermittelt ist,
- die Menge an Ergebnisdatensätzen, die durch die Bildung der jeweiligen Anfragen entstehen,
- die Anzahl der Kategorien, die im jeweiligen Titeldatensatz mit Inhalten versehen waren,
- sowie die Anzahl der Kategorien, die zur Findung des Anfragebegriffs analysiert werden mussten.

Geprüft wurde die erzielte Menge an Titeldaten gegen sich selbst. Bezogen auf die obige Abbildung 1 bedeutet dies, dass in diesem Fall {`m1`} gleich {`m2`} ist.

In Abbildung 2 ist die einleitende Analyse eines Datensatzes durch ein Flussdiagramm dargestellt:
Nachdem geprüft wird, ob durch die Abfrage ein Ergebnis (eine Menge {`m1`}) erzielt wurde, wird daraus der erste Datensatz gelesen. Dieser wird im Weiteren auf Inhalte in den entsprechenden Kategorien geprüft. Sind Inhalte vorhanden, so wird aus der ersten Kategorie, die Inhalte aufweist (Personen- , Körperschaftname oder Einträge zum Titel des Werkes) der erste Begriff

entnommen. Sofern dieser kein Stoppwort darstellt, wird er für die weitere Verarbeitung verwendet. Ob dieser jedoch auch für die Bildung des Anfragebegriffs verwendet wird, hängt vom entsprechend gewählten Ansatz ab. Für den Fall, dass der jeweilige Begriff indessen ein Stoppwort darstellt, wird sukzessive der jeweils nächste zur Prüfung herangezogen.
Ist letztlich kein einziger Begriff vorhanden, der sich aufgrund dieser Logik als Treffer qualifiziert, wird der Datensatz ignoriert und, sofern ein weiterer vorhanden ist, der nächste Titeldatensatz eingelesen und dieser Prüfung unterzogen.

In allen Ansätzen wurden die Kategorien in folgender Reihenfolge verwendet:
1. Personenname,
2. Körperschaftsname und
3. Einträge zum Titel des Werkes.

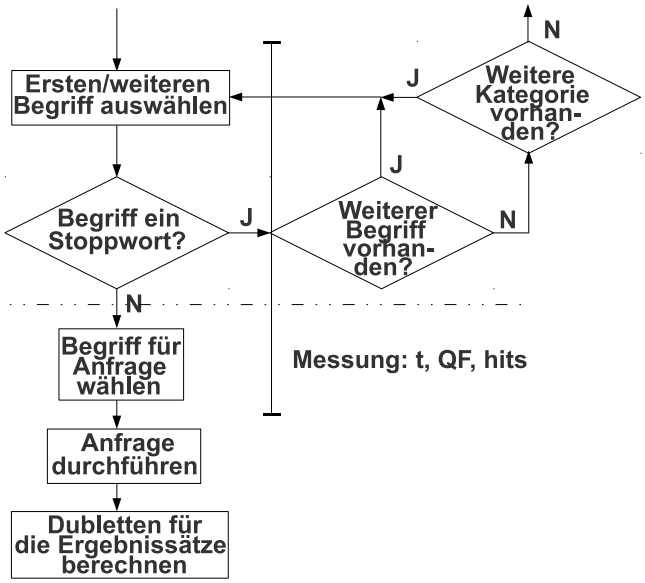

Abbildung 3: Flussdiagramm zum ersten Ansatz: Der erste Begiff, der kein Stoppwort ist, wird für die Bildung des Anfragebegriffs verwendet.

4.1 Der erste Ansatz

Die Entscheidung, den ersten Begriff, der kein Stoppwort ist, für die Bildung des Anfragebegriffs aus den heranzuziehenden Kategorien zu verwenden bringt die Vorteile mit sich, dass dieser Ansatz

- einfach und schnell zu implementieren ist, da keine besonders aufwändigen oder gar diffizilen Prüfungen im Datensatz durchzuführen sind
- und in der Regel kurze Reaktionszeiten des Systems mit sich bringt.

Für die Prüfung von kleineren Datenmengen (< 10.000 Datensätze) sowie in Systemumgebungen, in denen keine großen Mengen zu deduplizieren sind und keine sehr großen Treffermengen erwartet werden, ist dieser Ansatz sicher ein passender.
Zu beachten bleibt der Umstand, dass eine Datenbankabfrage in der Regel wesentlich zeitintensiver ist als die anschließende Berechnung von Dubletten. Wenn also wenig Zeitaufwand in das Finden eines passenden Anfragebegriffs gesteckt wird, lassen sich im Normalfall auch die sehr kurzen Reaktionszeiten eines solchen Systems beobachten.

Unter den oben angeführten Umständen, also bezogen auf die Datenmenge einer durchschnittlichen österreichischen Universitätsbibliothek, zeigen sich jedoch auch rasch die Nachteile dieses Ansatzes bei Umsetzung und Inbetriebnahme:

- die entstehende Menge an Ergebnisdatensätzen kann mitunter sehr hoch sein, da jeder Begriff für die Anfrage verwendet wird, solange er kein Stoppwort darstellt und es im Durchschnitt so zu sein scheint, dass viele Begriffe existieren, die schnell zu mehr als 1.000 Treffern führen.

Dieser Umstand mag für ein System, das die Anzeigeergebnisse von Online-Abfragen bearbeitet weniger relevant sein als für die Prüfung von Dubletten im Batchverfahren: Die erwartbar hohe Menge an Ergebnisdatensätzen muss im Anschluss an die Berechnung der Titeldubletten noch analysiert werden. Das heißt, dass am Ende eines solchen Vorgangs damit zu rechnen ist, dass eine besonders große Menge an Daten zustande kommt, deren Verarbeitung sich im Weiteren auch sehr aufwändig gestalten kann.

In Abbildung 3 ist die Logik dieses sehr einfach gehaltenen Ansatzes in einem Flussdiagramm wiedergegeben: Sobald ein erster Begriff, der kein Stoppwort darstellt, aufgefunden ist, wird er zur Bildung der Anfrage herangezogen und diese durchgeführt. In der Abbildung sind zudem die beiden Punkte innerhalb der Logik vermerkt, die für die Detailmessung (hier: die Zeitmessung pro Datensatz; siehe dazu auch Abbildung 4) herangezogen wurden.

Zu prüfen waren in weiterer Folge die konkreten Ausprägungen der oben angeführten Vor- und Nachteile. Es wurde daher eine Teilmenge von 10.000 Titeldatensätzen entnommen und diese gegen die Gesamtmenge geprüft.
Die sich dabei ergebende Gesamtdauer war $6.245{,}762\,\mathrm{s}$ ($\approx 1\,\mathrm{h}\ 45\,\mathrm{min}$), wobei diese Zeitdauer bereits die Berechnung der Dubletten beinhaltet. Die Anzahl der Ergebnisdatensätze war $1.049.057$.

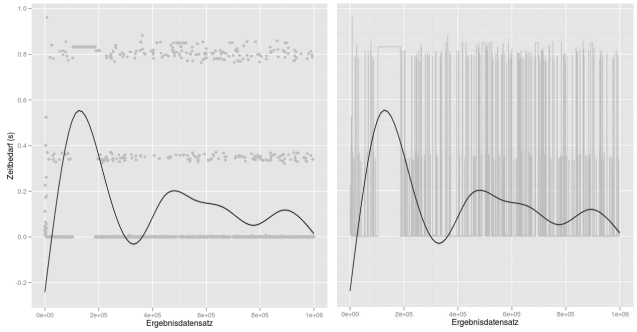

Abbildung 4: Ansatz 1: Messung der Zeitdauer zur Ermittlung des Anfragebegriffs pro Datensatz im Batchverfahren. Die Anfrage wird durch den ersten Begriff gebildet, der kein Stoppwort darstellt. Links die Darstellung der Werte in einem Streudiagramm, rechts als Liniendiagramm.

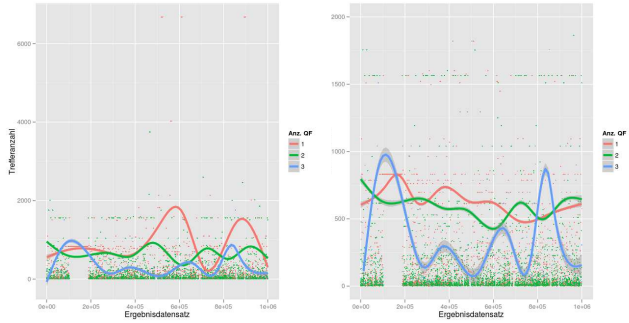

Abbildung 5: Ansatz 1: Die Anzahl der entstehenden Ergebnisdatensätze pro Anfrage.
`{m1} = 10.000`, `{m2} = 400.000`. `QF` = Anzahl der für die Bildung des Anfragebegriffs vorhandenen Kategorien mit verwertbaren Einträgen (max = `3`, min = `1`).

Abbildung 4 zeigt den Verlauf des Zeitbedarfs pro Ergebnisdatensatz. Aus dem Streudiagramm erkennt man deutlich, dass die ermittelten Werte zwischen Maximal- und Minimalausprägungen nicht kontinuierlich verteilt sind. Vielmehr ist eine deutliche Häufung um drei Werte zu beobachten: `0 s`, `0.35 s` und `0.8 s`. Diese Werte ergeben sich aus dem Umstand, dass der Anfragebegriff aus den Einträgen gebildet wird, die in einer von drei Kategorien vorkommen.
Kann der Begriff aus dem Personennamen (= die erste Kategorie, die herangezogen wird) gebildet werden, so wird dieser für die Anfrage verwendet. Die Messung erzielt dabei einen Wert um null Sekunden. Wird entsprechend der Logik aus Abbildung 3 zudem gleich der erste Begriff aus dem Personennamen (das ist zumeist der Familienname) für eine Anfrage herangezogen, so ist der sich daraus ergebende, tatsächliche Zeitaufwand nahe der Messgenauigkeit angesiedelt (gemessen wurde „hochauflösend“ auf drei Nachkommastellen). Ansonsten ist der sich ergebende Wert typischerweise um `0.03 s` angesiedelt.
Führen die Inhalte aus den Kategorien zum Personennamen zu keinem Anfragebegriff, so werden die Einträge zu den Körperschaftsnamen entsprechend geprüft. Dieser neuerliche Durchlauf bringt, wenn man die Messergebnisse betrachtet, jedoch keinen kontinuierlichen Anstieg im Zeitaufwand mit sich. Obwohl der Datensatz bereits vollständig im Arbeitsspeicher abgelegt ist und keine neuerliche Datenbankabfrage getätigt werden muss, nimmt der Aufwand vielmehr sprunghaft zu. Durch die neuerliche Abfolge der Logik zur Ermittlung eines passenden Abfragebegriffs kommen jene Messwerte zustande, die in Abbildung 4 zu einer Häufung der Ergebnisse um den Wert `0.35 s` führen.
Der gleiche Effekt, nämlich die nicht kontinuierliche Steigerung im Zeitbedarf, wird erzielt, wenn auch aus den Einträgen zum Körperschaftsnamen kein Anfragebegriff gebildet werden kann. Dann nämlich wird versucht, aus den Inhalten zum Titel einen entsprechenden Begriff zu bilden. Der Aufwand, der durch diesen letzten Programmschritt verursacht wird, führt zu einer Verdoppelung von `0.35 s` ($\approx$`0.4 s`) auf `0.8 s`.

Interessant ist in diesem Zusammenhang auch die Auswertung zu sehen, die den zeitlichen Aufwand pro Titeldatensatz bei fortschreitender sequentieller Abarbeitung im Batchlauf betrachtet: Anfragen, die ein- oder mehrmals aufgrund ein und desselben Begriffs zu einer Datenbankabfrage geführt haben, bleiben im Cache der Datenbank erhalten. Dieser Umstand bringt mit sich, dass die selbe Anfrage zu einem späteren Zeitpunkt in deutlich kürzeren Antwortzeiten erledigt werden kann. Dies wird in Abbildung 4 jedoch erst erkennbar, wenn die einzelnen Messwerten durch die Ausgabe einer passenden Glättungsfunktion überlagert werden.

Für die einzelnen Messreihen, die zum Vergleich gegenübergestellt werden, bedeutet dies, dass vor dem Start eines jeden Durchlaufs der Datenbankcache (in diesem Fall der Querycache) gelöscht werden muss. Anderenfalls würden nachfolgende Messungen von vorangegangenen profitieren bzw. zu verfälschten Einzelergebnissen führen. Die Verfälschung der Ergebnisse bezogen auf den Gesamtdurchlauf ist dabei eher gering. Wird ein und derselbe sequentielle Batchdurchlauf wiederholt, so benötigt dieser im zweiten Durchlauf `6.149,775 s` gegenüber `6.245,762 s` aus dem ersten Durchlauf. Die Differenz stellt einen Zeitunterschied von ca. `2 min` bei einer Durchlaufzeit von `1 h 45 min` bei einer Verarbeitung von `10.000` Datensätzen dar. Bei der Verarbeitung der Gesamtmenge von ca. `400.000` Titeldatensätzen wird die Differenz zwischen dem Erst- und dem Zweitdurchlauf geringer, da der maximal verfügbare und sinnvoll einsetzbare Querycache, bezogen auf ein und dieselbe Hardware, nicht beliebig erhöht werden kann.

Neben der Messung der Zeitdauer pro Datensatz, die verstreicht, um einen passenden Anfragebegriff zu ermitteln, ist die Erhebung der Anzahl der durch die entsprechende Anfrage entstehenden Ergebnisdatensätze von Relevanz. Bereits angesprochen wurde dazu der Um-

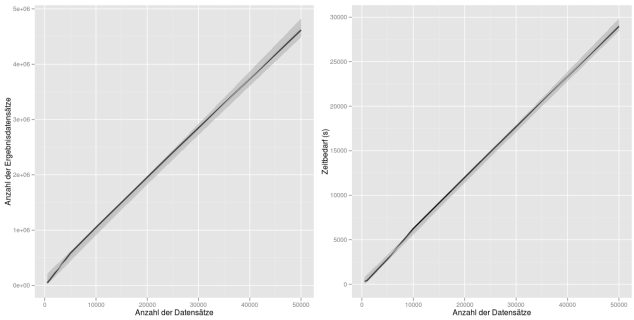

Abbildung 6: Ansatz 1: Die Anzahl der Ergebnisdatensätze sowie der Zeitaufwand zur Dublettenberechnung in Abhängigkeit von der Anzahl der zu prüfenden Datensätze. Die dargestellten Werte entsprechen jenen aus Tabelle 1.

stand, dass durch Anwendung der Methode, den ersten Begriff, der kein Stoppwort ist, für die Anfrage zu verwenden, davon auszugehen ist, dass mitunter eine große Menge an Ergebnisdatensätze entstehen, die in weiterer Folge auf mögliche Dubletten zu prüfen sind. Der Vorteil, dass diese Methode sehr rasch zum Auffinden des Anfragebegriffs führt, würde sich in einem solchen Fall durch den Nachteil einer aufwändigen Dublettenberechnung in einen Nachteil verwandeln.
Die bereits genannte Gesamtzahl an Ergebnisdatensätzen (`1.049.057`), die der sequentielle Durchlauf erzeugt, deutet auf diesen Umstand hin. Da dies im Einzelfall bei der praktischen Anwendung in einem Online-System jedoch nicht auffällig ist, wurde dem im Einzelnen nachgegangen.

Abbildung 5 zeigt links in einem Streudiagramm die einzelnen Werte. Die Messpunkte wurde in Abhängigkeit von der Anzahl der verfügbaren Kategorien, die zur Bildung des Anfragebegriffs herangezogen werden konnten, in unterschiedlichen Farben dargestellt: Blau für das Vorhandensein von drei, Grün für zwei und Rot für eine Kategorie.
Aus dieser Abbildung erkennt man deutlich, dass das Vorhandensein von drei Kategorien auch dann von Vorteil für die Bildung des Anfragebegriffs ist, wenn dieser aus dem ersten Eintrag, der kein Stoppwort darstellt, gebildet wird. Sind hingegen nur zwei oder ist gar nur eine Kategorie vorhanden, so führt dieser Umstand, bezogen auf die Einzelabfrage, zu einer deutlich größeren Treffermenge. Dies wird jedoch erst erkennbar, sobald die Einzelwerte durch eine Glättungsfunktion überlagert werden.
In Abbildung 5 werden auf der rechten Seite nur jene Werte abgebildet, die zu Anfragen führten, deren Ergebnis < 2.000 war. Aus dieser Abbildung erkennt man zudem, dass der überwiegende Anteil an Anfragen zu Ergebnissen führen, die deutlich < 500 sind. Dies mag als ein erster Hinweis auf die Sinnhaftigkeit der Einführung eines Schwellwertes gelten.

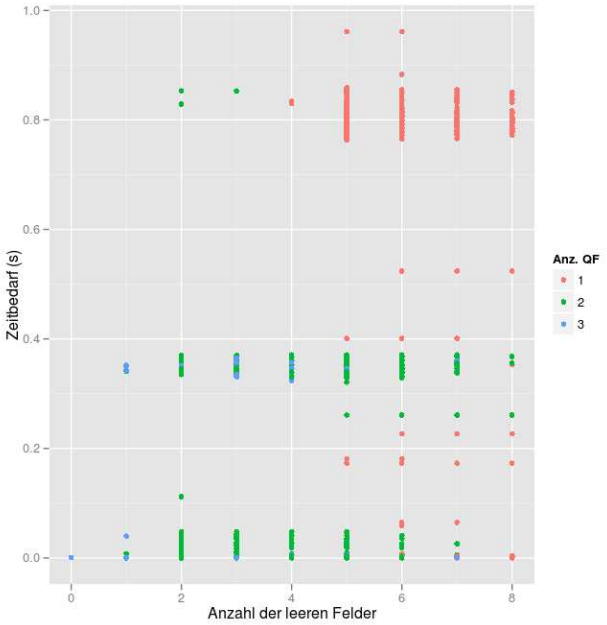

Abbildung 7: Ansatz 1: Der Zeitbedarf zur Ermittlung des Anfragebegriffs pro Datensatz in Abhängigkeit von der Anzahl der leeren Kategorien im Titeldatensatz.
`{m1} = 10.000`, `{m2} = 400.000`. `QF` = Anzahl der für die Bildung des Anfragebegriffs vorhandenen Kategorien mit verwertbaren Einträgen (max = `3`, min = `1`).

Dass die Anzahl jener Kategorien, die für die Bildung des Anfragebegriffs verwendet werden können, ein deutlich wahrnehmbares Maß für den dabei zu berechnenden Zeitaufwand ist, konnte bereits in der Auswertung zur Abbildung 4 gezeigt werden. Der Umstand, dass die Zeitdauer dabei nicht kontinuierlich, sondern vielmehr diskret steigt, zeigt zudem, dass die Anzahl der Begriffe, die eine Kategorie beinhaltet, wesentlich geringere Auswirkungen auf die Bildung des Anfragebegriffs hat als der Umstand, wieviele Kategorien auszuwerten sind.

In Abbildung 7 wurde der Zusammenhang dargestellt, der zwischen dem Zeitbedarf zur Ermittlung des Anfragebegriffs pro Datensatz und der Anzahl der leeren Kategorien im Titeldatensatz besteht. Aus der Darstellung erkennt man, dass von den zehn Kategorien, die letztlich die zusammengeführten Inhalte repräsentieren, bis zu acht ohne Einträge sein können. Für die Auswertung von Interesse ist eigentlich die Anzahl der verwertbaren Kategorien (in Abbildung 7 durch den Faktor `QF` (Queryfields) gekennzeichnet).
Trotzdem zeigt sich hier auch, dass der Aufwand, eine für die Ermittlung des Anfragebegriffs verwertbare Kategorie zu finden, direkt mit der Anzahl der Kategorien zusammenhängt, die leer sind. Die Auswertung der Ergebnisse, die in den bisherigen Abbildungen wiedergegeben wurden, lässt auf diesen Umstand noch nicht schließen. Aus diesen kann nur auf den Zusammenhang zwischen dem Zeitaufwand und der vorhandenen Anzahl der eigentlich verwertbaren Kategorien geschlossen werden. Bei der Verarbeitung der bibliographischen Daten

Anfragesätze	Ergebnissätze	Zeit (s)
500	47881	335.450
1000	103754	418.308
5000	581753	2845.639
10000	1049057	6245.762
25000	2407018	14873.324
50000	4617445	28967.480

Tabelle 1: Ansatz 1: Die Anzahl der Ergebnisdatensätze sowie der Zeitaufwand zur Dublettenberechnung in Abhängigkeit von der Anzahl der zu prüfenden Datensätze.

ist somit zu berücksichtigen, dass der Umstand leerer Kategorien überwiegend jene trifft, die für die weitere Vorgehensweise eigentlich benötigt würden.

Zur Abschätzung der sich ergebenden Treffermengen in Abhängigkeit von der Anzahl der zu prüfenden Titeldaten wurde eine Messreihe zum Ansatz 1 gebildet, deren Werte in Tabelle 1 wiedergegeben sind. Aus den Ergebnissen zeigt sich, dass ein linearer Zusammenhang zwischen der Menge der zu prüfenden Titel in {m1} und der sich daraus ergebenden Menge {m3} besteht: Eine Verdopplung der Menge {m1} führt zu einer Verdopplung der Ergebnismenge {m3}. Der gleiche Umstand gilt für den sich daraus ergebenden Zeitaufwand, der letztlich zur Deduplizierung benötigt wird. Auch dabei führt eine Verdopplung der Menge {m1} zu einer Verdopplung des Zeitaufwands. Dieser Zusammenhang ist in Abbildung 6 dargestellt.

Zusammenfassend lässt sich für den ersten Ansatz sagen, dass eine höhere Anzahl an vorhandenen und zugleich auch verwertbaren Kategorien zu einem geringeren Zeitaufwand zur Ermittlung des Anfragebegriffs führt. Dieser Umstand ist durchaus erwartbar. Unerwartet ist jedoch, dass der Zeitaufwand pro Datensatz dabei eine Größe annimmt, die im Vorfeld der Dublettenermittlung bestimmt werden kann, da sie direkt von der Anzahl der vorhandenen Kategorien abhängt.
Die größte Dichte an Ergebnissen wird aufgrund von Anfragebegriffen erzielt, die eine Treffermenge haben, die < 500 ist. Bei der Verwendung von Schwellwerten (vgl. Ansatz 3) kann dieses Maß bereits als ein möglicherweise sinnvoller Richtwert gesehen werden. Der Nachteil der hohen Ergebnismenge zeigt sich deutlich; wenngleich dieser Umstand erst aus dem nachfolgenden Vergleich mit den Ergebnissen aus den Ansätzen zwei und drei dargestellt werden kann.

4.2 Der zweite Ansatz

Im Batchlauf eines Dublettenabgleichs sind – entgegen der Anwendung zur Deduplizierung von Ergebnisdatensätzen in Online-Systemen – auch beim Abgleich kleinerer Titelmengen (< 10.000 Datensätze) stets große Ergebnismengen zu erwarten. Der das Ergebnis minimierende Ansatz ist, jenen Begriff zur Anfrage zu verwenden, der kein Stoppwort darstellt und dessen Treffermenge pro Anfragedatensatz am kleinsten ist.
Der sich daraus ergebende Nachteil ist, dass der Anfragebegriff in diesem Ansatz nur durch eine aufwändige Prüfung zu ermitteln ist: Alle Begriffe, die kein Stoppwort darstellen, müssen aus 1 - 3 Kategorien einem Datensatz entnommen und die damit jeweils erzielbare Treffermenge gezählt werden. Anschließend muss der Begriff mit der kleinsten Treffermenge gewählt und mit diesem die Anfrage gebildet werden.

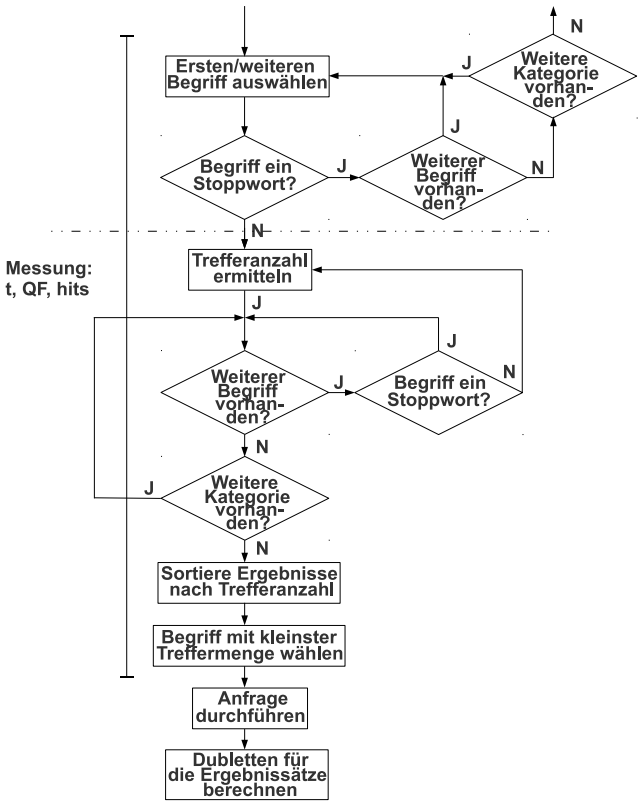

Abbildung 8: Flussdiagramm zum zweiten Ansatz: Jener Begriff wird als Anfragebegriff verwendet, der kein Stoppwort darstellt und dessen Treffermenge am kleinsten ist.

Unter den bereits genannten Rahmenbedingungen, dass eine Teilmenge von 10.000 Titeldatensätzen der Gesamtmenge von 400.000 entnommen und diese auf Dubletten innerhalb der Gesamtmenge geprüft wurden, ergibt sich eine Gesamtlaufzeit von 26.815,551 s ($\approx$ 7 h 27 min). Dabei wurden 210.661 Ergebnisdatensätze produziert. Das entspricht in etwa einer Vervierfachung im Zeitaufwand und gleichzeitig einem Fünftel der Ergebnismenge gegenüber dem Ansatz 1. Es bleibt also im jeweiligen Einsatzfall zu prüfen, ob die Minimierung der Ergebnisse die deutliche Verlangsamung in jedem Fall rechtfertigt bzw. als dringend notwendig erscheinen lässt. Im Fall, dass große Datenmengen auf Dubletten geprüft werden müssen, kann diese Methode sinnvoll erscheinen, da sich die nachfolgende Auswertung entsprechend einfacher gestalten kann.

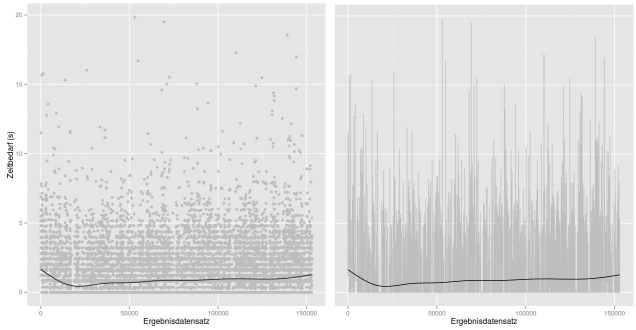

Abbildung 9: Ansatz 2: Messung der Zeitdauer zur Ermittlung des Anfragebegriffs pro Datensatz im Batchverfahren. Die Anfrage wird durch jenen Begriff gebildet, der kein Stoppwort darstellt und der die kleinste Treffermenge hervorruft. Links die Darstellung der Werte in einem Streudiagramm, rechts als Liniendiagramm.

In Abbildung 8 ist die Bildung des Anfragebegriffs in einem Flussdiagramm dargestellt: Den entsprechenden Kategorien werden sukzessive alle Begriffe entnommen, diese darauf geprüft, ob sie ein Stoppwort darstellen und welche Treffermenge mit ihnen erzielt wird. Anschließend wird jener Begriff gewählt, der kein Stoppwort ist und der bei einer Anfrage die kleinste Treffermenge hervorruft. Pro Datensatz muss diese Logik für eine Vielzahl an Begriffen (nämlich allen, die in den zur Bildung herangezogenen Kategorien enthalten sind) durchlaufen werden. Zudem beinhaltet jeder Durchlauf zwei Datenbankabfragen (Klärung von Stoppwort und Treffermenge). Daher muss damit gerechnet werden, dass das Auftreten von großen Begriffsmengen auch zu einem großen zeitlichen Aufwand führt.

Bei der Betrachtung der Einzelergebnisse, die in Abbildung 9 dargestellt sind, erkennt man, dass der Wertebereich zwischen `0` und `5 s` am dichtesten belegt ist und dass „gar nicht wenige“ Einzelergebnisse oberhalb dieses Bereichs angesiedelt sind. Was sich am Fazit des Batchlaufs jedoch zeigt ist, dass von den Einzelergebnissen, im Gegensatz zum ersten Ansatz, nicht auf das Gesamtergebnis geschlossen werden kann. Die Anzahl der Einzelwerte, die in Abbildung 9 oberhalb von `5 s` ihren Eintrag haben, ist beachtlich. Letztlich führt die Methode, den Begriff mit der kleinsten Treffermenge für eine Abfrage zu verwenden, jedoch bloß zu einer Vervierfachung im Zeitaufwand.

Ob diese Methode sinnvoll für eine Integration in ein Online-System verwendet werden kann, muss im Einzelfall entschieden werden. Zu berücksichtigen ist dabei auf jeden Fall der Umstand, dass gerade Suchmaschinenbasierte Online-Systeme dahingehend optimiert werden, mit möglichst kurzen Antwortzeiten zu reagieren. Unter diesen Umständen mag es daher eigentümlich anmuten, die möglichst kurzgehaltene Antwortzeit durch dieses Verfahren, das die erzielbaren Ergebnismengen optimiert, wieder zu verlängern.

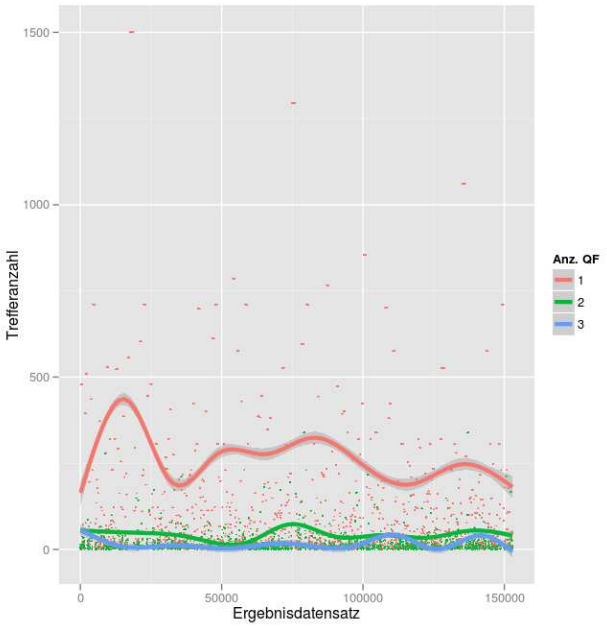

Abbildung 10: Ansatz 2: Die Anzahl der entstehenden Ergebnisdatensätze pro Anfrage.
`{m1} = 10.000`, `{m2} = 400.000`. `QF` = Anzahl der für die Bildung des Anfragebegriffs vorhandenen Kategorien mit verwertbaren Einträgen (max = `3`, min = `1`).

Entgegen den Ergebnissen aus Ansatz 1 zeigt die Darstellung der Treffermengen pro Ergebnisdatensatz im Ansatz 2, dass die Anzahl der Treffer deutlich minimiert werden kann. Ist die größte Dichte an Treffern in Abbildung 5 (entsprechend dem Ansatz 1) zwischen `0` und `500` ausfindig gemacht werden, so liegt diese hier deutlich darunter im Wertebereich von `0` bis `50`. Abbildung 10 zeigt, dass für jenen Fall, in dem `2` oder alle `3` Kategorien verwertbare Inhalte tragen, die Trefferanzahl pro Anfrage sehr gering (mitunter auch < 10) gehalten werden kann. Im Gesamtergebnis drückt sich dieser Umstand deutlich aus: Die Gesamtzahl an Ergebnisdatensätzen schrumpft auf ein Fünftel gegenüber dem Ansatz 1.

Für den Einzelfall zeigt sich in Abbildung 10 zudem der (auch erwartbare) Umstand, dass das Vorhandensein weniger Begriffe zur Ermittlung des Anfragebegriffs auch den Umstand mit sich bringt, dass weniger Begriffe vorhanden sind, die zu kleinen Treffermengen führen. Entgegen den erreichbaren Treffermengen mit < 50 Ergebnisdatensätzen sind im Fall nur einer verwertbaren Kategorie zur Bildung des Anfragebegriffs Werte üblich, die in Abbildung 10 um die Anzahl von `250` schwanken. Erkennbar wird dies wiederum durch die Überlagerung der Einzelergebnisse im Streudiagramm durch eine passende Glättungsfunktion.

Der Umstand, dass eine geringe Anzahl an verwertbaren Kategorien zu höheren Treffermengen pro Anfrage führt zeigt sich auch darin, dass diese letztlich zu einem gesteigerten Zeitaufwand führen. In Abbildung 11 ist der Zusammenhang zwischen der Anzahl der in den

einzelnen Datensätzen vorhandenen, leeren Datenfeldern und dem Zeitbedarf zur Ermittlung des Anfragebegriffs in Abhängigkeit von der Anzahl der nicht verwertbaren Kategorien dargestellt. Daraus erkennt man im Streudiagramm aufgrund der farblichen Kennzeichnung deutlich, dass insgesamt mehr leere Felder (erwartbar) auch zu weniger verwertbaren Kategorien und letztlich zu einem geringen Zeitaufwand in der Bestimmung des Abfragebegriffs führt. Umgekehrt beschrieben bringt eine größere Menge an verwertbaren Kategorien einen größeren Zeitaufwand, letztlich jedoch eine, wie bereits beschrieben, deutlich geringere Anzahl an Ergebnisdatensätzen mit sich. Das heißt für den Einzelfall, dass der Umstand geringer Ergebnisdatensätze nur durch größeren Zeitaufwand erzielt werden kann. Im Gesamtergebnis schlägt sich dies jedoch nicht so deutlich nieder, da eine kleinere Treffermenge in Folge auch einen geringeren Zeitaufwand für die Berechnung und Auswertung der Titeldubletten bewirkt.

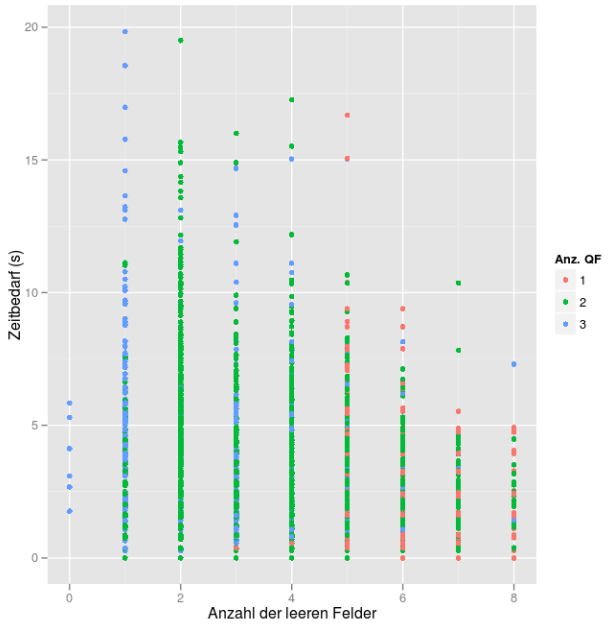

Abbildung 11: Ansatz 2: Der Zeitbedarf zur Ermittlung des Anfragebegriffs pro Datensatz in Abhängigkeit von der Anzahl der leeren Kategorien im Titeldatensatz.
`{m1} = 10.000`, `{m2} = 400.000`. `QF` = Anzahl der für die Bildung des Anfragebegriffs vorhandenen Kategorien mit verwertbaren Einträgen (max = 3, min = 1).

Zusammenfassend lässt sich für den zweiten Ansatz sagen, dass die Zeitdauer pro Einzeldatensatz durchaus erheblich sein kann, sodass dieser im Batchlauf wohl eher nur dann zur Anwendung gelangt, wenn die Anzahl der Ergebnisdatensätze möglichst klein sein muss.
Ob dieser Ansatz zudem in einem Online-System zu Einsatz kommen kann, hängt von der gesamten, „gefühlten“ Zeitdauer ab, die ein möglicherweise auf kurze Antwortzeiten optimiertes System zusätzlich erfährt. Im Einzelfall mag der Ansatz 2 als deutlich störend empfunden werden, das Gesamtverhalten gegenüber einem Benutzer kann nur durch Beobachtung eingeschätzt werden.

4.3 Der dritte Ansatz

Die Vor- und Nachteile, die sich aus den beiden, bereits beschriebenen Ansätzen ergeben, können durch einen Quasi-Kompromiss günstig zu einem dritten Ansatz ergänzt werden: Der Vorteil, rasch zu einem Anfragebegriff zu gelangen, ohne in jedem Fall alle verwertbaren Einträge zu prüfen (= Zeitoptimierung), kann mit dem Vorteil kombiniert werden, dass nicht einjeder Begriff zur Auswahl gelangt und so die Treffermenge eher klein bleibt (= Optimierung der Menge). Führt man einen Schwellwert für die höchst zulässige Treffermenge pro Anfragebegriff ein, so kann der erste Begriff, dessen Ergebnismenge unter dieser Schwelle bleibt, verwendet werden – ohne, dass weitere, oder gar sämtliche, Begriffe geprüft werden müssen. Ein solcher Ansatz verspricht relativ schnelle Antwortzeiten (nahe denen aus Ansatz 1) bei einer gleichzeitig geringen Gesamtmenge an Treffern (im günstigen Fall nahe denen aus Ansatz 2). Im Ansatz 3 wird also der erste Begriff, dessen Treffermenge unter einem bestimmten Schwellwert liegt und der kein Stoppwort ist, als Anfragebegriff für die nachfolgende Deduplizierung verwendet.

Die hohe Dichte an Ergebniswerten, die in Abbildung 5 im Bereich von `0` bis `500` erkennbar ist, ließ bereits einen möglichen Schwellwert erahnen, der unter `500` liegen kann. `100` wird als ein günstiger Schwellwert z. B. in Lohrum u. a. (1999, S. 3) angegeben. Als Begründung wird genannt, dass diese Treffermenge sehr rasch auf Dubletten geprüft werden kann und größere Treffermengen deutlich zeitintensiver zu verarbeiten sind. Zieht man jedoch die Ergebnisse heran, die sich aus den entsprechenden Messungen und deren Darstellung in Abbildung 4 zeigen, so erkennt man, dass eine möglicherweise notwendige, weitere Datenbankanfrage zur Ermittlung des Anfragebegriffs wesentlich zeitaufwändiger ist als die Deduplizierung einer größeren Ergebnismenge.

Die Beschäftigung mit dem dritten Ansatz soll zeigen, ob sich empirisch ein besonders günstiger Schwellwert ermitteln lässt und welche weiteren Implikationen die Einführung eines solchen hat.

Abbildung 12 zeigt die zur Ermittlung des Anfragebegriffs heranzuziehende Logik: Aus der ersten verwertbaren Kategorie wird der erste Begriff, der kein Stoppwort darstellt, entnommen und dessen Treffermenge in einer Anfrage ermittelt. Ist die Menge kleiner als durch den Schwellwert definiert, so ist der entsprechende Begriff der Anfragebegriff und die gerade erzielte Treffermenge die passende Ergebnismenge, die im Weiteren dedupliziert wird. Liegt die Treffermenge je-

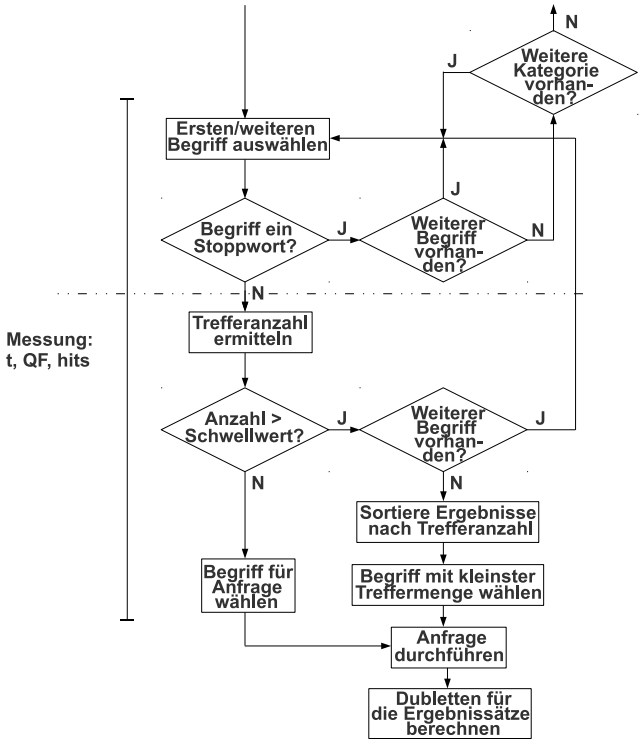

Abbildung 12: Flussdiagramm zum dritten Ansatz: Jener Begriff wird als Anfragebegriff verwendet, der kein Stoppwort darstellt und dessen Treffermenge unter einem definierten Schwellwert liegt.

doch über dem Schwellwert oder entspricht sie diesem, so wird nach einem weiteren Begriff gesucht.
Kann kein Anfragebegriff ermittelt werden, dessen Treffermenge unter dem Schwellwert liegt, so wird der Begriff mit der kleinsten Treffermenge zur weiteren Anfrage verwendet. Je kleiner die Schwelle definiert wird, desto häufiger wird dieser Fall eintreten. Das heißt, dass eine Minimierung der Schwelle die Ergebnisse dieses Verfahrens jenen des Ansatzes 2 und eine zunehmende Vergrößerung die Ergebnisse jenen des Ansatzes 1 annähern wird.

Um einen Überblick über das Verhalten eines solchen Verfahrens zu gewinnen, wurden dreizehn unterschiedliche Schwellwerte im Bereich von `2` bis `1.000` definiert. Anschließend wurde in einem Batchlauf wiederum die Menge von `10.000` Titeldatensätze gegen die Gesamtmenge von `400.000` auf Dubletten geprüft. Gemessen wurde jeweils die Gesamtlaufzeit. Die Auswertung zur Messung der Einzelergebnisse (das sind: die Anzahl der einzelnen Treffer pro Anfragebegriff, der Zeitaufwand für die Ermittlung des Begriffs sowie die Anzahl der vorhandenen, verwertbaren Kategorien) brachte gegenüber den angeführten aus Ansatz 1 und Ansatz 2 keine neuen Erkenntnisse, sodass diese hier im Einzelnen, also für jeden Schwellwert separat, nicht angeführt sind.

In Tabelle 2 sind die Messwerte für den Zeitaufwand sowie die Größe der Gesamttreffermengen aufgelistet, die

Schwelle	Zeit (s)	Menge
2	26815.551	210661
25	9022.699	238164
50	7696.171	262276
100	6620.200	305345
200	6020.932	376623
300	5723.736	466029
400	5653.442	509084
500	5880.583	692117
600	5924.068	720809
700	5921.980	790049
800	5953.812	825637
900	6183.725	952801
1000	6217.125	958996

Tabelle 2: Ansatz 3: Höhe der Schwellwerte, gemessener Zeitaufwand und Größe der Gesamttreffermenge.

mit den entsprechenden Schwellwerten im Bereich von `2` bis `1.000` ermittelt wurden.

In Abbildung 13 wird der Zusammenhang durch eine graphische Darstellung verdeutlicht:
Unterschreitet der Schwellwert die Größe von `200`, so steigt der Zeitaufwand, der zur Ermittlung des Anfragebegriffs benötigt wird, zuerst kontinuierlich (siehe Abbildung 13 links). Ab dem Unterschreiten des Wertes `100` hingegen steigt der Aufwand sehr steil (fast exponentiell) an.
Wird der Schwellwert hingegen Schrittweise vom Ausgangswert (hier: `200`) um `100` vergrößert, so zeigt sich zuerst eine leichte Abnahme im Zeitaufwand, die ab dem Schwellwert von `400` wieder langsam ansteigt.

Begründet kann dieses Gesamtverhalten damit werden, dass eine Minimierung der Schwelle dazu führt, dass letztlich alle Begriffe aus den heranzuziehenden Kategorien darauf geprüft werden müssen, ob sie bei einer Anfrage zu einer Treffermenge führen, die unter dem Schwellwert liegt. Der Zeitaufwand für diese Prüfung scheint deutlich größer zu sein, als jene Ersparnis, die eintritt, wenn in weiterer Folge sehr kleine Treffermengen dedupliziert werden müssen. Das leichte und kontinuierliche Ansteigen des Zeitaufwands ab einem Schwellwert von `400` ist darauf zurückzuführen, dass ab diesem Wert der Zeitaufwand zur Ermittlung des Anfragebegriffs eher stagniert, zugleich aber die erzielbare Treffermenge, die dedupliziert werden muss, kontinuierlich ansteigt (siehe Abbildung 13 rechts).
Dieses Verhalten deckt sich in ihren Auswirkungen mit jener Annahme, die aus dem Streudiagramm der Abbildung 5 abgeleitet wurde: Da schon der erste Begriff, der für eine Anfrage verwendet werden kann, in den allermeisten Titeldatensätzen zu einer kleineren Treffermenge als `500` führt, sind auch „oberhalb“ des Schwellwerts von `400` kaum noch Datensätze anzutreffen, deren Ergebnis zu einer Prüfung eines weiteren Begriffs führt.

Abbildung 13 (rechts) zeigt den fast linearen Zusammenhang zwischen der Größe des Schwellwerts und der

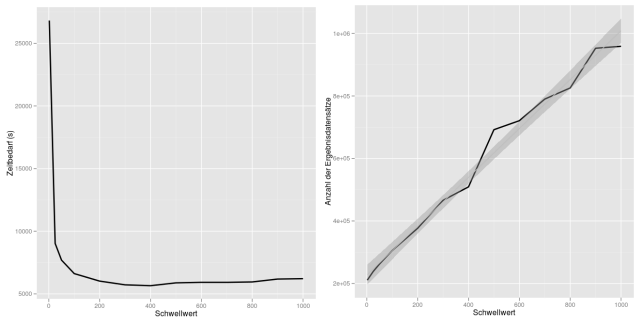

Abbildung 13: Ansatz 3: Der gesamte Zeitbedarf in Abhängigkeit vom Schwellwert zur Ermittlung bibliographischer Dubletten (= Darstellung links).
Die Anzahl der entstehenden Ergebnisdatensätze in Abhängigkeit vom Schwellwert zur Ermittlung bibliographischer Dubletten (= Darstellung rechts).
{m1} = 10.000, {m2} = 400.000.

Menge an Ergebnisdatensätzen. Spielt die Menge an Ergebnisdatensätzen in weiterer Folge eine große Rolle (soll diese Menge etwa optimiert werden), so kann ein Wert zwischen 100 und 200 durchaus als ein sinnvoller angesehen werden. Dies mag im Fall einer Deduplizierung großer Datenmengen im Batchlauf der Fall sein. Zu beachten ist dabei der Umstand, dass allein die Erhöhung des Schwellwerts von 100 auf 200 bereits zu einer Vergrößerung der Gesamttreffermenge um 24 % führt, obwohl der damit verbundene Zeitaufwand gleich bleibt.
Im Fall einer Deduplizierung von Abfrageergebnissen in einem Online-System hingegen wird mit einem Schwellwert von 400 im Durchschnitt eine schnellere Antwortzeit erreicht, wenn davon ausgegangen werden kann, dass die Deduplizierung durch Rechenleistung wesentlich schneller erfolgt, als weitere notwendige Datenbankanfragen zur Ermittlung des passenden Anfragebegriffs.

5 Zusammenfassung

Die im Text angeführten Vor- und Nachteile der einzelnen Ansätze konnten im Wesentlichen bestätigt werden.

So zeigt sich beim ersten Ansatz sehr deutlich der Vorteil, dass die Wahl des Anfragebegriffs durch den ersten Begriff, der kein Stoppwort darstellt, durchaus eine Methode darstellt, die für die Integration in ein Online-System geeignet ist. Vor allem ist dieser Ansatz gerechtfertigt, wenn das betreffende Online-System auf sehr kurze Antwortzeiten optimiert wurde. Für eine Deduplizierung im Batchverfahren nimmt hingegen der Nachteil dieses Ansatzes deutlich Überhand: Die großen Ergebnismengen, die durch den Anfragebegriff hervorgerufen werden, erzeugen einen nicht unerheblichen Aufwand in der nachfolgenden Auswertung.

Für das Batchverfahren hingegen sehr gut geeignet ist der Ansatz 2 immer dann, wenn kein großer Zeitdruck auf dem Verfahren lastet bzw. wenn die zu deduplizierende Menge an bibliographischen Titeln nicht sehr groß ist. Durch die Ermittlung jenes Anfragebegriffs, der die geringste Treffermenge erzielt, entsteht zwar ein deutlich größerer Zeitaufwand (etwa eine Vervierfachung gegenüber dem ersten Ansatz), jedoch wird gleichzeitig die Treffermenge auf ein Fünftel reduziert. Für die Integration in ein Online-System mag die Implementierung dieses Ansatzes eventuell eine ungewollte Verlangsamung des Gesamtsystems darstellen.

Der dritte Ansatz, der quasi einen Kompromiss zwischen den beiden zuvor genannten darstellt, kann sowohl für ein Batchverfahren als auch für die Integration in ein Online-System interessant sein. Bei der Wahl eines günstigen Schwellwerts muss darauf geachtet werden, dass entweder kurze Antwortzeiten (durch einen größeren Schwellwert) oder geringe Treffermengen (durch einen kleineren Schwellwert) erzielt werden. Interessant ist in diesem Zusammenhang jedenfalls der Umstand, der in Abbildung 13 links dargestellt wird: Wird der Schwellwert sukzessive erhöht, so erkennt man zwischen den Werten 200 und 400 zuerst einen leichten Abfall im zeitlichen Aufwand und anschließend einen ebenso leichten, sehr kontinuierlichen Anstieg.
Dies ist auf den Umstand zurückzuführen, dass in typischen Datenumgebungen, wie sie hier verwendet wurden, die allermeisten Begriffe, die für eine Anfrage verwendet werden können, deutlich weniger als 400 Treffer hervorrufen (siehe Abbildung 5). Damit fällt ab diesem Schwellwert zugleich auch der Aufwand weg, einen Begriff durch Mehrfachabfrage erst ermitteln zu müssen: der zuerst oder im zweiten Anlauf aufgefundene Begriff erfüllt damit schon die Bedingungen und kann verwendet werden.
Für den Einsatz im vorliegenden Messverfahren hat sich 200 als ein passender Wert sowohl für die Integration in ein Online-System als auch für das Batch-Verfahren gezeigt.
Die Verringerung des Schwellwerts auf < 100 hingegen hat dazu geführt, dass der eigentliche Vorteil, geringe Treffermengen zu erzielen, insgesamt verloren ging und zum Nachteil wurde, da der zeitliche Aufwand, einen solchen Anfragebegriff zu ermitteln, rapide anstieg.
Offen geblieben ist die Antwort auf die Frage, ob es sinnvoll ist, die Reihenfolge der zu durchsuchenden Kategorien veränderbar zu halten, oder ob die gewählte Reihenfolge (Personenname, Körperschaftsname, Titeleinträge zum Werk) eine unter bestimmten Umständen sogar ungünstige darstellen kann.

6 Literaturverzeichnis

Ananthakrishna, Rohit; Chaudhuri, Surajit & Ganti, Venkatesh (2002): Eliminating fuzzy duplicates in data warehouses. In: Proceedings of the 28th international conference on Very Large Data Bases, VLDB '02. S. 586 - 597.

Baxter, Rohan; Christen, Peter & Churches, Tim (2003): A Comparison of Fast Blocking Methods for Record Linkage. In: SIGKDD Workshop on Data Cleaning, Record Linkage and Object Consolidation, Washington DC.

Bilenko, Mikhail; Kamath, Beena & Mooney, Raymond (2006): Adaptive Blocking: Learning to Scale Up Record Linkage. In: Sixth International Conference on Data Mining. ICDM '06. S. 87 - 96.
Online unter:
http://dx.doi.org/10.1109/ICDM.2006.13

Draisbach, Uwe & Naumann, Felix (2009): A comparison and generalization of blocking and windowing algorithms for duplicate dedection. In: Proceedings of the International Workshop on Quality in Databases, S. 51 -56.
Online unter:
http://www.hpi.uni-potsdam.de/fileadmin/hpi/FG_Naumann/publications/2009/QDB09_crc.pdf

Draisbach, Uwe & Naumann, Felix (2010): DuDe: The Duplicate Detection Toolkit. In: 8th International Workshop on Quality in Databases, QDB'10. Singapore.
Online unter:
http://www.vldb2010.org/proceedings/files/vldb_2010_workshop/QDB_2010/Paper5_Draisbach_Naumann.pdf

Hernández, Mauricio & Stolfo, Salvatore (1995): The merge/purge problem for large databases. In: SIGMOD '95 Proceedings of the 1995 ACM SIGMOD international conference on Management of data, S. 127- 138.
Online unter:
http://dx.doi.org/10.1145/568271.223807

Hernández, Mauricio & Stolfo, Salvatore (1998): Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. In: Data Mining and Knowledge Discovery. Vol. 2 (Issue 1), S. 9 -37.
Online unter:
http://dx.doi.org/10.1023/A:1009761603038

Jele, Harald (2009): Erkennung bibliographischer Dubletten mittels Trigrammen: Messungen zur Performanz. In: BIT-Online (Heft 3, 2009), S. 265 - 272 (= Teil 1) sowie BIT-Online (Heft 4, 2009), S. 385 - 390 (= Teil 2).
Pre-Print online unter:
http://wwwu.uni-klu.ac.at/hjele/publikationen/ngramme/2009_ngramme_main.pdf

Lohrum, Stefan; Schneider, Wolfram & Willenborg, Josef (1999): De-duplication in KOBV. Konrad-Zuse-Zentrum für Informationstechnik Berlin. Preprint SC 99-05 (Juni 1999).
Online unter:
http://opus4.kobv.de/opus4-zib/files/393/SC-99-05.pdf

Paulevéa, Loïc; Jégoub, Hervé & Amsalegc, Laurent (2010): Locality sensitive hashing: A comparison of hash function types and querying mechanisms. In: Pattern Recognition Letters (Vol. 31, Iss. 11), S. 1348 - 1358.
Online unter:
http://dx.doi.org/10.1016/j.patrec.2010.04.004

Schneider, Wolfram (1999): Ein verteiltes Bibliotheks-Informationssystem auf Basis des Z39.50 Protokolls. Diploma Thesis, Technische Universität Berlin.
Online unter:
http://wolfram.schneider.org/lv/diplom/diplom.pdf

Stein, Benno & Potthast, Martin (2006): Hashing-basierte Indizierung: Anwendungsszenarien, Theorie und Methoden. In: Workshop Special Interest Group Information Retrieval (FGIR 06) (= Hildesheimer Informatikberichte). S. 159 - 166.
Online unter:
http://nbn-resolving.de/urn:nbn:de:gbv:hil2-opus-672

Unwin, Antony; Theus, Martin & Hofmann, Heike (2006): Graphics of Large Datasets. Visualizing a Million. Springer, New York

Wickham, Hadley (2009): ggplot2. Elegant Graphics for Data Analysis. Springer, Dordrecht.
Auszugsweise online unter:
http://books.google.com/books?isbn=9780387981420

Yan, Su; Lee, Dongwon; Kan Min-Yen & Giles, Lee (2007): Adaptive sorted neighborhood methods for efficient record linkage. In: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, JCDL'07. S. 185 - 194.
Online unter:
http://dx.doi.org/10.1145/1255175.1255213

Dr. Harald Jele
ist Mitarbeiter der Universität Klagenfurt
Adresse:
Robert Musil-Institut
Bahnhofstraße 50
9020 Klagenfurt, Österreich
E-Mail:harald.jele@uni-klu.ac.at