Universität Hildesheim

Internationales Informationsmanagement

Magisterarbeit

Mehrsprachiges Information Retrieval im Rahmen von CLEF 2003

René A. Hackl

Erstgutachter:	Zweitgutachterin:
Dr. Thomas Mandl	Prof. Dr. Christa Womser-Hacker

Hildesheim, im Januar 2004

Zusammenfassung:

Fusion und *Relevance Feedback* sind IR–Strategien zur Verbesserung der Effektivität. Diese Strategien wurden bei der Teilnahme am „multilingual–4“–Task von CLEF 2003 erprobt. Die Ergebnisse sind zufrieden stellend, auch wenn auf Seiten der benutzten Software MySQL deutlich weniger performant war als Lucene.

Abstract:

Fusion and Relevance Feedback are well–known effectiveness improving IR–Strategies. These strategies were experimented with in the CLEF 2003's „multilingual–4“ task employing freely available software. Although MySQL performed considerably less favourable than Lucene, results are passable.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Die Suche nach Information ist ein sehr altes Problem. Schon in den ersten Bibliotheken wurden Bücher ihren Sachgebieten zugeordnet und entsprechend aufbewahrt. Karteikartensysteme wurden entwickelt, mit denen die Suche erleichtert werden sollte. Dieser Vorgang wurde schließlich durch die aufkommenden Personalcomputer automatisiert und beschleunigt. Erste Suchmöglichkeiten, etwa nach Autorennamen und Titeln, wurden verfügbar gemacht. Bestehende Klassifikationen wurden verbessert. Professionelle Recherchen waren außerhalb universitärer Einrichtungen selten gefragt und konnten nur von wenigen Experten[1] durchgeführt werden.

Mit der Entwicklung und Verbreitung des *World Wide Web* hat sich der Fokus verändert. Innerhalb weniger Jahre hat die Informationsversorgung rasant zugenommen. Immer mehr Dokumente liegen in digitaler Form vor. Nicht nur kann praktisch jeder „im Netz“ publizieren, viel wichtiger ist, dass potentiell jeder Zugriff auf die Dokumente hat. Längst schon ist die Informationssuche in großen digitalen Beständen keine ausschliessliche Aufgabe von Recherche–Experten mehr. Häufig muss beruflich wie privat schnell mal „gegoogled[2]“ werden.

[1] Die Verwendung des einen Geschlechts schließt das andere mit ein.

[2] Neologismus nach der Internet–Suchmaschine Google

Nutzer ohne Hintergrundwissen über die Funktionsweise von Retrievalsystemen sehen sich dann häufig vor das Problem gestellt, dass sie im Umgang mit Suchmaschinen nicht die gewünschten Ergebnisse bekommen. Vor allem bei Suchen im Internet tritt dieses Problem deutlich hervor. So hat z.B. eine Untersuchung von [Hölscher und Strube 1999] ergeben, dass eine Anfrage im Schnitt 1.66 Wörter lang ist. Da das Deutsche viele Komposita aufweist, können mit 1.66 „Wörtern“ immer noch präzisere Anfragen gestellt werden, als mit der gleichen Anzahl im Englischen (z.B. Apfelkuchen, Bundesverfassungsgericht vs. apple cake, Federal Constitutional Court). Dennoch bleibt die Effizienz der Suche häufig mangelhaft.

Die Disziplin, die sich mit der Erforschung von Verfahren zum Heraussuchen von Informationen aus Datenbeständen beschäftigt, heißt *Information Retrieval*. Prinzipiell lassen sich hier zwei Seiten unterscheiden, die Nutzerseite und die Systemseite. Die Untersuchung zur durchschnittlichen Anfragelänge zielt mehr auf die Nutzerseite ab. Nutzer können lernen, längere Anfragen mit wohl überlegten Anfragetermen zu formulieren. Im Fokus dieser Magisterarbeit steht die Systemseite. Der Anwendungskontext wird durch die Teilnahme der „Information Science Unit“ des Instituts für Angewandte Sprachwissenschaft der Universität Hildesheim an dem Information Retrieval Evaluierungswettbewerb CLEF 2003 (Cross-Language Evaluation Forum) gebildet.

Der Ruf nach gefestigten Erkenntnissen über die Leistungsfähigkeiten von Suchmaschinen hat dazu geführt, dass sich solche Evaluierungsinitiativen gebildet haben, die ihren Teilnehmern Datensätze zur Verfügung stellen. Diese Datensätze bestehen aus Kollektionen von Dokumenten, Beispielanfragen und Relevanzbewertungen. Durch Versuchsreihen und gezielte Änderungen im Versuchsaufbau werden Erfolgsfaktoren von Suchsystemen erforscht.

Die Basis für die Durchführung der Experimente in dieser Arbeit bildet

das MIMOR–Modell von [Womser–Hacker 1997]. MIMOR bedeutet „Mehrfachindexierung zur dynamischen Methoden–Objekt–Relationierung“. Das Modell demonstriert, wie ein dynamisches, lernfähiges und benutzerspezifisch einsetzbares Information Retrieval System für heterogene Datenbestände geschaffen werden kann. Ein Kernpunkt des Modells ist adaptives Verhalten bei der Fusion von verschiedenen Systemen. Daher bilden Fusionsverfahren einen Hauptbestandteil der vorliegenden Arbeit.

Die praktischen Ziele bestanden im wesentlichen darin

- eine Anwendung zu schreiben, mittels der die einzelnen Systeme verbunden werden. Der Programmablauf soll von bestehenden Indices und vorbereiteten Anfragen vollautomatisch zu Ergebnisdateien im offiziellen CLEF – Format führen.
- *Blind Relevance Feedback* (BRF) als fortgeschrittene IR–Technik zu implementieren. Das BRF soll modular in die Hauptanwendung eingegliedert werden können.
- in einem *preprocessing* genannten Arbeitsschritt die Rohdaten mit geeigneten Mitteln zu bearbeiten. Die Systeme sollten die Daten danach direkt verarbeiten können.
- Optimierungen mit Daten voriger Jahrgänge durchzuführen. Das Training soll mit den Daten und Anfragen von 2001 und 2002 durchgeführt werden. Die Ergebnisse sollten zu aussagekräftigem Lernerfolg der Systeme beitragen.
- die optimierten Systeme im aktuellen Jahr einzusetzen.

Zuletzt wurde außerdem angestrebt, mehr als triviale Ergebnisse im *Task* „multilingual–4“ zu erreichen.

Die Magisterarbeit gliedert sich in sieben Kapitel.

Im hierauf folgenden zweiten Kapitel werden die Grundlagen von Information Retrieval, verschiedene IR–Modelle, Verfahren und Methoden behandelt. Der Schwerpunkt liegt dabei auf den im späteren Verlauf der Magisterarbeit eingeführten IR–Systemen und insbesondere auf dem Cross–lingualen IR und der Bedeutung von Evaluierung und Evaluierungsinitiativen im IR.

Das dritte Kapitel zeigt die im Rahmen der Projektplanung gewonnenen Erkenntnisse auf. Die während der Recherche gefundenen Komponenten werden beleuchtet.

In Kapitel vier wird die Vorbereitung der Rohdaten und der *Topics* erläutert. Neben Problemen maschineller Übersetzung stehen dabei linguistische Verfahren wie *stemming* und das Entfernen von Stopwörtern im Vordergrund.

Im fünften Kapitel geht es dann um die Optimierung und Testläufe mit Daten der beiden vorhergehenden CLEF–Initiativen 2001 und 2002.

Im sechsten Kapitel werden die bei CLEF 2003 erreichten Ergebnisse dann analysiert. Zudem werden einige weitere Erkenntnisse vorgestellt, die auf der Basis nachträglicher Experimente gewonnen werden konnten.

Das siebte Kapitel fasst die Ergebnisse zusammen und überprüft, ob die Erwartungen erfüllt wurden. Zuletzt werden mögliche Verbesserungen im Systemansatz angesprochen.

Im hinteren Deckblatt befindet sich eine CD–ROM, auf der alle eingesetzten Quellen, Ergebnisse etc. zu finden sind.

Alle Internetquellen wurden am 10.01.2004 verifiziert.

Kapitel 2

Grundlagen

2.1 Information Retrieval

Information Retrieval (IR) ist ein fachgebietsübergreifender Begriff, dessen Reichweite unterschiedlich interpretiert wird. Nach [Salton und McGill 1987] fallen

> „die Repräsentation, Speicherung und Organisation von Informationen und der Zugriff zu Informationen".

darunter. Zu dieser sehr universalen Definition zählen dann auch Katalogsysteme, Zusammenfassungen, Inhalts– und Stichwortverzeichnisse.

In diese sehr universale Definition fallen dann auch die frühen Methoden des Bibliothekswesens wie Karteisysteme oder Abstrakte, auch Inhalts– oder Stichwortverzeichnisse zählen dazu.

Eine andere Sichtweise vertritt [Mayfield 2002], nach dem der Kern von IR in

> „the automatic identification of those documents in a large document collection that are relevant to an explicitly–stated information need"

besteht. Mayfields Sichtweise ist technisch geprägt und nimmt an, dass Repräsentation, Speicherung und Organisation von Information eine Vorbedingung für IR darstellt. Abgesehen von der sehr engen Auffassung von IR hat diese

Definition noch zwei große Schwachpunkte. Erstens ist die Frage nach der „Relevanz“ bislang nicht eindeutig zu klären (cf. Kapitel 2.5). Zweitens ist das explizit[1] vorzubringende Informationsbedürfnis ein Grundproblem im IR. Dazu führt [Mandl 2003a] aus:

> „Der entscheidende Aspekt [...] von Information Retrieval besteht in der Vagheit“.

und zitiert wenig später die [Fachgruppe IR 1996]:

> „Vage Anfragen sind dadurch gekennzeichnet, dass die Antwort a priori nicht eindeutig definiert ist.“

IR kann weiterhin abgegrenzt werden von Fakten–Retrieval und Data–Retrieval. Letztgenannte Begriffe werden nicht immer eindeutig und häufig auch synonym verwendet. Dabei ist mit Fakten–Retrieval meist die Suche nach kurzen Beschreibungen von Sachverhalten, mit Data–Retrieval der *total match* beim Vergleich von Zeichenketten gemeint. Um im Beispiel der Einleitung zu bleiben – bei der Suche nach „Bundesverfassungsgericht“ wären bei Data–Retrieval alle Dokumente (und auch nur die), die diese genaue Zeichenkette enthalten, relevant. Im Fakten–Retrieval wäre man bestrebt, in ein bis zwei Sätzen die Institution BVG zu beschreiben. IR würde auch sehr ausführliche Informationen zu Geschichte, Gerichtsentscheiden, rechtlicher Stellung etc. versuchen zu finden. Auch hier tritt die Vagheit deutlich hervor: Detailgrad und Ausführlichkeit der gesuchten Information sind völlig unbekannt.

Trotz ihrer Bezeichnung finden die meisten IR–Systeme im Grunde keine Informationen, sondern Dokumente. Unter einem Dokument versteht man nicht notwendigerweise ein Textdokument, es kann sich auch um ein Audio-, Video-, oder Bild-„Dokument“ etc. handeln. Die gesuchte Information soll sich dann in den Dokumenten befinden. Das *passage retrieval* nimmt eine Zwischenstellung

[1] explicit – adj 1 clear and fully expressed 2 with full details. aus [Longman 1996]

zwischen IR und Fakten–Retrieval ein. Einerseits wird eine Anfrage mit unsicherem Wissen bearbeitet, andererseits ist die Ergebnismenge – die relevantesten Passagen aus den relevantesten Dokumenten – unter Umständen klein genug, um den Anforderungen des Fakten–Retrieval zu genügen.

Das folgende Kapitel über Retrieval–Modelle beschäftigt sich mit den Versuchen „Relevanz“ messbar zu machen.

2.2 Retrieval Modelle

Im Allgemeinen beziehen sich Retrieval Modelle auf den gesamten Ablauf wie in der Salton'schen Definition von IR. Die meisten Ansätze basieren darauf, Dokumente als *bags-of-words* zu betrachten und entsprechend zu indexieren und zu verarbeiten. Die Analyse semantischer Inhalte wird erst in jüngerer Zeit systematisch vorangetrieben und steht noch ganz am Anfang.

Eines der zentralen Probleme von Information Retrieval Systemen ist die Vorhersage, ob ein Dokument relevant ist oder nicht. Wenn die Relevanz aller Dokumente für alle Anfragen bekannt wäre, müsste nur noch eine Liste aus Verweisen erstellt werden, die im Bedarfsfall auf die ideale Antwortmenge weist. Diese Annahme ist natürlich völlig unrealistisch und funktioniert nur für sehr kleine, geschlossene Systeme. Unter realistischen Bedingungen sind also Modelle nötig, die die Ähnlichkeit zwischen Anfrage und Dokument messen, und darüber Aussagen über die Relevanz treffen. Solche Modelle nennt man *Matching*-Modelle.

Von diesen Modellen gibt es sehr viele. Ihre Klassifikation ist je nach Literatur und Perspektive nicht einheitlich (z.B. [Baeza-Yates und Ribeiro-Neto 1999] vs. [Belkin und Croft 1987]). Einigkeit herrscht jedoch darüber, dass die Basis der meisten Modelle die drei klassischen Modelle sind. Sie werden in den folgenden Abschnitten vorgestellt.

$D_1 = \{A, B\}$
$D_2 = \{B, C\}$ Q = A AND B AND NOT C
$D_3 = \{A, B, C\}$

D_1 retrieved
D_2, D_3 not retrieved

Abbildung 2.1: *Einfaches Beispiel zu Boole'schem Retrieval*

2.2.1 Boole'sches Retrieval Modell

Das Boole'sche Retrieval Modell ist ein exaktes Matchmodell. Das heißt, dass alle Bedingungen in einer Anfrage mit „zutreffend" erfüllt werden müssen, damit ein Dokument als relevant gewertet wird. Die Entscheidung über die Relevanz wird also rein binär gefällt – relevant vs. nicht-relevant.

Die Anfrageterme können mittels der logischen Operatoren AND, OR und NOT kombiniert werden. In einem relevanten Dokument

- müssen alle mit AND verbundenen Terme vorkommen. Das hat zur Folge, dass, wenn N Terme mit AND verknüpft sind, aber nur N-1 Terme gefunden werden, das Dokument als nicht-relevant eingestuft wird.

- müssen von mit OR verbundenen Termen wenigstens einer vorkommen. Der Gebrauch von OR ist unproblematischer aber auch unpräziser.

- dürfen mit NOT verbundene Terme nicht vorkommen. Dementsprechend lassen sich mit NOT viele Dokumente herausfiltern.

Ein einfaches Beispiel soll die Funktionsweise demonstrieren. Angenommen, es gibt drei Dokumente D_1, D_2, D_3, die je eine bestimmte Anzahl von Termen [A-C] haben (Abbildung 2.1, nach [van Rijsbergen 1979])

Mit der Anfrage Q, die besagt, dass ein Dokument die Terme A und B enthalten muss, aber den Term C nicht enthalten darf, ist nur D_1 nachzuweisen; D_2 enthält nicht Term A und D_3 enthält den verbotenen Term C.

Operatoren lassen sich mittels Klammerung zu Gruppen beliebiger Komplexität zusammenfassen, z.B. A AND (B OR NOT C). Diese Anfrage würde erfordern, dass in einem relevanten Dokument der Term A gefunden wird, und dass entweder Term B gefunden oder Term C nicht gefunden wird. Wenn beide Bedingungen in der Klammer erfüllt werden – B ist im Dokument enthalten und C nicht – so ist das entsprechende Dokument trotzdem nicht mehr relevant als andere.

Boole'sches Retrieval hat die Vorteile, dass der erste Einstieg grundsätzlich einfach ist, und dass ein klarer Formalismus zugrunde liegt. Die wesentlichen Nachteile sind,

- dass das exakte Matchmodell häufig zu zu großen oder zu kleinen Ergebnismengen führt.
- dass die Ergebnismengen unsortiert sind, d.h. alle der Anfrage entsprechenden Dokumente „gleich“ relevant sind.
- dass mehr als „naive“ (simple) Anfragen nur von Experten durchgeführt werden können.

Das Boole'sche Retrieval Modell gilt heute als nicht geeignet, IR Aufgaben befriedigend wahrzunehmen. [Baeza-Yates und Ribeiro-Neto 1999] erklären dazu, es handele sich mehr um ein Data Retrieval als ein Information Retrieval Modell.

2.2.2 Vektormodell

Das Vektormodell ist ein partielles Matchmodell. Dokumente können auf eine Anfrage teilweise relevant sein, Ergebnismengen werden absteigend nach der Relevanz sortiert. Um die Relevanz zu ermitteln, werden Anfragen und Dokumente als Vektoren in einem mehrdimensionalen Raum dargestellt, der ebenso viele Dimensionen aufweist wie Indexterme (vgl. Abbildung 2.2). Man spricht daher auch von t-dimensionalen Vektoren.

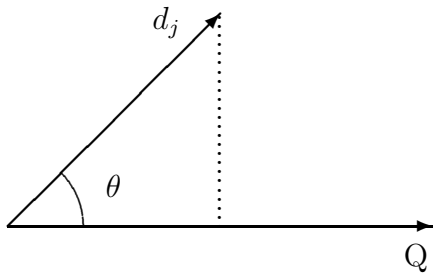

Abbildung 2.2: *Der Kosinus von θ wird als Ähnlichkeitsmaß für $sim(d_j, q)$ genommen.(Abbildung nach [Baeza-Yates und Ribeiro-Neto 1999, S.28])*

Danach werden Vektorenpaare gebildet und auf Ähnlichkeit(= Nähe) überprüft. Als erstes wird also ermittelt, ob ein Term in einem Dokument überhaupt vorhanden ist. Nimmt man für eine einfache Anfrage „t1 t2“ ein Dokument mit den Indextermen „t1 t3 t4“ und eines mit „t1 t2 t4“ an, erhält man eine Matrix von $< 1\ 1 >$ für die Anfrage und für die Dokumente $< 1\ 0\ 1\ 1 >$ bzw. $< 1\ 1\ 0\ 1 >$. In dieser Repräsentation wäre das zweite Modell relevanter, da es beide Anfrageterme enthält. Allerdings fehlen in dieser einfachen Darstellung Aussagen über die Termhäufigkeiten. Termhäufigkeiten werden zum einen lokal für ein Dokument berechnet - dies ist die *term frequency* (tf).

$$tf_{ij} = h(i, j) \tag{2.1}$$

Die *term frequency* wird als die Häufigkeit *h(i,j)* eines Terms t_j in einem Dokument d_i betrachtet. Man geht davon aus, dass je öfter ein Term in einem Dokument zu finden ist, dieser desto besser den Inhalt des Dokuments beschreibt.

Zum anderen werden Termhäufigkeiten auch global für die gesamte Dokumentkollektion angegeben - als *inverse document frequency* (idf). Die IDF ist definiert als

$$idf(j) = \frac{1}{d(j)} \tag{2.2}$$

denn in je mehr Dokumenten d(j) ein Term t_j vorkommt, desto weniger ist er geeignet, jene voneinander unterscheidbar zu machen.

Aus diesen Überlegungen wird geschlussfolgert, dass Termgewichte w_{ij} mit

$$w_{ij} = tf_{ij} \times idf_j \tag{2.3}$$

berechnet werden können.

Diese Formel, die oft auch nur kurz als „tf-idf“ referenziert wird, ist mit erfolgreichen Ergebnissen vielen Veränderungen unterzogen worden.([Salton und Buckley 1988])

Mittlerweile am gebräuchlichsten ist die Berechnung des Kosinus für den Winkel zwischen zwei Vektoren $\vec{d_j}$ und $\vec{q_j}$ als Maß für die Ähnlichkeit. Die klassische Formel, die ebenfalls vielfach angepasst wurde, lautet

$$cos(\theta) = sim(d_j, q) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}| \times |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}} \tag{2.4}$$

Je kleiner der Winkel zwischen den Vektoren ist, desto ähnlicher sind sie. Im Idealfall liegen beide Vektoren auf derselben Geraden, der Winkel ist demnach gleich null und cos(0)=1. Das Vektorraummodell wird in [Ferber 2003] sehr ausführlich dargestellt. Vorschläge für Anpassungen in der Ähnlichkeitsberechnung, um z.B. auch die Länge von Dokumenten in das Ranking mit einzubeziehen, machen [Singhal et al. 1996] und [Singhal 1997]. Die bekannteste Implementierung eines Vektorraummodells in einem IRS ist das SMART-System[2].

2.2.3 Probabilistisches Modell

Probabilistische Rankingmodelle stützen sich auf Erkenntnisse aus der Wahrscheinlichkeitsrechnung. Die Kernfrage hierbei ist,

[2] ftp://ftp.cs.cornell.edu/pub/smart/

„Wie groß ist die Wahrscheinlichkeit, dass ein gegebenes Dokument d für eine Anfrage q als relevant eingeschätzt wird?“
[Ferber 2003]

Je höher ein Dokument als potentiell relevant eingestuft wird, desto höher erscheint es in der Ergebnisliste. Diese Relevanz-Wahrscheinlichkeit lässt sich mit vielen Berechnungsmöglichkeiten ermitteln. An dieser Stelle sollen nur die drei bekanntesten aufgeführt werden.

Das erste Maß für die mögliche Relevanz eines Dokuments ist der **Retrieval-Status-Wert** (retrieval status value[3])

$$rsw = \sum_{\{i \in I \,|\, t_i \in q \wedge d\}} \left(log \frac{r_i}{n_i} + log \frac{(1 - n_i)}{(1 - r_i)} \right) \tag{2.5}$$

Die Herleitung kann in [Ferber 2003] in voller Länge nachvollzogen werden. Ein Problem des RSW ist, dass er Werte >1 zulässt.

Das zweite Maß ist die **Robertson-Sparck-Jones** Formel nach [Robertson und Sparck-Jones 1976].

$$rw_k = log \frac{\frac{(R(q,k)+0{,}5)}{(R(q)-R(q,k)+0{,}5)}}{\frac{(d(k)-R(q,k)+0{,}5)}{(N-d(k)-R(q)+R(q,k)+0{,}5)}} \tag{2.6}$$

mit

R(q) = Anzahl der zur Anfrage q relevanten Dokumente

N = Anzahl aller Dokumente

R(q,k) = Anzahl der relevanten Dokumente, die term k enthalten

n(k)= Anzahl aller Dokumente, die term k enthalten

[3]Die Abkürzung für retrieval status value ist mit RSV leider identisch mit der Abkürzung für Robertson Selection Value (Kapitel 2.3.1). Für eine deutliche Unterscheidung wird „retrieval status value“ mit dem von der deutschen Übersetzung abgeleiteten „rsw“ abgekürzt.

Die Grundannahmen dieses Ansatzes geben [Grossman und Frieder 1998] wieder.

Das dritte und komplexeste Modell ist bekannt als **Okapi BM25**[4].

$$\sum_{t \in Q} \frac{(k_1+1)tf}{(K+tf)} \times \frac{(k_3+1)qtf}{(k_3+qtf)} log_2 \frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)} \tag{2.7}$$

mit

qtf = Häufigkeit eines Terms in der Anfrage

Der dritte Bruch ist nichts anderes als die Robertson-Sparck-Jones Formel in umgeformter Schreibweise. K ist definiert als

$$K = k_1((1-b) + b\left(\frac{l}{avg\_l}\right)) \tag{2.8}$$

mit

l = Länge eines Dokument j in einer geeigneten Einheit – z.B. die Anzahl der Terme, Buchstaben oder Bytes.

avg_l = Durchschnittliche Länge eines Dokuments

Das besondere an diesem Modell sind aber die drei Faktoren k_1, b und k_3. k_1 modifiziert den Einfluss der Termhäufigkeit tf, b wirkt auf die Dokumentlänge. Ausprägungen von b deuten [Jones und Lam-Adesina 2001] so:

> „High values of b imply that documents are long because they are verbose, while low values imply that they are long because they are multitopic.“

Die Werte, die für k_1 und b gewählt werden, bewegen sich meistens in den Intervallen [1,2] und [0.6,0.75] ([Robertson et al. 1996]). k_3 wird dagegen häufig

[4] cf. [Robertson et al. 1996]

auf 0 oder 1000 gesetzt.

Das Okapi–Modell ist im Moment das beste Verfahren im Bereich probabilistischer Gewichtung. Das größte Problem ist jedoch weiterhin, dass die Menge R – die Menge der auf eine Anfrage relevanten Dokumente in der gesamten Kollektion – geschätzt werden muss.

2.3 Information Retrieval Techniken

Die hier beschriebenen Techniken verfolgen die Ziele, mehr relevante Dokumente bei gleichzeitig besserer Präzision zu finden.

Die IR Techniken, die zur Verbesserung der Retrievalergebnisse angewandt werden, lassen sich unterscheiden in Techniken, die auf lokalen Informationen beruhen, und solchen, die auf globalen Informationen beruhen.

2.3.1 Lokale Techniken

Auf der Basis lokaler Information, also dem Inhalt eines Dokuments, sind folgende Techniken gebräuchlich:

- **Grundformenreduktion** – Dieses Vorgehen wird auch *stemming* genannt. *Stemming* ist ein sprachspezifischer Prozess, der Terme auf Endungen überprüft, um diese ggf. zu entfernen. Ein einfaches Beispiel wäre das Entfernen von Plural-s im Englischen („boats“ zu „boat“). Ausgereifte Stemming-Algorithmen erkennen auch Präfixe (z.B. „gegeben“ zu „geben“)und Infixe (z.B. bei slawischen Sprachen). Durch *stemming* wird die Anzahl der einzelnen Terme in einem Index deutlich verringert.

- Eliminierung von Wörtern mit hoher Frequenz – Wörter, die sehr häufig vorkommen, eignen sich nicht zur Unterscheidung von Dokumenten (z.B. „und“). Diese Wörter werden in *Stopwortlisten* festgehalten und mit in Dokumenten und Anfragen vorkommenden Worten verglichen. Wörter, die

sich in der Liste wiederfinden, werden nicht in den Index geschrieben bzw. aus der Anfrage entfernt. **Stopworteliminierung** verringert die Größe von Indizes sehr stark. (cf. Kapitel 4.1)

- Mithilfe von **Synonymlisten** führt man Terme auf einen Grundterm zurück. Beispielsweise könnten „spazieren“, „schreiten“, „wandern“, „schlurfen“ auf „gehen“ zurückgeführt und entsprechend indexiert werden.

- **Thesauri** und **Ontologien** enthalten Verweise auf synonyme Terme, aber auch verwandte und gegensätzliche Terme. Sie sind überdies durch Relationen wie „allgemeiner“ (*broader terms*) und „spezieller“ (*narrower terms*) hierarchisch aufgebaut. Für den Ausgangsterm „Auto“ wäre „Fahrzeug“ der Oberbegriff und „Audi“ ein Unterbegriff.

Während *stemming* und Eliminierung von Stopwörtern anerkannte Verfahren mit reproduzierbar guten Ergebnissen sind, führen die anderen Techniken nicht immer zu den gewünschten Resultaten. Dies hängt z.T. sehr stark vom Anwendungskontext ab, für allgemeine Nachrichtentexte und Zeitungsartikel ist es schwer, sinnvolle Thesauri zu konstruieren, für bestimmte Domänen wie z.B. Chemie oder Physik kann sich ein guter Thesaurus bezahlt machen.

2.3.2 Globale Techniken

Globale Verfahren sind bislang weniger gründlich erprobt worden als lokale. Das liegt vor allem daran, dass erst der technologische Fortschritt in den letzten Jahren die Mittel zu äußerst rechenintensiven Experimenten bereitgestellt hat. Schließlich müssen Verteilungen über ganze Dokumentbestände berechnet werden.

- In **Ähnlichkeitsthesauri** werden Term–Term–Relationen festgehalten. Diese Relationen werden nicht durch Kookurrenz ermittelt, sondern durch Korrelation. Terme werden als Konzepte betrachtet, die von den jeweiligen

Dokumenten indexiert werden. Die Korrelation von Termpaaren wird als Hinweis auf die Deskriptortauglichkeit interpretiert.

- **Statistische Thesauri** werden durch Klassen von *Clustern* erstellt. Diese Klassen müssen in der richtigen Granularität vorliegen, um aussagekräftig zu sein.

Clusteringverfahren sind sowohl auf einzelne Antwortmengen, als auch auf ganze Dokumentkollektionen anwendbar. Dabei werden ähnliche Dokumente in Gruppen (Klassen) eingeteilt. Durch *clustering* können Nutzer auf die verschiedenen Facetten ihrer Anfrage aufmerksam gemacht werden. *Alltheweb*[5] ist eine bekannte Internetsuchmaschine, die *clustering* auf Antwortmengen durchführt.

Eine weitere bekannte Technik ist das *term reweighting* mit oder ohne Anfrageerweiterung. Hierbei werden die Gewichte, die einzelne Terme in der Anfrage haben, verändert. Einen guten Überblick über die möglichen Techniken bieten [Baeza-Yates und Ribeiro-Neto 1999] und [Belkin und Croft 1987].

Relevance Feedback ist in Form von *Blind Relevance Feedback* bei CLEF 2003 eingesetzt worden und wird ausführlich im folgenden Abschnitt beschrieben.

2.3.3 Relevance Feedback

Diese Möglichkeit der Anfrageerweiterung ist in dem Kontext von Ad-Hoc Retrieval mit gutem Erfolg erprobt worden. Ausgehend von den mit einer ursprünglichen Anfrage erzielten Treffern kann man versuchen, die Anfrage zu verbessern, um mehr relevante Treffer zu erhalten, bzw. irrelevante Treffer zu entfernen.

Der Mechanismus, mit dem Relevanzurteile über bestimmte Dokumente abgegeben und auf deren Grundlage neue Anfragen erstellt werden können, lautet *Relevance Feedback*(RF). RF kann vom Nutzer oder vom System kommen.

[5] http://www.alltheweb.com

Das erstere, sog. *User Relevance Feedback*, bei dem ein Nutzer Dokumente aus der Antwortmenge zu seiner ursprünglichen Anfrage betrachtet, Relevanzurteile über ein geeignetes *Interface* abgibt und auf deren Grundlage neue Anfragen erstellt werden können, wird in diesem Zusammenhang nicht weiter betrachtet (hierzu cf. [Baeza-Yates und Ribeiro-Neto 1999, S.118ff]).

Das zweite, das sog. *Blind Relevance Feedback* (BRF, auch Pseudo-Relevance Feedback), bestimmt die Relevanz anhand von vorher festgelegten Kriterien automatisch. Durch die Relevanzinformationen können

- die Terme der Originalanfrage anders gewichtet werden, wie dies von [Croft und Harper 1979] gezeigt wurde,
- Erweiterungsterme ausgewählt werden,
- oder beide Verfahren kombiniert werden ([Attar und Fraenkel 1977]).

Der folgende Abschnitt konzentriert sich auf die Auswahl von Erweiterungstermen, die ohne Neugewichtung zu einer Anfrage hinzugefügt werden.

Im allgemeinen wird so vorgegangen, dass die besten D_q Treffer auf eine Anfrage Q als relevant betrachtet werden. Aus dieser Dokumentmenge werden dann sämtliche Terme gewichtet und nach ihrem Gewicht geordnet. Aus dieser Liste wiederum werden die besten Terme t_q zu der ursprünglichen Anfrage hinzugefügt. Mit der ergänzten Anfrage wird dann der Suchvorgang wiederholt.

Zur Bewertung der Terme können verschiedene Gewichtungsalgorithmen herangezogen werden. Als Ausgangspunkt der Überlegungen wird häufig [Rocchio 1971] zitiert. Dieser Ansatz wurde zwar durch [Salton und Buckley 1990] verbessert, gilt aber im Vergleich mit anderen Verfahren als nicht mehr konkurrenzfähig.

Ein solches Verfahren ist der *Robertson Selection Value* (RSV, [Robertson 1991]). Man erhält diesen Wert, indem man zuerst die Robertson–

Sparck–Jones–Formel anwendet (wie gesehen in Kapitel 2.2.3, hier vereinfacht dargestellt mit aufgelösten Doppelbrüchen)

$$rw(i) = log\frac{(r(i)+0,5)(N-n(i)-R+r(i)+0,5)}{(n(i)-r(i)+0,5)(R-r(i)+0,5)} \tag{2.9}$$

und dann die Anzahl der relevanten Dokumente, die den Term i enthalten mit rw(i) multipliziert:

$$RSV = r(i) \times rw(i) \tag{2.10}$$

Der RSV und Varianten davon sind mit seit Jahren guten Ergebnissen im Einsatz.

Die Abweichungsdistanz nach Kullback-Leibler ist eine weitere Methode zur Auswahl von Termen.

$$w(t) = P_R(t) \times log\frac{P_R(t)}{P_C(t)} \tag{2.11}$$

mit

R = Anzahl der als relevant erachteten Dokumente

P_R = Die Wahrscheinlichkeit, dass Term t in R vorhanden ist

P_C= Die Wahrscheinlichkeit, dass Term t in der Kollektion C drin ist

Die Terme, die am meisten dazu beitragen P_R von P_C zu unterscheiden, also die höchsten Werte w(t) haben, werden in die neue Anfrage übernommen.

Zahlreiche Versuche haben [Carpineto et al. 2001] durchgeführt. Ihre Versuchsreihen belegen, dass von den momentanen Ansätzen RSV und KL in der Regel die besten sind, KL meistens sogar etwas besser abschneidet als RSV.

Da das BRF in multilingualem Kontext implementiert worden ist, ist noch der Zeitpunkt der Erweiterung zu beachten. Die Teilnahmeregeln bei

CLEF legen fest, dass man nur *post-translation feedback* verwenden darf. Eine Anfrage darf demnach erst nach erfolgter Übersetzung erweitert werden. [Ballesteros und Croft 1997] haben anhand des INQUERY[6] Information Retrieval Systems gezeigt, dass eine Kombination aus *pre-* und *post-translation feedback* am effektivsten ist.

Der Erfolg von BRF ist an zwei Bedingungen geknüpft. Zum einen müssen von vornherein genügend relevante Dokumente in einer Kollektion vorliegen, zum anderen muss das zugrunde liegende Suchsystem an sich gut sein. Sonst werden unter Umständen erst gar keine oder nur sehr wenige relevante Dokumente gefunden. Darauf aufgebautes Relevance Feedback kann dann die eigentlich relevanten, aber nur sehr wenigen, Dokumente weit niedriger gewichten, als im ersten Suchschritt.

Für CLEF 2003 ist BRF so implementiert worden, dass die Top 5 Dokumente als relevant erachtet wurden. Aus ihnen sind dann die besten 10 (Optimierung) bzw. 20 (offizielle Runs) Terme extrahiert und der Anfrage hinzugefügt worden. Auch zur besten Kombination dieser Parameter cf. [Carpineto et al. 2001].

2.4 Mehrsprachiges Information Retrieval

Mehrsprachiges Information Retrieval berücksichtigt verschiedene Sprachen. [Oard 1997] unterscheidet zwischen multilingualem IR, bei dem in einem System zwar in mehrsprachigen Dokumenten gesucht werden kann, aber eine Anfrage immer nur gegen Dokumente in der Sprache der Anfrage gestellt wird, und cross-lingualem IR, dass sich darauf bezieht, mit einer Anfrage Treffer in verschiedensprachigen Dokumentmengen zu finden. Diese Unterscheidung wird nicht immer eingehalten. So wäre z.B. der „multilingual-4“ Task bei CLEF 2003 tatsächlich eher ein „cross-lingual-4“ Task. In jedem Fall liegt der Schwerpunkt

[6] beschrieben in [Callan et al. 1992]

hier auf cross-lingualem IR (CLIR).

Die zwei Hauptprobleme mehrsprachigen IR's finden sich in den Bereichen Übersetzung und Erstellen einer Ergebnisliste. Zur Überwindung der Sprachgrenzen gibt es Korpusbasierte und Wissensbasierte Ansätze. Wenn parallele oder vergleichbare Dokumentmengen (Korpora) vorliegen, lassen sich daraus mit Kookurrenzverfahren automatisch assoziative Ähnlichkeitsthesauri gewinnen. Mit diesen werden Anfragen dann „übersetzt". Das Verfahren wurde u.a. von [Sheridan und Ballerini 1996] mit deutschen und italienischen Meldungen der Schweizerischen Depeschen-Agentur und von [Sheridan et al. 1997] auf der Basis mehrsprachiger juristischer Dokumente erprobt. Wissensbasierte Ansätze gehen dagegen von Wörterbüchern oder Ontologien aus, die bereits in 2.3 vorgestellt wurden. Man findet entweder lokal auf einem Rechner zu installierende Software oder geeignete Dienste im Internet.

Eine gute automatische Übersetzung hängt wesentlich davon ab,

- ob Mehrwortkonstrukte (Phrasen) erkannt und adäquat behandelt werden. Ein Beispiel, auf das später noch zurückgekommen wird, ist „fast food". In der Übersetzungsrichtung Englisch–Deutsch mit „schnelles essen" oder ähnlichem übersetzt verfälscht natürlich den Sinn der Anfrage.

- ob Eigennamen identifiziert werden. [Mandl und Womser-Hacker 2003b] haben gezeigt, dass das Vorhandensein von Eigennamen die Präzision von Suchsystemen erhöht. Wenn etwa der Name „Kiesbauer" nicht als Eigenname erkannt wird, ist es möglich, dass er als „gravel farmer" übersetzt in die Anfrage einfliesst (cf. Kapitel 4.2.1).

Beim Erstellen einer mehrsprachigen Ergebnisliste werden Treffer aus verschiedensprachigen Kollektionen in einer gemeinsamen, geordneten Trefferliste zusammengefasst. Es gibt etliche Ansätze zur Fusion mehrerer Trefferlisten in eine einzige. Die einfachsten sind „*raw score merging*"(RSM) und „*round robin*

merging“(RR). Bei RSM werden Ergebnisse aus verschiedenen Kollektionen einfach ihrer berechneten Relevanz nach sortiert. Es wird keine Normalisierung durchgeführt. Die Nachteile dieses Verfahrens liegen darin, dass zum einen eventuell unterschiedliche Gewichtungsalgorithmen gleich behandelt werden und dass zum anderen die Eigenschaften der Korpora (durchschnittliche Dokumentlänge, IDF, Anzahl aller Dokumente etc.) nicht berücksichtigt werden. Für relativ vergleichbare Korpora kann RSM funktionieren.

Bei RR wird eine der Trefferlisten als die erste bestimmt. Von dieser ersten Trefferliste geht dann das erste Dokument als erstes in die gemeinsame Liste. Als nächstes das erste Dokument aus der zweiten Trefferliste, bis alle Einzellisten ihr erstes Dokument an die gemeinsame Liste übergeben haben. Dann werden analog die zweiten, dritten, n-ten Dokumente aus diesen Listen fusioniert, bis ein definierter Schwellenwert erreicht ist, z.B. maximal 1000 Treffer im Gesamtergebnis. RR eignet sich vor allem dann, wenn Relevanzen nicht vergleichbar sind.

Darüber hinaus gibt es reichlich Ansätze zur Normalisierung und Fusion von Trefferlisten. Ein Überblick mit weiterführenden Hinweisen findet sich in [Si und Callan 2003].

Das bei CLEF 2003 verwendete Fusionsverfahren ist bekannt als „*weighted round robin merging*“ und eine Erweiterung des klassischen RR. Zusätzlich zu dem dort illustrierten Vorgehen werden den beteiligten Systemen Gewichte zugewiesen. „Gute“ Systeme erhalten mehr Gewicht an der endgültigen Bewertung als „schlechte“ (cf. Kapitel 5).

2.5 Evaluierung von Information Retrieval Systemen

Die Evaluierung von IRS ist ein breites Forschungsgebiet und der zentrale Bestandteil dieser Arbeit. Man unterscheidet die Bereiche Funktionalität und

Performanz. Als Parameter dieser Bereiche trifft man die Faktoren Laborversuch vs. *real life*–Versuch und *batch* vs. interaktive Anfragen an. Die hier dargestellte und durchgeführte Evaluierung bezieht sich auf Experimente in Labor–ähnlicher Umgebung. Dazu zählt die Wiederholbarkeit und Skalierbarkeit in einer geschlossenen Umgebung. Weiterhin sind die Anfragen im *batch* Modus verarbeitet worden, d.h. dass auf eine gegebene Anfrage ohne Interaktion mit einem Nutzer eine Trefferliste ausgegeben wird. Schließlich kann weiter eingrenzend noch angeführt werden, dass der Bereich Performanz betrachtet wird, die Bereitstellung der Funktionalität wird als gegeben betrachtet.

Damit ist der Rahmen definiert, es muss allerdings noch eine weitere Linie gezogen werden. Bei Untersuchungen dieser Art hat man es mit zwei Größen zu tun: Effizienz und Effektivität. Unter Effizienz versteht man den Einsatz von Ressourcen, die ein System benötigt - die beiden Schlagworte sind hier Zeit (Antwortzeit, Dauer eines Indexiervorgangs) und Raum (Speicherplatz). Mit voranschreiten der technologischen Entwicklungen wurde das räumliche Problem vorerst ausreichend in den Griff bekommen. Durch schnellere Prozessoren ist bei gleichzeitig steigenden Datenmengen zwar auch die Bearbeitungszeit gesunken, wie sich jedoch noch zeigen wird, ist Zeit in dieser Arbeit der knappste Erfolgsfaktor gewesen (Tabelle 5.8).

2.5.1 Effektivitätsbeurteilung

Im weiteren geht es in diesem Abschnitt um die Beurteilung der Effektivität von IRS. Als Effektivität bezeichnet man die Fähigkeit eines Systems auf gestellte Anfragen möglichst präzise und zugleich erschöpfend zu antworten. Die Gewichtung dieser Faktoren ist kontextgebunden. Wenn z.B. ein Nutzer wissen möchte, in welchem Land 1908 die olympischen Spiele stattgefunden haben, genügt ihm womöglich ein einziges Dokument, in dem sich diese Information befindet[7].

[7]Wenn die Informationsquelle als vertrauenswürdig eingeschätzt wird.

Solch ein Dokument soll sich auch möglichst weit vorne in einer Trefferliste befinden, damit Nutzer nicht lange danach suchen müssen. Wenn dagegen ein Patentrechercheur ein bestimmtes Gebiet erschliessen möchte, ist es sehr wichtig, alle diesem Gebiet zugeordneten Dokumente (Patente) nachzuweisen. Die Genauigkeit in der Trefferliste spielt keine so große Rolle mehr.

Die genannten Kriterien Vollständigkeit und Präzision nennt man im IR *Recall* und *Precision*. In Formeln ausgedrückt ist die *Precision* P definiert als

$$P = \frac{Zahl\,der\,nachgewiesenen\,relevanten\,Dokumente}{Zahl\,aller\,nachgewiesenen\,Dokumente} \tag{2.12}$$

und der *Recall* R als

$$R = \frac{Zahl\,der\,nachgewiesenen\,relevanten\,Dokumente}{Zahl\,aller\,relevanten\,Dokumente\,in\,der\,Datenbank} \tag{2.13}$$

Dies sind die beiden Standardmaße zur Evaluierung von IRS. Andere Maße haben sich nicht flächendeckend durchsetzen können. *Recall* und *Precision* werden gegeneinander in Graphen aufgetragen (cf. Abbildung 2.3). Dazu werden elf *Recall*–Stufen zwischen 0.0 und 1.0 ihre *Precision*–Werte zugeordnet. *Recall* 0.1 bedeutet, dass an dieser Stelle 10% aller relevanten Dokumente gefunden wurden. Es ist einfach vorzustellen, dass dies nicht immer glatt aufgeht, daher werden die Werte durch ein Interpolationsverfahren geglättet (cf. [Baeza-Yates und Ribeiro-Neto 1999]).

Ein System, dessen Kurvenverlauf unter gleichen Bedingungen über dem eines anderen System liegt, kann als besser bezeichnet werden. Die größte Kritik an den beiden Quotienten liegt darin begründet, dass Wissen über alle relevanten Dokumente in der Datenbasis angenommen wird. Aber wie erlangt man dieses Wissen und was macht ein Dokument relevant?

Der Frage nach einer Definition von Relevanz nachgehend, zitieren [Salton und McGill 1987] eine Studie von [Cuadra und Katter 1967] nach der

„unter Relevanz die kontextuelle Übereinstimmung zwischen der

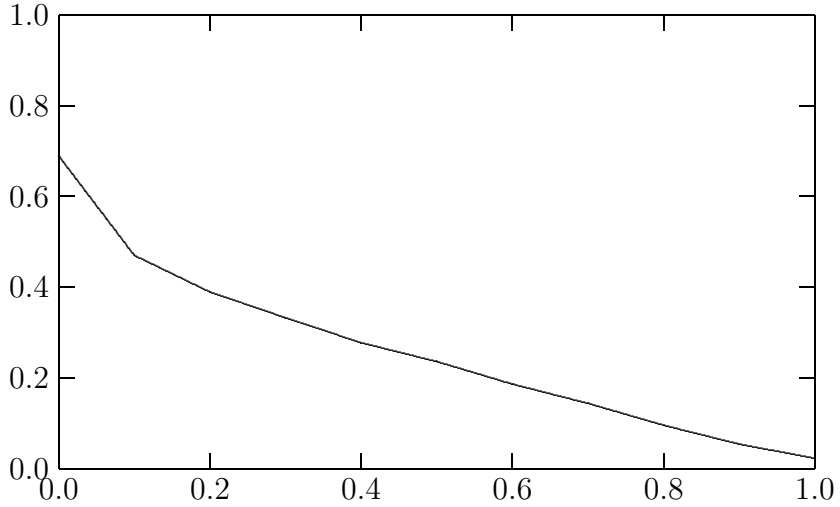

Abbildung 2.3: *Beispiel für einen Recall–Precision–Graph.*

> Suchaussage einer Anfrage (Suchanfrage) und einer Publikation (einem Dokument) bzw. das Ausmaß der Übereinstimmung zwischen einer Publikation und den von der Suchaussage nachgewiesenen Dokumenten zu verstehen ist.“

Auch hier wird die Frage nicht geklärt, schließlich geht es gerade um diesen Übereinstimmungsgrad, der sich schwer definieren lässt. Grundsätzlich unterscheidet man zwischen Systemrelevanz und Benutzerrelevanz.

> „Die Systemrelevanz ist die Entscheidung des Systems über den Nachweis eines Dokumentes bezüglich einer Suchanfrage gemeint. Die Systemrelevanz beschreibt den Grad der formalen Übereinstimmung. Von der Benutzerrelevanz spricht man, wenn der Benutzer diese Entscheidung trifft. die Benutzerrelevanz beschreibt den Grad der vom Benutzer definierten Übereinstimmung.
>
> Ein Dokument wird für einen Benutzer dann relevant sein, wenn ihm das gefundene Dokument einen zusätzlichen Nutzen einbringt. [...] dasselbe Dokument kann für zwei Anwender von unterschiedlichem Nutzen und damit von unterschiedlicher Relevanz sein. Damit ist die Maßeinheit Relevanz etwas Unscharfes und Subjektives.“

[Kukuk 2003]

Letztendlich obliegt die Relevanzentscheidung also bei Menschen. Die benötigten Informationen im Kontext von Recall und Precision werden entweder durch Schätzwerte ermittelt oder aber bei großen Testkollektionen durch **Juroren** erarbeitet.

Testkollektionen sind ein wichtiger Bestandteil von IR Evaluierungsmaßnahmen. Die drei größten Initiativen, die Testkollektionen bereit stellen, sind das europäische Cross-Language Evaluation Forum (CLEF), die NII[8]-NACSIS[9] Test Collection for IR Systems (NTCIR[10]) in Japan und die Text Retrieval Conference (TREC[11]) in den USA. Testkollektionen bestehen aus Dokumentkorpora, beispielhaften Informationsbedürfnissen und Relevanzurteilen. Relevanzurteile werden von den bereits erwähnten Juroren erstellt. Im nächsten Kapitel wird der Ablauf einer Evaluierungskampagne am Beispiel von CLEF nachgezeichnet.

2.5.2 Cross Language Evaluation Forum (CLEF)

Das Cross Language Evaluation Forum (CLEF) ist eine Initiative zur Evaluierung von multilingualen Information Retrieval Systemen und existiert seit 2000. Es ist hervorgegangen aus dem *Cross-Language Track* der Text Retrieval Conference (TREC) in den USA und konzentriert sich auf europäische Sprachen.

Die wichtigsten Aufgabenstellungen, sog. *Tracks*, unterteilen sich in die Bereiche *Core Tracks* und *Additional Tracks*. Zu letzteren gehören Aufgabenstellungen für *Question–Answering*–Systeme, für Bild– und Sprach–Retrieval unter den Bedingungen der Mehrsprachigkeit. Die Kernaufgaben, um die es auch hier gehen soll, gehören zum Ad–Hoc–Retrieval. Sie werden weiter unterteilt in einsprachiges (monolinguales), zweisprachiges (bilinguales) und mehrsprachiges (multilinguales) IR.

[8] National Institute of Informatics
[9] National Center for Science Information Systems
[10] http://research.nii.ac.jp/ntcir/
[11] http://trec.nist.gov/

Registrierung	Dezember 2002
Datenfreigabe	15.01.2003
Topic–Freigabe	01.03.2003
Einsendeschluss der Ergebnisse	15.05.2003
Veröffentlichung der Ergebnisse	01.07.2003
Workshop	21.–22. August 2003

Abbildung 2.4: *Zeitliche Abfolge bei CLEF 2003.*

Evaluierungskampagnen finden jährlich statt. Anhand eines jedes Jahr neu festgelegten Zeitschemas werden verschiedene Phasen durchlaufen (Abbildung 2.4).

Die Daten werden von Zeitungen und Nachrichtenagenturen bereitgestellt. Sie umfassen vollständige Jahrgänge, bei CLEF 2003 waren die Jahrgänge 1994 und 1995 aktuell (cf. Kapitel 4.1). *Topics* sind Themen bzw. Beschreibungen von Informationsbedürfnissen aus denen die Teilnehmer Anfragen erstellen (cf. Kapitel 4.2). Den Prozess zur Erstellung thematisch ausgewogener und sprachlich korrekter *Topics* beschreibt [Womser–Hacker 2002] im Detail.

Eingesandte Ergebnisse werden über alle teilnehmenden Gruppen je *Task* in einem *Pooling*–Verfahren zusammengespielt. Die so erzeugten Ergebnislisten werden von Juroren auf relevante Dokumente überprüft. Dieser Vorgang basiert auf klaren Vorgaben, der hohe Anforderungen an die Juroren stellt. Auch die rein binär zu treffende Unterscheidung in relevante und nicht–relevante Dokumente ist nicht immer klar zu entscheiden. Aus den Relevanzurteilen werden dann die offiziellen Ergebnisse für die einzelnen Teilnehmer ermittelt. Einen umfassenden Überblick über die hier nur skizzierten Abläufe geben [Kluck et al. 2002].

An CLEF 2003 haben insgesamt 43 verschiedene Gruppen teilgenommen, darunter drei aus Deutschland: Universität Hagen, Universität Hildesheim und das

DFKI[12] Saarbrücken. Eine komplette Aufstellung findet sich in Anhang A.5.

[12]Deutsches Forschungsinstitut für Künstliche Intelligenz

Kapitel 3

Versuchsaufbau

In diesem Kapitel wird der Versuchsaufbau dokumentiert. Eine der Hauptanforderungen bestand darin, eine integrierende Umgebung zu schaffen, mit der der vollautomatische Ablauf von multilingualem IR zu realisieren war. Die in Kapitel 2.4 skizzierten Probleme des Fusionierens mussten dabei ebenso Beachtung finden wie die Einbindung von *Blind Relevance Feedback*, die Generierung von passenden Dateiformaten und die Möglichkeit, an verschiedenen Stellen im Prozess Qualitätsproben in Form von fusionierten Listen entnehmen zu können.

Der Versuchsaufbau wird Ablaufdiagramm–artig in Abbildung 3.1 demonstriert. Bestehende Indizes werden vorausgesetzt.

Ausgangspunkt dieses Zyklus sind die Anfragen (Über das Erstellen von Anfragen aus Topics und den Indexaufbau berichtet Kapitel 4). Sie werden in den benötigten Repräsentationen an die registrierten IR Systeme übergeben. Die Suchsysteme führen die Suchen aus und erstellen Ergebnismengen der Art D_{qijs}. Der Indexbuchstabe q steht für die Anfragenummer. Anstelle des i steht die Sprache, der die Ergebnisse zuzuordnen sind. Mit dem j werden die sprachspezifischen Kollektionen identifiziert. Und schliesslich gibt s an, mit welchem Suchsystem die Treffer gefunden wurden.

Eine Beispielergebnismenge für die deutsche Anfrage #151 und in der SDA95,

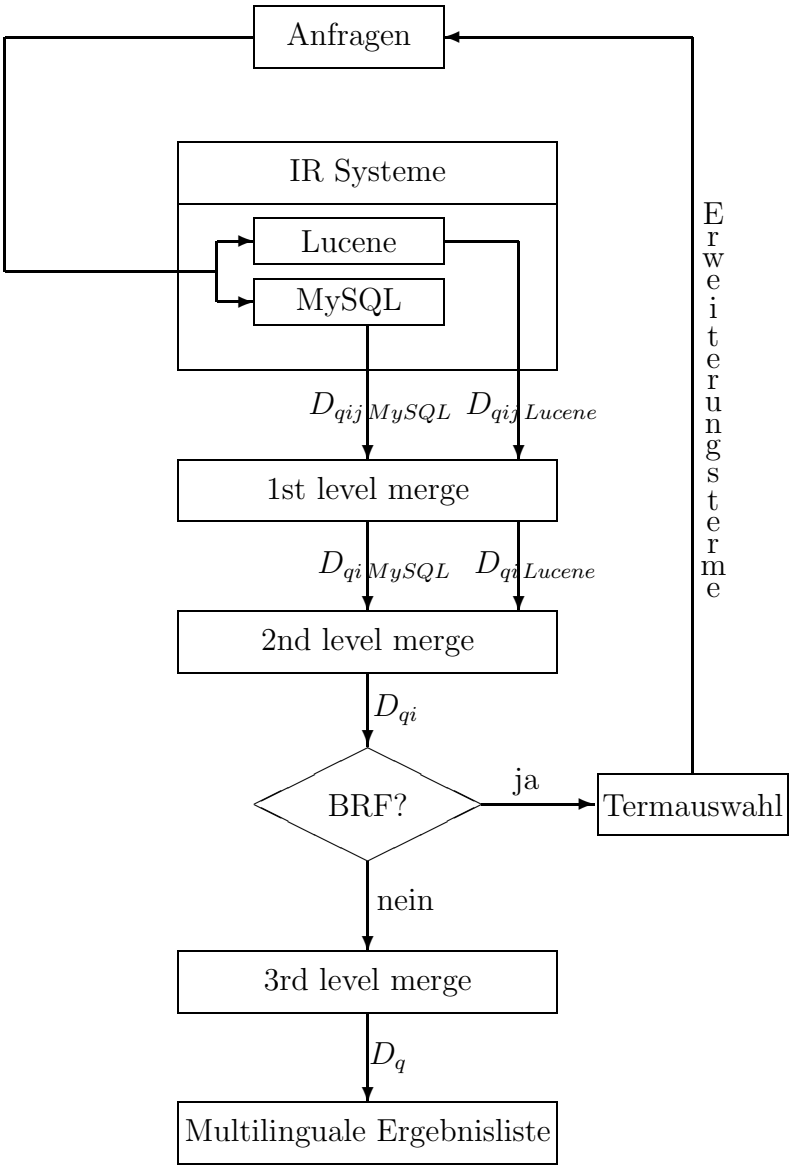

Abbildung 3.1: *Das Ablaufdiagramm beschreibt den Fluss von den Anfragen hin zu einer multilingualen Ergebnisliste.* D_{qij} *ist die Dokumentmenge für Query q in der Sprachgruppe i in der Kollektion j.*

mit Lucene gesucht, hätte demnach die Form $D_{151\,german\,sda95\,Lucene}$.

Der erste Schritt beim Erstellen einer gemeinsamen Ergebnisliste vereinigt dann also die D_{qis} über die beteiligten Kollektionen. Man erhält also z.B. $D_{151\,german\,Lucene}$. In dieser Liste wären die 1000 besten Treffer aus allen Kollektionen per *raw score merging* zusammengefasst.

Dann werden in einem zweiten Schritt die Treffer von den verschiedenen Suchsystemen mittels *weighted round robin merging* in sprachspezifischen Trefferlisten zusammengefasst, für das Beispiel bliebe also $D_{151\,german}$. Darauf folgt die Entscheidung, ob *Blind Relevance Feedback* zur Anwendung kommen soll. Da der gesamte modellierte Prozess vollautomatisch abläuft, ist diese Entscheidung natürlich schon zu Beginn getroffen worden. Bei positivem Bescheid wird das BRF-Modul aktiviert. Es führt eine Termauswahl durch, wobei es zur Berechnung der Gewichte auf die im Lucene–Index abgelegten Termstatistiken zugreift, und ergänzt die bestehenden Anfragen um die Erweiterungsterme. Für den nächsten Durchlauf wird BRF dann deaktiviert. Ohne BRF werden dann in einer letzten Fusion die Trefferlisten aus den einzelnen Sprachen wieder über die *raw score* zu einer einzigen, multilingualen Ergebnisliste verschmolzen ($D_{151\,multilingual}$).

In der Darstellung des Versuchsablaufs wurde schon vorweg auf die implementierten IRS hingewiesen. Im folgenden werden diese Systeme nun vorgestellt. Die Länge und der Detailgrad der Vorstellung sind bereits Indikatoren auf das individuelle Abschneiden der Systeme in den Experimenten und den daraus geschlussfolgerten Erkenntnissen.

3.1 MySQL

Die Volltextsuche ist kein klassisches Betätigungsfeld von relationalen Datenbanken. In neuerer Zeiten warten allerdings immer mehr kommerzielle und *open source* Produkte mit dieser Funktionalität auf. Die Möglichkeiten einer solchen

Lösung sollten im Vergleich zu einer reinen Suchmaschine untersucht werden.

Von den kommerziellen Systemen wurde der *Text Extender* zu IBM's DB2 näher betrachtet. Diese Erweiterung wurde schon von [Li 2002] erfolgreich eingesetzt. Die Installation und Konfiguration der Datenbank war zwar nach einem Tag abgeschlossen, aber die Volltext–Erweiterung konnte nicht wie vorgesehen verknüpft werden. Das eigentlich sehr umfassende Handbuch[1] war nur bedingt hilfreich.

Von den frei verfügbaren Datenbanken verfügte nur MySQL[2] über die benötigte Volltext–Suchmöglichkeit[3]. Die Installation von MySQL 4.0.12 auf einem Tomcat–Server[4] war innerhalb weniger Stunden zu bewerkstelligen, als Administrationswerkzeug wurde phpMyAdmin[5] 2.3.1 eingesetzt.

Die anzulegende Datenstruktur benötigte nur drei Felder (sh. Kapitel 4):

- Das Feld „doc_no“ als *varchar(255)*, gleichzeitig der Primärschlüssel.
- Das Feld „title“ als *mediumtext*.
- Das Feld „text“ als *longtext*.

Die Installation von MySQL ging problemloser, schneller und im Bedarfsfall besser dokumentiert vonstatten als bei DB2. Hier war die Felderbelegung umständlicher und nicht völlig aufgabengerecht. Zum Beispiel müssen für Textfelder Spalten als CLOB (*Character Large OBject*) mit festem Wert definiert

[1] IBM (International Business Machines Corporation) (2000): DB2 Universal Database, DB2 Text Extender Verwaltung und Programmierung. Version 7. Diese Veröffentlichung ist eine Übersetzung des Handbuchs: IBM DB2 Universal Database Text Extender Administration and Programming Version 7. IBM Form SC26-9930-00

[2] http://www.mysql.com

[3] Untersucht wurden ausserdem PostgreSQL 7.3.1, Firebird 1.02, SAP DB 7.4.3

[4] http://jakarta.apache.org/tomcat/

[5] http://www.phpmyadmin.net

werden. Wenn einem CLOB also 40 KB zugewiesen werden, dann werden diese 40 KB je Spalte auf der Festplatte unabhängig von der tatsächlichen Auslastung vollständig belegt. Die Frankfurter Rundschau hatte allerdings einige Dokumente mit einer Größe von mehr als 40 KB. Der Wert müsste für diese Kollektion auf 120 KB erhöht werden, damit alle Dokumente vollständig eingelesen werden. Dabei würden durchschnittlich über 90 KB pro Dokument nutzlos verloren gehen, für die ca. 140.000 Dokumente wäre das ein Kapazitätsverlust von 12.6 GB. Es wird wohl Möglichkeiten geben, dies anders zu regeln, doch wurden sie nicht aus dem Handbuch ersichtlich. Bei MySQL wird der Speicherplatz dagegen dynamisch vergeben, es gibt nur vordefinierte Obergrenzen, für *mediumtext* liegt die Obergrenze bei 2^{24} bytes, also etwa 16 KB, für *longtext* bei 2^{32}, etwa 4.2 MB.

Die Suche auf einem MySQL–Volltext–Index lässt sich allgemein mit

> String select = „SELECT doc_no, MATCH(title,text) AGAINST('„+queryString+"') AS RELEVANCE FROM „+collectionName+"WHERE MATCH(title,text) AGAINST('„+queryString+"')";

ausführen. Anstelle des *queryString* stehen dann die mit boole'schem OR verknüpften Terme. Die mit den Treffern zurückgegebene Relevanz ist von der Anzahl der Suchterme abhängig und theoretisch nach oben hin unbegrenzt. Relevanzen von z.B. „194.6" waren daher ganz normal. Die Normalisierung auf Werte zwischen 0 und 1 wurde so implementiert, dass alle Werte einer Anfrage durch den höchsten Wert geteilt wurden. Dieses Verfahren ist anderen unterlegen (cf. [Singhal 1997]), aber mit relativ wenig Aufwand zu realisieren.

3.2 Lucene

Die Entscheidung für eine frei verfügbare Suchmaschine fiel bereits im Planungsstadium. Kommerzielle Dokumentsuchsysteme sind i.d.R. nicht so frei konfigurierbar, wie man es sich für IR Experimente wünschen würde. Dazu kommen

Lizenzierungskosten; Vorab–Evaluierungen mit voll funktionsfähiger Software erfordern Zeit für Verhandlungen.

Alternativen

Zusätzlich zu der schon bekannten Suchmaschine Lucene[6] sollten mögliche Alternativen gesucht und evaluiert werden. Die Bedingungen an eine potentielle Suchmaschine waren, dass sie

- die gerankte Ausgabe von Ergebnissen unterstützt.
- Schnittstellen für die Verknüpfung mit anderen Programmen aufweist.
- ausreichend dokumentiert ist.
- idealerweise auf Java basiert.

Für die folgende Auswahl trifft in jedem Fall das letztgenannte Kriterium zu.

Die neben Lucene vielversprechendste Suchmaschine war **Egothor**[7], leider war die Website im Frühjahr 2003 einige Wochen nicht erreichbar.

eXist[8] ist eine XML-basierte Datenbank und Suchmaschine. Die Anfragesyntax ist Xpath, die Indexkomprimierung ist allerdings nicht herausragend (inklusive Index 1.2 bis 2mal soviel wie die Rohdaten). Für eXist braucht man einen laufenden Server, die Installation ist relativ einfach, aber auch mehr auf Web-Applikationen und stark strukturierte Daten abgezielt. In der vorliegenden Version 0.9.1 nur mit binärer Relevanzbewertung, daher für uns unbrauchbar.

Das **Information Retrieval Framework**[9] (IRF) war bereits bei CLEF 2002 zum Einsatz gekommen. Dabei sind neben einer schlechten Retrievalleistung auch Probleme mit dem sehr unübersichtlichen und wenig performanten Quellcode aufgetreten.

[6] http://jakarta.apache.org/lucene/docs/index.html
[7] http://www.egothor.org
[8] http://exist.sourceforge.net
[9] http://www.itl.nist.gov/iaui/894.02/projects/irf/irf.html

JXTA Search[10] ist eine Suchsystem für verteilte Umgebungen. *Clients* werden in die Lage versetzt, verschiedene Suchsysteme zu befragen. JXTA verbindet also eher bestehende Suchmaschinen anstatt selbst zu suchen.

NeatSeeker[11] ist vom Anspruch her eine 100%-ig auf Java basierende Programmbibliothek, mit der man einfach Suchmaschinen – vor allem mit Servlets – bauen kann. Das NeatSeeker Projekt wird allerdings seit Januar 2001 nicht weiterentwickelt.

BDDBot[12] wurde von Tim Macinta für sein Buch „Web Developer's Guide to Search Engines“ geschrieben. Dieses Programm ist auch wieder mehr mit Blick auf Web Server und Robots geschrieben, außerdem wurde es seit Jahren nicht mehr weiterentwickelt.

Die französische Suchmaschine **Sdx**[13] (Système Documentaire XML) konnte aufgrund einiger undokumentierter Fehler nicht installiert werden. Die internen Klassen sind jedoch alle von Lucene übernommen. Daher wäre ein Einsatz vor dem Hintergrund der französischen Daten sehr interessant gewesen.

Von den *open source* Suchmaschinen konnte also nur Lucene sinnvoll eingesetzt werden.

Lucene

Lucene gilt als das performanteste nicht-kommerziell verfügbare *Application Programming Interface* (API). API deshalb, da Lucene auf der Seite der im Umfang enthaltenen Demo–Applikationen zwar eine einsatzbereite Suchmaschine ist, aber die tatsächliche Kraft erst entfaltet, wenn etwas Entwicklungs– und Anpassungsaufwand erfolgt ist.

Federführend bei der Erstellung von Lucene war Doug Cutting[14]. Cutting ist

[10] http://www.jxta.org
[11] http://neatseeker.sourceforge.net
[12] http://www.twmacinta.com/bddbot
[13] http://sdx.culture.fr/sdx
[14] http://lucene.sourceforge.net/background.html

auch bekannt als der primäre Autor der Suchmaschine V-Twin[15] und arbeitet momentan als „senior architect“ bei dem Anbieter der Websuchmaschine Excite. Neben einer möglichst unkomplizierten Einbindung in bestehende Applikationen beschreibt er sein Ziel bei der Entwicklung von Lucene als „simplicity without loss of power or performance“[16].

Die wichtigsten Eigenschaften von Lucene:

- Indexkomprimierung auf bis zu 30% des Originaltextes
- Geringe Anforderung an RAM
- Gerankte Ergebnislisten
- Relativ viele *query term operations* möglich
- Feldbasierte Suche
- Mehrsprachige Dokumente unproblematisch zu integrieren
- 100% Java

Diese Eigenschaften werden in den folgenden Kapiteln wieder aufgegriffen und weitergehend illustriert. Es soll zum einen dargestellt werden, wie Lucene in den Experimenten konkret eingesetzt wurde, zum anderen aber auch ein Einblick in die Arbeitsweise von Lucene gegeben werden. Benutzt wurde die Version 1.3RC1[17].

3.2.1 Architektur

In diesem Unterkapitel wird auf den Aufbau von Lucene aus strategischer Perspektive eingegangen. Die wichtigsten Klassen und ihre Funktionen werden vorgestellt. Lucene ist so entworfen, dass die Verarbeitung von Text

[15] Teil von Apple's Copland OS

[16] http://www.javaworld.com/javaworld/jw-09-2000/jw-0915-lucene-p2.html

[17] RC = Release Candidate

hochperformant abläuft. Daraus resultiert, dass man jedwedes Format mit Lucene verarbeiten kann, wenn man den Dateiinhalt in reinen Text konvertieren kann.

Die Schnittstelle von Lucene mit der Außenwelt ist also ein Parser, der Informationen über die zu indexierenden Dokumente und deren Inhalt in Form von Text verarbeitet. Der Ablauf ist so, dass ein Dokument gelesen wird und in eine Software–interne Repräsentation *Document* umgewandelt wird. Dann werden diesem *Document Fields* hinzugefügt. Es ist auch möglich, Systemdaten (Erstellungszeitpunkt, Datum, Größe der Datei etc.) über Fields in einem Document abzulegen. In einem fertigen Index stehen Terme in Fields.

Nach der Erstellung eines Documents sind die Terme als solche aber noch nicht bekannt. Pro Field sind vielmehr *TokenStreams* eingetragen. TokenStreams sind nichts weiter als Aneinanderreihungen von *characters*(Buchstaben, Symbolen, Zeichen). Um aus einem TokenStream die einzelnen Token herauszubekommen, wird er an einen Analyzer übergeben. Analyzer sind für das Verarbeiten von Input zuständig. Sie bestehen aus Tokenizern und Filtern.

Tokenizer haben die Kontrolle darüber, welche Zeichen als Separatoren zwischen Wörtern angesehen werden. Für Texte bietet sich der *WhitespaceTokenizer* an, der Zeichen zwischen Leerstellen als einzelne Token interpretiert. Diese einzelnen Token werden dann durch Filter geschickt.

Filter sind Klassen, die Operationen an Token vornehmen. Entscheidend ist hierbei neben der generellen Auswahl geeigneter Filter die Reihenfolge, in der Token durch Filter gereicht werden. Die in den Experimenten eingesetzten *LowerCaseFilter* – macht aus allen Großbuchstaben Kleinbuchstaben – und *StopFilter* – hier werden Token auf ihre Zugehörigkeit zu einer Stopwortliste überprüft und gegebenenfalls aus dem TokenStream entfernt – sind Bestandteile von Lucene, der *SnowballFilter* – eine Sammlung von auf der Sprache „Snowball“

basierenden Stemmern – gehört zu einer Lucene–Erweiterung[18].

Selbst geschriebene Analyzer, Tokenizer und Filter lassen sich problemlos modulartig einfügen. In den meisten Anwendungsfällen sind solche Anpassungen notwendig. Es muss immer darauf geachtet werden, dass bei der Suche der gleiche Analyzer wie bei der Indexierung eingesetzt wird. Sonst kann es vorkommen, dass Suchterme und Indexterme nicht mehr übereinstimmen und damit keine Dokumente gefunden werden.

3.2.2 Suchoptionen

Einleitend wurde darauf hingewiesen, dass Lucene relativ viele Suchoptionen anbietet. „Relativ“ deshalb, da Lucene unter den *open source* Projekten wohl die meisten Möglichkeiten out-of-the-box unterstützt, aber ein Vergleich mit z.B. Messenger deutlich zu ungunsten von Lucene ausginge. Beispielsweise geht Lucene bei der Suche nach Phrasen davon aus, dass die genannten Begriffe in beliebiger Reihenfolge hintereinander vorkommen. Eine Suche nach „fast food“ fände auch Dokumente, in denen „food fast“ vorkommt.

Die Suche von Lucene läuft *feldspezifisch* ab. Das heisst, dass nur in Feldern, die während der Indexerstellung vorgesehen wurden, gesucht werden kann. Solche Felder können z.B. TITLE, AUTHOR, TEXT etc. heissen und müssen nicht unbedingt groß geschrieben werden. Welche Suchoptionen Lucene zur Verfügung stellt, hängt auch davon ab, wie man Anfragen übergibt. Es gibt eine *convenience*–Klasse namens *QueryParser*, die bestimmte Arten von Suchen unterstützt und automatisch die dazugehörigen Anfragen erstellt. Die Möglichkeiten des QueryParsers waren für dieses Projekt hinreichend.

[18] http://cvs.apache.org/dist/jakarta/lucene/sandbox/snowball/snowball-0.1/
oder
http://cvs.apache.org/viewcvs/jakarta-lucene-sandbox/contributions/snowball

Suche mit

- Phrasen. Verhalten wie in obigem Beispiel.
- boole'schen Operatoren. Es werden „AND“, „OR“ und „NOT“ unterstützt.
- Platzhaltern. *Wildcards* können entweder als „?“, als einstellige Platzhalter, oder als „*“, als mehrstellige Platzhalter auftauchen. Man kann rechts–, links–, binnen– und mehrfach–Platzhalter einsetzen. Die Performanzeinbußen sind dazu analog.
- Reichweitenoperatoren. Mit eckigen Klammern werden Grenzen für Zahlen festgelegt, die automatisch jeden dazwischenliegenden Wert annehmen. Ein Anwendungsbeispiel wäre „PRICE:[500 TO 800]“, dass auf dem Feld PRICE alle Dokumente mit Werten zwischen 500 und 800 heraussucht. Die Eckwerte können in- oder exklusiv sein.
- Abstandsoperatoren. Durch hinzufügen von „˜X“ mit X als eine ganze Zahl kann festgelegt werden, wie weit Terme maximal voneinander entfernt sein dürfen.
- unscharfen Termen (*fuzzy search*). Hinzufügen einer Tilde kennzeichnet Terme als *fuzzy*. Ein intern voreingestellter Parameter bestimmt, wie unscharf gesucht wird (Levenshtein–Distanz).
- Gewichten. Durch das „ˆ“–Zeichen werden Termen oder Ausdrücken Gewichte zugeordnet. Der Grundwert ist immer „1“.

Alle Ausdrücke werden durch geeignete Klammerung zu Gruppen zusammengefasst. Eine Anfrage auf einen korrespondierenden Index könnte lauten:

> YEAR:[1995 TO 2003] AND (TITLE:(“wilcoxon rangsummentest“˜4 OR (genmanipulation AND mäuse)) OR FULLTEXT:(wilcoxon AND rangsummentest AND genmanipulation AND mäuse))

Diese Anfrage sucht alle Dokumente aus den Jahren 1995 bis 2003 heraus, in denen entweder

- im Titel die Terme „genmanipulation“ und „mäuse“ vorkommen oder die Terme „wilcoxon“ und „rangsummentest“ innerhalb von vier Termen zusammen vorkommen.
  oder
- im Volltext alle vier Terme vorkommen.

Die Anfragen der Experimente bestanden aus mit OR verknüpften Termreihen.

3.2.3 Rankingverfahren

Das Rankingverfahren von Lucene ist seit jeher ein beliebtes Thema auf der *Mailing*–Liste. Die FAQ[19] sind hierzu nicht auf dem aktuellsten Stand, vor allem bei der frühen Aussage von Doug Cutting – „For the record, Lucene's scoring algorithm is, roughly, [...]“ – hat das „roughly“ für erhebliche Verwirrung gesorgt – wieso sollte eine mathematische Formel nicht exakt beschrieben werden können?

Mit etwas Aufwand lässt sich die Formel aus dem Quellcode herleiten. Mittlerweile sind aber auch andere Dokumentationen (wie die Java–Dokumentation) auf einem neueren Stand. Die Berechnungsvorschrift lautet jedenfalls

$$score(q,d) = \sum_{tinq} tf(tind) * idf(t) * boost * lNorm * coord(q,d) * qNorm(q) \quad (3.1)$$

mit

tf	Häufigkeit eines Terms in einem Dokument
idf	inverse Häufigkeit eines Terms in der Dokumentkollektion
boost	per default ist der Wert von boost 1.0. Damit findet kein boost statt.

[19] http://lucene.sourceforge.net/cgi-bin/faq/faqmanager.cgi

Man könnte den boost nutzen, um z.B. bei BRF die Originalterme höher zu gewichten als die zusätzlichen Terme.

lNorm dieser Faktor bezieht sich auf die Länge des Felds, in dem ein Term vorkommt. Je länger ein Feld ist – gemessen an der Anzahl der Terme – desto kleiner ist dieser Faktor. Damit soll dem Rechnung getragen werden, dass lange Dokumente normalerweise auch mehr unterschiedliche Terme enthalten als kurze.

coord coord ist umso höher, je mehr von den angegebenen *query terms* in einem Dokument vorhanden sind.

qNorm dieser Normalisierungswert verändert das Ranking nicht, sondern stellt sicher, dass Rankingergebnisse maximal den Wert 1.0 annehmen.

Grundsätzlich erkennt man eine tf/idf–Konstruktion, die um verschiedene Normalisierungswerte ergänzt ist. Das zustandekommen dieser einzelnen Werte soll im Rahmen der Magisterarbeit nicht weiter dargelegt werden. In z.T. hitzigen Debatten haben Subskribenten der Mailing–Liste vergeblich versucht, die Frage nach dem zugrundeliegenden Modell eindeutig zu beantworten. Eine dabei aufgestellte Behauptung war, dass auch ein Vektorraummodell im wesentlichen nichts anderes als ein Probabilistisches Modell wäre, da es auf dem Vergleich von Zahlen beruht und dies eine Domäne der Statistik sei.

Wie auch immer diese Debatte enden wird, festzuhalten bleibt, dass es denkbar schlechtere Algorithmen gibt. Das Hauptproblem der obigen Formel liegt darin, dass durch die Normalisierungen die Relevanzen über verschiedene Anfragen hinweg nicht vergleichbar sind.

Kapitel 4

Datenaufbereitung

In diesem Abschnitt wird zum einen dargestellt, welche Schritte von den Rohdaten zu fertigen Indizes geführt haben, zum anderen, wie aus *Topics* Anfragen erstellt wurden. Entsprechend gliedert sich das Kapitel in die Unterabschnitte „Kollektionen“ und „Topics“. Die zur Durchführung der Experimente benötigten Daten konnten von den passwortgeschützten Bereichen der CLEF-Internetseite[1] bzw. deren Host Eurospider[2] heruntergeladen werden.

4.1 Kollektionen

Für den Task „multilingual-4“ bei CLEF2003 standen insgesamt elf Kollektionen zur Verfügung. Diese Kollektionen bestehen aus vollständigen Artikeln zum Tagesgeschehen, nationalen und internationalen Nachrichten etc. und liegen mit validen DTD's[3] im SGML[4]-Format vor. Task „monolingual-en“ bildet eine Untermenge von erstgenannter Aufgabe ab. Dadurch, dass Englisch als Ausgangssprache festgelegt wurde, konnte während der Erstellung der mehrsprachigen Liste im System an geeigneter Stelle auf die einsprachigen Retrievalergebnisse zugegriffen werden.

[1] http://clef.iei.pi.cnr.it:2002

[2] http://www4.eurospider.ch/CLEF

[3] *Document Type Definition*

[4] *Standardized Generalized Markup Language*

Indexfelder	Dokumentfelder
DOCNO	DOCID DOCNO
TITLE	TITLE TI HEADLINE
TEXT	TEXT TX ST LD LEAD LEAD1

Tabelle 4.1: *Zusammenfassung verschiedener Dokumentfelder zu einem Indexfeld, DOCID und DOCNO haben den identischen Inhalt und lagen in der Regel pro Dokument vor, daher ist ein Element ohnehin redundant. Im Fall des Titels sind TITLE, TI, und HEADLINE das Markup der unterschiedlichen DTD's, gleiches gilt für TEXT und TX. ST (für Subtitle) und LD (für Lead) nehmen eine Sonderstellung ein. Diese Tags wurden den TEXTen zugerechnet.*

Ein Beispieldokument ist in Abbildung 4.1 zu sehen. Von den vorhandenen *Tags* sind jedoch nicht alle freigegeben, d.h. zur Erstellung eines Index durften nur bestimmte Felder benutzt werden. Vor dem Hintergrund, dass manche Kollektionen auch Felder mit intellektueller Beschlagwortung und Kategorisierung hatten, ist dies eine sinnvolle Einschränkung. Tabelle 4.1 gibt eine Übersicht über die freigegebenen Felder und wie diese indexiert wurden.

4.1.1 Indexerstellung

Die Rohdaten wurden zuerst geparst, um die relevanten Tags und deren Inhalt zu extrahieren und in eine einzige große Datei je Kollektion zu schreiben. Damit wurde die Gesamtgröße der Kollektionen um etwa 10% verringert. Dieser Ballast hätte sonst im weiteren Vorgehen Rechenzeit beansprucht und weitere Möglichkeiten für Fehler gegeben, wie noch gezeigt werden wird. Nach diesem Vorgang wurden bei sämtlichen Kollektionen Stopworte entfernt und *Stemming* durchgeführt. Im Anschluss daran war die Größe der Kollektionen auf 49.6% der Originalgröße geschrumpft. Einen Überblick über die Auswirkungen der verschiedenen Verfahren auf die einzelnen Kollektionen zeigt Tabelle 4.2. Die so vorbereiteten Daten konnten dann indexiert werden.

<DOC>
<DOCID>SDA.950101.0001</DOCID>
<DOCNO>SDA.950101.0001</DOCNO>
<LC>D</LC>
<KW>
usa russland tschetschenien kosyrew dole
</KW>
<TI>
USA kritisieren russisches Vorgehen in Tschetschenien.
</TI>
<LD>
Washington, 1. Jan. (sda/afp/dpa/Reuter) Die USA haben das militärische Vorgehen der russischen Armee in Tschetschenien verurteilt. In einem am Sonntag im US-Fernsehsender NBC ausgestrahlten Interview kritisierte der Sicherheitsberater von US- Präsident Bill Clinton, Anthony Lake, die „militärischen Taktiken“ der Russen.
</LD>
<TX>
Zwar unterstützten die USA die territoriale Einheit Russlands. Doch durch die Vorgehensweise der russischen Armee habe es [...]
</TX>
<ST>
Kosyrew: Keine Gefahr einer Rückkehr zum Totalitarismus
</ST>
<TX>
Russlands Aussenminister Andrej Kosyrew hat Befürchtungen zurückgewiesen, dass Moskau in den Totalitarismus zurückfallen könnte. [...]
</TX>
<TX>
Er werde sich Mitte Januar in Genf mit US-Aussenminister Warren Christopher treffen, kündigte Kosyrew an. Das Gipfeltreffen stehe in Zusammenhang mit dem Dialog zwischen Moskau und Washington über die Erweiterung der NATO um Staaten aus Osteuropa. [...]
</TX>
</DOC>

Abbildung 4.1: *Beispieldokument aus SDA95 (aus Platzgründen gekürzt). LC steht für* Language Code*, KW sind die genannten, manuell erstellten und bei CLEF nicht erlaubten, Schlüsselwörter.*

	# Dokumente	Größe(MB)	nur relevante *Tags*(MB)	stemmed (MB)	# Stopworte
rundschau	139715	320	312	188	673
sda94g	71677	144	125	74.1	673
sda95g	69438	141	121	71.7	673
spiegel	13979	63	59	31.9	673
gh95	56472	154	144	76.7	588
lat94	113005	425	375	201	588
efe94	215738	509	458	244	379
efe95	238307	577	517	275	379
lm94	44013	157	135	73.7	478
sda94f	43178	86	73.2	42.4	478
sda95f	42615	88	121	71.7	478
$\sum$	1048137	2664	2393.7	1321.4	N/A

Tabelle 4.2: *Anzahl der Dokumente und Größe in MB der Kollektionen für CLEF2003.*

Der Zeitverbrauch war bei Lucene und MySQL sehr unterschiedlich. Der erste Schritt zum entfernen überflüssiger Element hat 41 Minuten gedauert, das *stemming* dann 6.5 Stunden. Die verschiedenen Indices konnten dann mit Lucene in etwa acht Stunden aufgebaut werden, mit MySql dauerte dieser Vorgang fast 19 Stunden. Der Lucene–Index war mit 708 MB um 46.4% kleiner als die zu indexierenden Daten, der Platzverbrauch von MySQL betrug mit 3309 MB das zweieinhalbfache. Allerdings ist zu beachten, dass zum indexieren auch sämtliche Daten in die Datenbank gelesen werden mussten. Zur Bereinigung zieht man daher 1321 MB ab, so kommt man auf nur noch 1988 MB für die Größe des Index - dies entspricht etwa dem 1.5-fachen der ursprünglichen Datenmenge. Der durchschnittliche Zeitverbrauch für das indexieren lag bei Lucene um 2.79 MB/min, bei MySQL um 1.91 MB/min. Diese Werte stehen bei der Datenbank in einer antiproportionalen Beziehung zur Dateigröße. Je größer die Kollektion, umso langsamer wird MySQL. Lucene's Geschwindigkeit bleibt dagegen konstant.

4.1.2 Fehler in den Kollektionen

Bei der Indexerstellung sind ein paar unerwartete Probleme aufgetreten. Die Daten waren schließlich schon von den Organisatoren validiert worden.

Dokumente enthielten

- nicht SGML–konforme Entitätsreferenzen, vor allem im Zusammenhang mit dem *amper's and*–Zeichen „&“.
- Tagmarkierungszeichen, die unvollendet blieben (<).
- fehlerhafte Byte–Sequenzen.

Bei der Kollektion EFE 1994 traten insgesamt sieben Fehler auf, davon vier solche *markup*–Fehler:

In EFE19940207-03712

> <TITLE> BOSNIA-YIRINOVSKI LIDER PLD DICE RUSIA SALDRA DE LA ONU SI BOMBARDEAN SERBIOS< </TITLE>
> ↑

Diese Fehler kamen ausserdem in den TITLE–Elementen von EFE19940403-01043, EFE19940605-02795 und EFE19940821-10438 vor.

Drei Mal, in EFE19941225-13893, EFE19941225-13896 und EFE19941225-13903, gab es die fehlerhaften Byte–Sequenzen, die betroffene Stellen unlesbar machten. Als Beispiel soll EFE19941225-13896 genügen:

> los EEUU'94 y el cuarto título mundial para eL 0 B â ¬L ¿ Àñ : ÆdÁ 2— úáÃ©U ÈÌ 0 RÜÿàé! ÇB ÿ<ÀMN Éò> à òddT$ jó>y A (Ü Õ los más de 200 pasajeros que permanecen secuestrados en un avión

Bei EFE 1995 kamen in den Dokumenten EFE19950516-10598 und EFE19951012-07941 die gleichen Markup–Fehler wie bei EFE 1994 vor.

Auch der Glasgow Herald hatte in GH951230-000067 so einen Fehler, einen kritischeren hatte das Dokument GH951209-000023:

> <HEADLINE>Forsyth wields tartan tax taunt in Scottish counter-attack<#LIN+ E> Blair bids to wreck Tory 'lie machine'</HEADLINE>

Die deutsche SDA 1994 hatte nur einen Fehler in SDA.940420.0081:

> </. Geschäftsbericht und Rechnung 1993 der PTT (Unternehmungsgewinn 19>

Die LA Times 1994 fiel dadurch auf, dass Dateinamen auf „. sgml“ anstatt auf „.sgml“ endeten. Fernerhin gab es hier viele nicht–maskierte &-Zeichen.

Alles in allem gab es also eher wenige Fehler, die schnell beseitigt werden konnten. Trotzdem ist nicht ganz offensichtlich, wie manche davon durch den Validator kommen konnten.

4.2 Topics

Als *Topics* werden die von der CLEF-Kommission zusammengestellten Beschreibungen von Informationsbedürfnissen bezeichnet. Sie beschreiben das Thema der Suche und grenzen es ein. Wie man anhand des Beispieltopic in Abbildung 4.2 erkennt, kann man je nach benutzten Feldern Anfragen auf verschiedenen Granularitätsebenen erzeugen. Deshalb ist bei offiziellen Einreichungen wenigstens ein Run mit den Elementen *title* und *desc* vorgeschrieben[5] – dieser wird *mandatory run* genannt. Er dient dazu, eine größere Vergleichbarkeit zwischen den Systemen herzustellen. Die Anfragen müssen weiterhin automatisch generiert worden sein. Bei den durchgeführten Experimenten waren alle Läufe von der Gestalt „auto, TD“.

Eine Anfrage ist die Darstellung eines Topics für ein Suchsystem. Für *title* und *description* in Abbildung 4.2 könnte der erste Schritt zu einer Repräsentation so aussehen, dass zunächst Stopwörter entfernt werden und *Stemming* durchgeführt wird:

[5] jedenfalls bei CLEF 2003; in den Vorjahren wurde dies z.T. anders gehandhabt

<top>
<num> C141 </num>
<EN-title>
Letter Bomb for Kiesbauer
</EN-title>
<EN-desc>
Find information on the explosion of a letter bomb in the studio of the TV channel PRO7 presenter Arabella Kiesbauer.
</EN-desc>
<EN-narr>
A letter bomb from right-wing radicals sent to the black TV personality Arabella Kiesbauer exploded in a studio of the TV channel PRO7 on June 9th, 1995. An assistant was injured. All reports on the explosion and police inquiries after the event are relevant. Other reports on letter bomb attacks are of no interest.
</EN-narr>
</top>

Abbildung 4.2: *Beispieltopic aus 2003. Ein Topic (top) besteht aus vier Teilen: num - ein Zahlenwert zur Identifikation, title - eine kurzbeschreibung des Themas, desc - eine etwas längere Beschreibung, narr - eine ausführliche Beschreibung. Da im* Narrative *häufig auch angegeben wird, welche Dokumente nicht relevant sind, dient er auch Juroren zur Orientierung.*
Normalerweise begegnet man der abkürzenden Schreibweise TDN für title, description, narrative, respektive.

explos letter(2x) bomb(2x) studio tv channel pro7 present arabella kiesbauer(2x)

Nun könnte man, wenn das Suchsystem diese Funktionalität anbietet, die mehrfach vorkommenden Terme mit einem höheren Gewicht versehen. Aufgrund fehlender Erfahrungswerte wurde bei CLEF 2003 darauf verzichtet.

Die englischen Anfragen können jetzt schon IRS–spezifisch generiert werden. Für die anderen Sprachen folgt erst die Übersetzung.

4.2.1 Übersetzen der Topics

In den CLEF–Kampagnen ergibt sich noch eine weitere Komponente neben der Umwandlung von Topics in Anfragen: Es müssen Anfragen in verschiedenen Sprachen erzeugt werden. Zur Problematik der Ressourcenfindung und -kombination äußert sich z.B. [Savoy 2001], [Savoy 2002] und [Savoy 2003][6] mit zunehmender Ausführlichkeit und als CLEF–Teilnehmer sehr aufgabenbezogen.

Anfangs steht offen, ob die mehrsprachigen Anfragen aus den Topics erzeugt, oder aus den Topics erst Anfragen erstellt und diese dann übersetzt werden. Das Problem von letzterem Ansatz ist, dass nur noch sehr eingeschränkt – wenn überhaupt noch – Kontextinformationen zur Verfügung stehen. Inwiefern im Moment maschinelle Übersetzungssysteme in der Lage sind, solche Informationen zu nutzen und mehr als eine Wort–für–Wort-Übersetzung zu leisten, ist nicht Bestandteil dieser Untersuchung[7]. Der Entschluss, erst die ursprünglichen Topics übersetzen zu lassen und dann Anfragen zu erstellen, beruht darauf, dass so aktuelle Entwicklungen in jedem Fall berücksichtigt werden.

[6] Zum Zeitpunkt dieser Versuchsreihe noch nicht erschienen

[7] Zu dem Thema cf. [Diekema 2003]

Auswahl der Übersetzungssysteme

Die Auswahl der Übersetzungssysteme erforderte Entscheidungen in drei Bereichen:

- Welche Sprachen sollen abgedeckt werden?
- Welche Sprache wird als Ausgangssprache festgelegt?
- Welche Mittel werden zum Übersetzen eingesetzt?

Die Beantwortung der ersten Frage, die den Anforderungen des Task „multilingual-4“ genügen musste, ergab sich daraus, dass die Sprachen Deutsch, Englisch, Französisch und Spanisch von den Durchführenden auf ausreichendem Niveau verstanden wurden. Schließlich mussten Übersetzungen hinsichtlich ihrer Qualität und gefundene Dokumente hinsichtlich ihrer Relevanz beurteilt werden.

Die Entscheidungen bei der zweiten und dritten Frage hingen eng miteinander zusammen. Schließlich decken viele Systeme die Übersetzung in beiden Richtungen nicht ab. Es zeigte sich außerdem, dass der Organisationsaufwand, der hätte betrieben werden müssen, um kommerzielle Systeme zur Verfügung gestellt zu bekommen, in dem gesteckten zeitlichen Rahmen zu groß gewesen wäre. Daher sollten in erster Linie frei verfügbare, Internet-basierte maschinelle Übersetzungsdienste verwendet werden. Die intellektuell zu evaluierenden Systeme waren babelfish[8], freeTranslation[9], google[10], linguatec[11], und reverso[12].

Bei fünf Systemen und vier Sprachen ergeben sich theoretisch insgesamt 20 zu prüfende Parameter. Für jede Einstellung muss nach den Kriterien

- Beachtung von Rechtschreibregeln

[8] http://babelfish.altavista.com
[9] http://www.freetranslation.com
[10] http://translate.google.com/translate_t
[11] http://www.linguatec.net/online/ptwebtext/index.shtml
[12] http://www.reverso.net

- Behandlung von Bindestrichen
- Identifizierung von Phrasen
- Abdeckungsgrad des Lexikons
- fehlende Wortformen

ein Vergleich durchgeführt werden (hierzu und zu den folgenden Ausführungen ausführlich [Plödt 2003]). Eine erste Sondierung zeigte jedoch, dass nicht alle benötigten Kombinationen von Sprachpaaren von jedem Übersetzungsdienst angeboten wurden. Von den möglichen Ausgangssprachen konnte nur Englisch alle Zielsprachen abdecken.

Nunmehr wurde die Qualität der Übersetzungen evaluiert. Die Systeme produzierten heterogene Ergebnisse unterschiedlicher Brauchbarkeit. Zum Beispiel wurde die Phrase „fast food“ für das Sprachpaar englisch-deutsch nur von Reverso richtig, d.h. gar nicht, übersetzt. Alle anderen Systeme warteten mit Lösungen wie „schnell essen“oder „Schnellimbiß“ auf. In einem weiteren Versuch wurde für das Topic aus Abbildung 4.2 zufällig die Übersetzungsrichtung deutsch–englisch ausgewählt. Dabei wurde der Name „Kiesbauer“ in seine Bestandteile zerlegt und mit „gravel farmer“ übersetzt. Von allen getesteten Übersetzungsdiensten machte nur Reverso diesen Fehler nicht. Es kristallisierte sich heraus, dass kein System allen anderen überlegen war, jede Komponente hatte Stärken und Schwächen[13]. Anhand der Evaluierung wurden schließlich FreeTranslation, Linguatec und Reverso für die Übersetzung aller Topics ausgewählt.

4.2.2 Bearbeitung der übersetzten Topics

Die übersetzten Topics wurden dann nach demselben Verfahren wie die Kollektionen von nicht benötigten Elementen bereinigt, mit einer Stopwortliste

[13]Hierzu cf. wie schon erwähnt [Savoy 2001,2002,2003]

abgeglichen und „gestemmed“.

Dem abschließenden Schritt der Fusion der drei Übersetzungen je Topic kam besondere Bedeutung zu. Wie kann ein Term in einer anderen Sprache ausgedrückt werden; wenn es mehrere Übersetzungen gibt, welches ist die beste; wie sollen diese Möglichkeiten gewichtet werden?

Zur Beantwortung dieser Fragen könnte man Erkenntnisse aus der Evaluation der Übersetzungssysteme einfließen lassen. Wenn man festgestellt hat, dass beispielsweise Reverso tendenziell bessere Ergebnisse für Deutsch und Spanisch erzeugt, dann könnte man diese Terme in der fusionierten Anfrage höher gewichten als solche, die von FreeTranslation und Linguatec übersetzt wurden. Auch könnten Terme, die von mehreren Systemen gleich übersetzt wurden, ein höheres Gewicht erhalten. Aufgrund der Kürze der Zeit konnten aber keine gesicherten Erkenntnisse gewonnen werden. Jeder Term wurde daher gleich gewichtet und kam maximal ein Mal pro Topic vor.

Kapitel 5

Optimierung

Die erfolgreiche Kombination von Information Retrieval Systemen setzt Kenntnisse über die Fähigkeiten dieser Systeme voraus. Heuristiken sind ein Weg, um Stärken und Schwächen zu identifizieren. Ein anderer Weg besteht darin, über Trainingsdaten die Verlässlichkeit der Systeme herauszufinden. Dieser Weg wird hier beschritten.

Mit Optimierung wird angestrebt, die beteiligten Systeme aufeinander abzustimmen und dadurch letztendlich bessere Ergebnisse zu bekommen. Optimierung läuft im Idealfall automatisch als lernender Prozess ab. In dieser Optimierung soll das bestmögliche Gewichtungsverhältnis zwischen den Komponenten Lucene und MySQL herausgefunden werden.

Am Anfang der Gewichtung sind die Komponenten absolut gleichberechtigt behandelt worden. Es wurden keine Voraussagen gemacht, mit denen a priori die Systeme parametrisiert wurden. In einem iterativen Verfahren wurden die Gewichte dann nach und nach dem in dieser Konfiguration besten Verhältnis angenähert. Die Vergleichsbasis, aufgrund der die Gewichte angepasst wurden, fundiert auf den ermittelten Recall- und Precisionwerten. Hierzu wurde das trec_eval Programm von Buckley eingesetzt[1].

[1] ftp://ftp.cs.cornell.edu/pub/smart/trec_eval.7.0beta.tar.gz

Die CLEF–Daten der vorangehenden Jahre bilden die Trainingskorpora. Daher stehen die erst neu dazugekommenen Kollektionen sda95f und sda95g, gh95 und efe95 hier außen vor. Zudem wurden bei der Teilnahme an CLEF 2003 keine italienischen Kollektionen benötigt. Die *relevance assessments* wurden diesbezüglich bereinigt.

Auf Seiten der Topics war die Übersetzung, wie im vorigen Kapitel dargestellt, händisch zu erledigen. Dieser enorme Aufwand wurde nur für die aktuellen 2003er Topics betrieben. Die Optimierung beruht daher auf den offiziellen Topics in der jeweiligen Sprache. Es ist kaum anzunehmen, dass eine schlechtere Qualität der Anfragen – von der bei maschinellen Übersetzungen auszugehen ist – die optimale Gewichtung nachhaltig beeinflussen könnte. Vielmehr sollten die Systeme parallel weniger gute Ergebnisse liefern, so dass das Verhältnis gleich bleibt.

In den folgenden Abschnitten werden zunächst die Daten von 2001 für eine anfängliche Tendenzeinschätzung benutzt. Danach wird diese Einschätzung mit Daten von 2002 verifiziert und ein Gewichtungsschema festgesetzt.

5.1 Optimierung mit Daten aus 2001

In diesen Optimierungsläufen wurden die Anfragen, die aus den Topics von 2001 generiert wurden, an die Suchsysteme gestellt.

5.1.1 Einzelperformanz

Im ersten Durchlauf wurden die in den einzelnen Kollektionen erzielten Ergebnisse für jedes Suchsystem einzeln fusioniert. Die Auswertung ist graphisch in Abbildung 5.1 dargestellt. In der Box neben den Graphen ist die *average precision* neben den Namen der beteiligten Systeme angegeben.

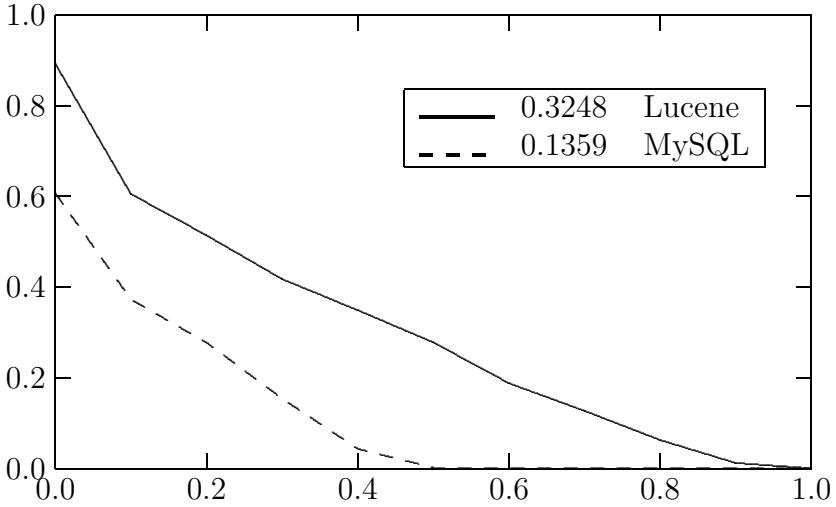

Abbildung 5.1: *Optimierung mit 2001er Daten - Verläufe der einzelnen IRS für die multilinguale Ergebnisliste.*

Lucene schneidet in diesem ersten Vergleich deutlich besser ab. Die erzielte *Precision* liegt für die jeweiligen Recall–Stufen zwischen dem doppelten und dem dreifachen über den von MySQL erreichten Werten (sh. Anhang A.1 für eine vollständige Wertetabelle). Der *Recall* selbst ist mit 5167 um fast 80% höher als bei MySQL mit 2873.

Diese erste Einschätzung ließ schon erwarten, dass sich die Gewichtung deutlich in Richtung Lucene ausdehnen werden würde. Zugleich fing auch schon mit diesem ersten Versuch die Suche nach den Faktoren für MySQL's schlechtes Abschneiden an.

5.1.2 Fusionierte Läufe

Trotz der besseren Individualperformanz von Lucene, werden Lucene und MySQL bei dem ersten Lauf mit fusionierten Ergebnissen wie angekündigt mit 0.5 zu 0.5 gewichtet. Die Gewichte werden so angegeben, dass ihre Summe immer 1 (=100%) ergibt. Diese Darstellungsweise hat gegenüber der in absoluten Zahlen (z.B. 1:1) den Vorteil, dass auch kompliziertere Verhältnisse überschaubar bleiben. Darüber hinaus spiegelt sie auch die interne Repräsentation der

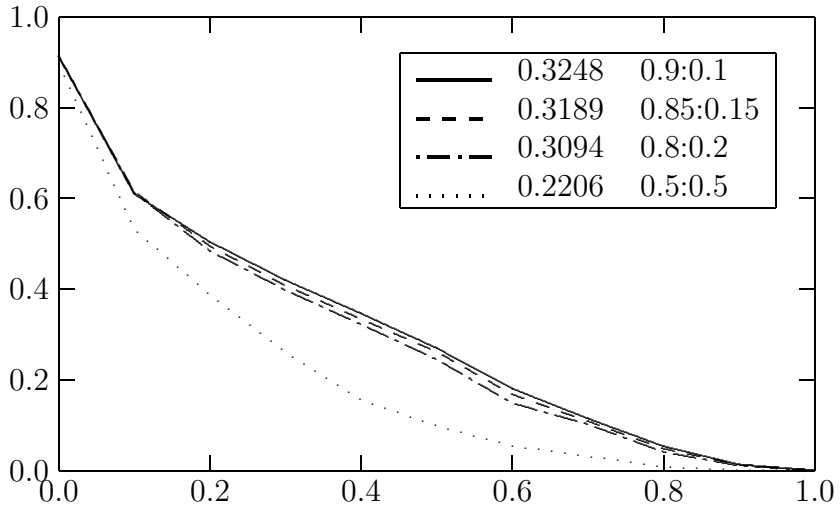

Abbildung 5.2: *Optimierung mit 2001er Daten - Verläufe der fusionierten IRS für die multilinguale Ergebnisliste.*

Gewichte wieder.

Es ist jedoch nicht überraschend, dass diese Gewichtung schlechtere Resultate bringt als Lucene allein. Der Verlust beim Recall beträgt in den einzelnen Sprachen zwischen 7 und 17 Prozent, für den multilingualen Teil beläuft sich der Verlust sogar auf 26%(vgl. Tabelle 5.1 und Tabelle 5.2).

Aufgrund dieser starken Unausgewogenheit wurde der erste Optimierungsschritt relativ groß gewählt. Die nächste Fusion fand mit den Gewichten 0.8 und 0.2 statt und zeigt eine deutliche Verbesserung in der Recall–Precision–Kurve. Die *average precision* stieg von 0.2206 auf 0.3094.

Die Gewichte wurden dann auf 0.85 und 0.15 gesetzt. Auch hier konnte ein weiterer, dieses Mal leichter Anstieg auf 0.3189 beobachtet werden.

Der letzte Lauf wurde mit den Gewichten 0.9 und 0.1 durchgeführt. Mit solchen Einstellungen sind keine besonderen Verbesserungen gegenüber dem besten Einzellauf mehr zu erwarten. Das liegt an der Normalisierung der MySQL–Werte (cf. Kapitel 3.1) und daran, dass bei einem Gewicht von 0.1 selbst ein Dokument mit der Relevanz von 1 nur noch sehr geringen Anteil an dem Gesamtergebnis hat. In einer Umgebung mit nur zwei Komponenten ist 0.9 als Maximalgewicht anzu-

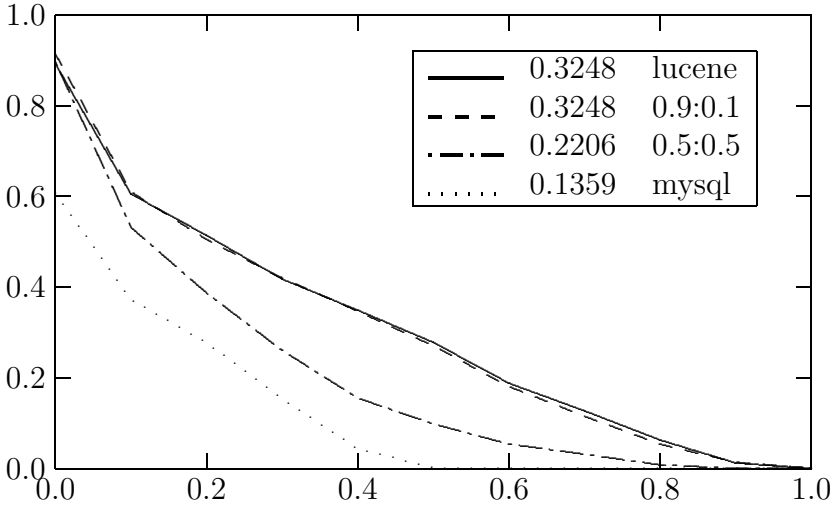

Abbildung 5.3: *Optimierung mit 2001er Daten - Verläufe der besten und schlechtesten Fusion vs. Lucene und MySQL für die multilinguale Ergebnisliste.*

sehen, bei höheren Werten hätte die Fusion keinen Zweck mehr. Einen Überblick über alle fusionierten Runs gibt Abbildung 5.2.

Alle drei Fusionen mit optimierten Gewichten liegen relativ nahe beieinander. In Abbildung 5.3 sind die beiden Einzelruns gegen den besten und den schlechtesten fusionierten Run aufgetragen. Die *average precision* ist für Lucene und den fusionierten Lauf jeweils bei 0.3248, wenn auch die Kurven einen geringfügig anderen Verlauf haben. Der gleichgewichtig fusionierte Lauf liegt erwartungsgemäß relativ mittig zwischen MySQL und Lucene.

Vom Standpunkt der *Precision* betrachtet hat die Fusion keinen Vorteil erzielen können. Nur einmal bei 0.9:0.1 konnte in der englischen Kollektion die *average precision* mit 0.4970 gegenüber 0.4941 bei Lucene verbessert werden (cf. Tabelle A.2).

Mit dem *Recall* als Maß verhält es sich etwas anders. Für die englische Kollektion gibt es gleich drei Mal etwas besseren Recall, für deutsch ein Mal. Die Recallquote liegt für den besten Wert bei 75.0% (cf. Tabelle 5.1).

Die Summe der in den Einzelkollektionen gefundenen relevanten Dokumente

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multil.
$\sum D_r$	856	1212	2130	2694	6892	6892
Lucene	801	**1143**	1873	**2504**	**6321**	**5167**
MySQL	416	547	962	1287	3212	2873
0.5:0.5	744	881	1561	2210	5396	3975
0.8:0.2	**803**	1080	1848	2500	6231	4984
0.9:0.1	**802**	1101	**1874**	**2504**	6281	5101
0.85:0.15	**804**	1094	1865	2503	6266	5056

Tabelle 5.1: *Recall der runs 2001; $\sum D_r$ bezeichnet die Summe der relevanten Dokumente.*

	absoluter Verlust	relativer Verlust
Lucene	1154	18.25%
MySQL	339	10.55%
0.5:0.5	1421	26.33%
0.8:0.2	1247	20.01%
0.85:0.15	1210	19.31%
0.9:0.1	1180	18.79%

Tabelle 5.2: *Verluste beim Recall durch sprachübergreifendes fusionieren 2001.*

ist kleiner als die Anzahl der relevanten Dokumente, die in den multilingualen Ergebnislisten noch vorhanden sind. Diese Verluste sind in Tabelle 5.2 aufgeführt. Auch hieraus lässt sich klar die Tendenz ablesen, dass der Einsatz von MySQL die Effektivität der Suche gehemmt hat. Die Datenbank gibt offensichtlich vielen irrelevanten Dokumenten hohe Werte. Insgesamt finden sich etwas mehr als 80% der in den einzelnen Kollektionen gefundenen relevanten Dokumente in den multilingualen Ergebnislisten wieder.

5.2 Optimierung mit Daten aus 2002

Die erste Optimierungsserie mit den Daten aus 2001 hatte deutlich Lucene favorisiert. Der Optimierungsprozeß wurde nun mit den Daten aus 2002 fortgesetzt. Dabei wurden drei Ziele verfolgt:

1. Überprüfung der bisherigen Ergebnisse auf Stabilität

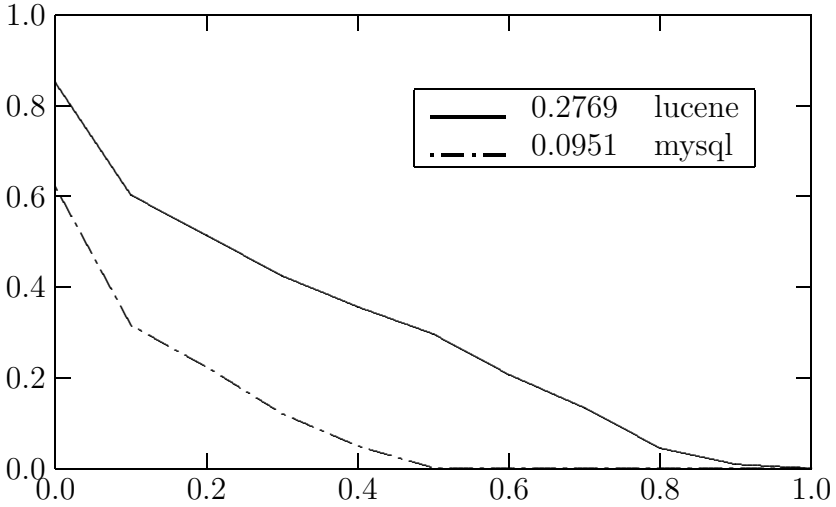

Abbildung 5.4: *Optimierung mit 2002er Daten Verläufe der einzelnen IRS für die multilinguale Liste.*

2. Verfeinerung der Gewichtung

3. Anwendung von *Blind Relevance Feedback*

Analog zur ersten Optimierungsphase wurden zuerst die IRS eigenständig evaluiert. Hier wiederholt sich das im vorigen Abschnitt gewonnene Bild (Abbildung 5.4). Lucene schneidet deutlich besser ab als MySQL. Indes fällt auch auf, dass bei beiden Systemen die *average precision* um vier bis fünf absolute Prozentpunkte niedriger liegt. Auch der *Recall* ist mit 4454 bzw. 2446 von 6996 auffällig schlechter als in 2001.

Besonders schwache Topics waren #95 („Konflikt in Palästina“) mit 219 von 774, #124 („Gemeinsame Außen– und Sicherheitspolitik (GASP)“) mit 108 von 301 und #126 („Aktionen gegen die Pelzindustrie“) mit 48 von 202 relevanten Dokumenten.

5.2.1 Ohne Blind Relevance Feedback

Weiterhin analog zur ersten Optimierungsphase wurden die Systeme jetzt mit verschiedenen Gewichten fusioniert. Dabei kamen als erstes die bislang ermittelten Gewichte zum Einsatz.

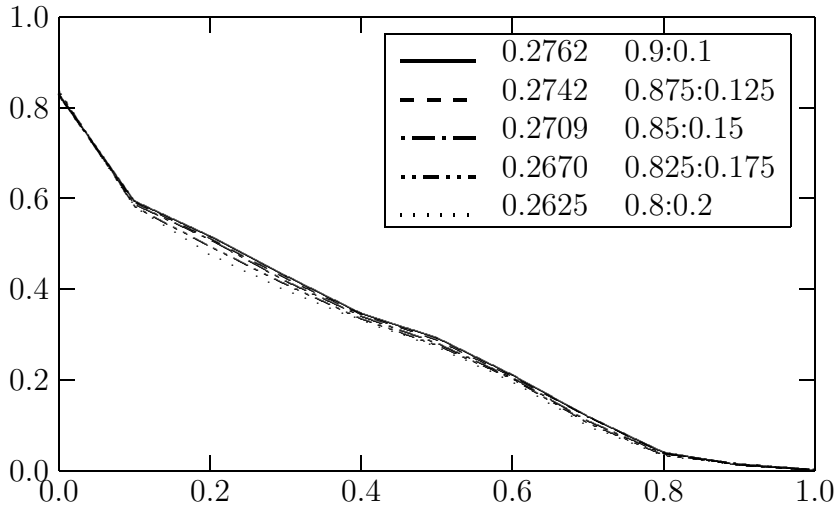

Abbildung 5.5: *Optimierung mit 2002er Daten - Verläufe der fusionierten IRS für die multilinguale Liste.*

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multil.
$\sum D_r$	821	1383	1938	2854	6996	6996
Lucene	**773**	1197	**1478**	2475	5923	4454
MySQL	371	668	731	1297	3067	2446
0.9:0.1	**773**	**1204**	1475	2475	**5927**	4543
0.875:0.125	772	**1203**	1466	**2477**	5918	**4553**
0.85:0.15	772	1185	1448	**2481**	5886	4533
0.825:0.175	770	1167	1438	**2484**	5859	4511
0.8:0.2	769	1141	1427	**2488**	5825	4496

Tabelle 5.3: *Recall der runs 2002.*

Ausgehend von den Resultaten wurden dann zwei weitere Parametrisierungen erprobt, die Schrittweite wurde von 0.5 auf 0.25 verringert.

Der 0.9 zu 0.1 gewichtete Lauf ist mit der *average precision* um 0.0007 unter dem Wert von Lucene geblieben. Die Gewichtung 0.125:0.875 hat auch nur 0.0020 absolute Prozent weniger als 0.9:0.1, andere Gewichtungen belegten keinen Vorteil (vgl. Abbildung 5.5). Insgesamt liegt das Spektrum mit weniger als 1.4% vom schlechtesten bis zum besten Lauf dicht gedrängt zusammen.

Die *Recall*–Werte über beide Jahre sind in Tabelle 5.3 aufgeführt. Bei Lucene

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multil.
$\sum D_r$	821	1383	1938	2854	6996	6996
Lucene 2001	93.57	94.30	87.93	92.95	91.72	74.97
Lucene 2002	94.15	86.55	76.26	86.72	84.66	63.66
MySQL 2001	48.60	45.13	45.16	47.77	46.60	41.69
MySQL 2002	45.19	48.30	37.72	45.44	43.84	34.96

Tabelle 5.4: *Recall der runs 2001 und 2002 in Prozent.*

	absoluter Verlust	relativer Verlust
Lucene	1469	24.80%
MySQL	621	20.25%
0.8:0.2	1329	22.82%
0.9:0.1	1384	23.35%
0.85:0.15	1353	22.97%
0.875:0.125	1365	23.07%
0.825:0.175	1348	23.01%

Tabelle 5.5: *Verluste beim Recall durch mergen in eine multilinguale Liste 2002.*

gab es in 2002 erheblich schlechtere Werte in den französischen, deutschen und spanischen Teilkollektionen. MySQL hatte nur in der deutschen Teilkollektion große Abstriche zu machen, der Recall der französischen Dokumente fiel sogar etwas besser aus.

Der 0.875:0.125 fusionierte Lauf war schon in den Einzelkollektionen in den oberen Wertebereichen, in der fusionierten Liste standen dann mit 4553 insgesamt 99 relevante Dokumente mehr als bei dem Lucene–Lauf. Daran lässt sich erkennen, dass MySQL einige relevante Dokumente sehr hoch bewertet hat, die von Lucene weniger wichtig eingeschätzt wurden.

Dennoch liegt der Verlust beim Erstellen einer multilingualen Liste um 5-6 Prozentpunkte höher als in 2001.

0.875:0.125 wurde als „optimale“ Gewichtung festgelegt. Die *Precision* dieser Verteilung lag nur sehr knapp unter der von Lucene allein. Beim *Recall* konnte, wie gerade gezeigt, ein Vorteil erreicht werden.

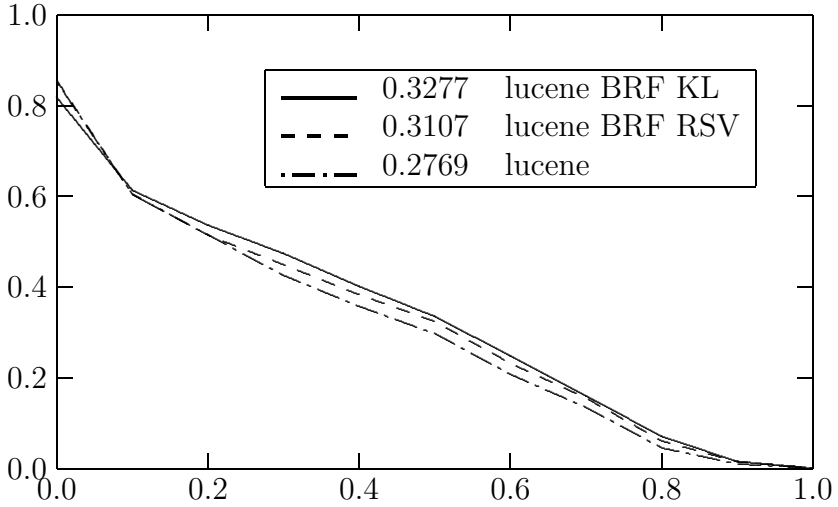

Abbildung 5.6: *Optimierung mit 2002er Daten Verläufe von Lucene allein und mit BRF nach Robertson und Kullback-Leibler.*

5.2.2 Mit Blind Relevance Feedback

Nachdem die Gewichtung festgelegt worden ist, kommt in den folgenden Versuchen *Blind Relevance Feedback* zur Anwendung. Andere Gewichtungsschemata konnten aus Zeitgründen nicht mehr gründlich mit BRF getestet werden. Auch ein Lauf mit nur MySQL fiel der fehlenden Zeit zum Opfer. Ob allerdings BRF bei einem unbefriedigenden IRS noch viel retten könnte, ist mehr als zweifelhaft.

Abbildung 5.6 stellt also dar, wie sich BRF auf einen multilingualen Lucene–Lauf ausgewirkt hat. Die *average precision* macht für BRF mit 10 Erweiterungstermen aus den besten 5 Dokumenten mit dem *Robertson Selection Value* einen Sprung von +3.38 Prozentpunkten, mit der Abweichungsdistanz nach Kullback–Leibler sind es +5.08 Prozentpunkte.

Die Auswirkungen von BRF auf einen Lauf mit fusionierten IRS sind Abbildung 5.7 zu entnehmen. Es zeigt sich auch hier gegenüber dem Lauf ohne Termerweiterung eine Verbesserung von +5.22 Prozentpunkten. Der Lucene–Lauf ist mit 0.0013% Vorsprung nur marginal besser.

In Tabelle 5.6 sind die *Recall*–Werte niedergelegt. Es zeigt sich, dass der Ab-

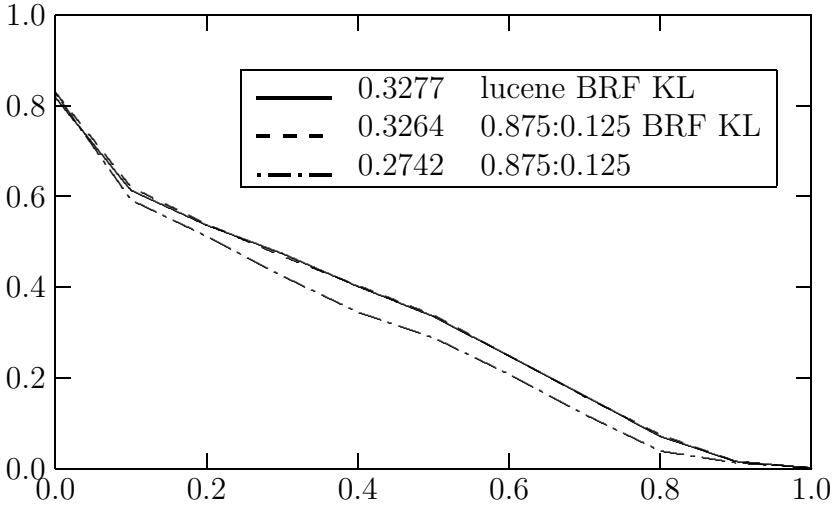

Abbildung 5.7: *Optimierung mit 2002er Daten Verläufe von fusionierten Läufen mit BRF nach Kullback-Leibler.*

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multil.
$\sum D_r$	821	1383	1938	2854	6996	6996
Lucene	773	1197	1478	2475	5923	4454
Lucene BRF RSV	**800**	1280	1650	2585	6315	5059
Lucene BRF KL	799	1297	**1658**	**2623**	**6377**	**5216**
0.875:0.125	772	1203	1466	2477	5918	4553
0.875:0.125 BRF KL	794	**1298**	1639	2588	6319	5157

Tabelle 5.6: *Recall der runs mit Fusion und BRF 2002.*

deckungsgrad durch BRF stark verbessert wird – durch KL noch mehr als durch RSV. Vor allem beim multilingualen fusionieren bleiben bei KL mehr relevante Dokumente übrig. Die 0.875:0.125–Gewichtung, die im direkten Vergleich mit Lucene noch als Sieger hervorgegangen war, kann nicht so stark von BRF profitieren und liegt hier mit 59 Treffern zurück. Insgesamt hat Lucene mit BRF hier eine Recall–quote von 75.6% erreicht, der beste fusionierte Lauf kam auf 73.7%.

Wie BRF geholfen hat die Verluste zu verringern dokumentiert Tabelle 5.7. Auch hier hat KL einen Vorsprung vor RSV.

	absoluter Verlust	relativer Verlust
Lucene	1469	24.80%
Lucene BRF RSV	1256	19.89%
Lucene BRF KL	1151	18.21%
0.875:0.125	1365	23.07%
0.875:0.125 BRF KL	1162	18.39%

Tabelle 5.7: *Verluste beim Recall durch sprachübergreifendes fusionieren 2002.*

5.3 Beurteilung der Optimierung

Im Laufe des Optimierungsprozesses konnten diverse Erkenntnisse gewonnen werden. Als wichtigstes konnte belegt werden, dass Fusion selbst unter ungünstigen Bedingungen einen vorteilhaften Einfluss haben kann. Obwohl ein gutes (Lucene) und ein schlechtes (MySQL) IRS vorlagen, ist eine nützliche Kombination aus beiden gefunden worden. Hierbei konnten auch – in viel kleinerem Rahmen – die Ergebnisse von [Carpineto et al. 2001] bestätigt werden. KL war RSV als Termauswahlkriterium überlegen.

Zeitfaktoren

Die Optimierung hat sehr viel Zeit in Anspruch genommen. Die Dauer einiger Optimierungsläufe ist in Tabelle 5.8 zu sehen. Es ist augenfällig, dass Läufe mit der Beteiligung von MySQL drastisch länger dauern. Die Zeiten, die für BRF angegeben sind, umfassen nicht nur die Berechnung der Erweiterungsterme, sondern auch den folgenden Anfragezyklus. Ausserdem ist der RSV etwas aufwändiger zu berechnen als die KL–Distanz.

Warum ist MySQL so schlecht?

Die Volltextsuche von MySQL wird auf der eigenen Homepage nicht eingesetzt. Statt dessen vertraut man mit Mnogosearch einer Web–basierten Volltextsuchmaschine. Es hat den Anschein, als ob man sich auf Herstellerseite der Unzulänglichkeiten von Datenbank–Erweiterungen für Textretrieval bewusst wäre. Insbesondere sind hier die langen Suchdauern anzuführen.

	Gesamtdauer [min]	davon für BRF [min]
2002 200 Queries		
Lucene	14	
MySQL	91	
0.875:0.125	224	
0.875:0.125 BRF KL 5 10	692	457
0.875:0.125 BRF RSV 5 10	740	509
2003 240 Queries		
Lucene BRF KL 5 20	859	824
0.875:0.125 BRF KL 5 20	1316	978

Tabelle 5.8: *Dauer einiger Optimierungsläufe und der offiziellen Runs.*

Die Gründe für die Performanz in diesen Experimenten sind teilweise dadurch zu erklären, dass

1. Wörter mit weniger als vier Buchstaben automatisch nicht für das Retrieval verwendet wurden (cf. Kapitel 6.2.6).

2. die wenig ausgefeilte Normalisierung bessere Platzierungen verhindert hat.

Die Datenbank kann so angepasst werden, dass sie im Verhalten mehr dem entspricht, was man erwartet hat. Dies setzt ausführliches Studium der Dokumentation voraus. Das Problem der langsamen Zugriffszeiten kann man damit nicht lösen.

Kapitel 6

Ergebnisse

In diesem Kapitel werden die bei CLEF 2003 erreichten Resultate vorgestellt. Zuerst werden die Runs, die durch offizielle Einreichung evaluiert wurden, vorgestellt. Dies sind die Runs mit den Namen UHImlt4R1 und UHImlt4R2 für den Task „multilingual–4“ bzw. UHImnenR1 und UHImnenR2 für den Task „monolingual–english“. Es werden Vergleiche mit den Strategien anderer Teilnehmer hergestellt und dadurch Verbesserungspotentiale erkannt.

Danach werden die Ergebnisse nachträglicher Runs, die auf den Daten von 2003 basieren, vorgestellt und hinsichtlich ihres Zustandekommens untersucht. Insbesondere werden dabei die einzelnen Sprachkollektionen analysiert und Schwächen im bisherigen Ansatz lokalisiert.

6.1 Ergebnisse der offiziellen Runs

Die Priorität bei der Bearbeitung der Tasks lag auf der Erstellung der multilingualen Ergebnisliste. Der einsprachige Teil konnte als Untergruppe des mehrsprachigen ohne zusätzliche Rechenzeit zu beanspruchen extrahiert werden.

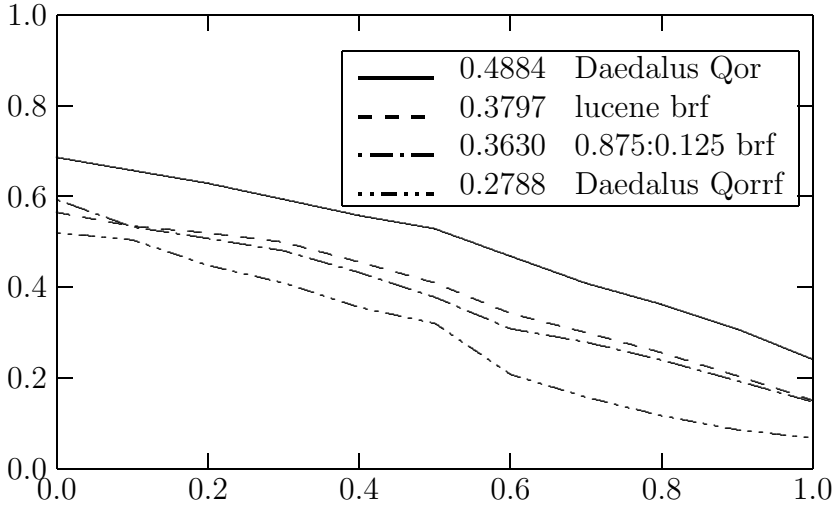

Abbildung 6.1: *Monolingual–englisch Runs.*

6.1.1 Ergebnisse der monolingualen Experimente

Die Ergebnisse für den Task „monolingual–englisch“ waren während der Erstellung der multilingualen Liste zustande gekommen, schließlich war als Ausgangssprache Englisch gewählt worden. Somit konnte mit englischen Originalqueries gesucht werden. Abbildung 6.1 zeigt den Verlauf für unsere beiden Runs im Vergleich mit Daedalus. Daedalus ist eine spanische Gruppe, die mit ihrem System MIRACLE (Multilingual Information RetrievAl for the CLEf campaign)[1] in diesem Jahr zum ersten Mal dabei waren.

Da der einsprachige Task wie gesagt geringe Priorität hatte, soll hier nur darauf hingewiesen werden, dass BRF sehr deutlich sichtbar einen schlechten Einfluss auf die Ergebnisse der spanischen Gruppe hatte. Deren Implementierung mit 250 Erweiterungstermen aus den besten 25 Dokumenten ist allerdings auch kaum alltäglich. Qor = Queries alle mit OR verknüpft, Qorrf = Queries mit OR verknüpft und zusätzlich (blind) relevance feedback.

Unsere Läufe lagen im Vergleich mit den anderen Teilnehmern im unteren

[1] cf. [Borri und Peters 2003, pp. 115–124]

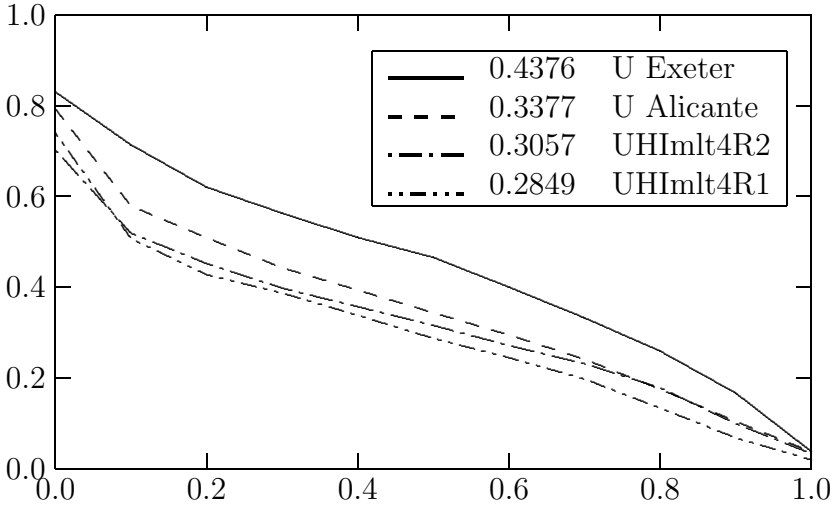

Abbildung 6.2: *Ergebnisse 2003 - Verläufe der Runs UHImlt4R1 und UHImlt4R2 im Vergleich mit den Ergebnissen der University of Exeter(1.) und der Universidad de Alicante(5.).*

Mittelfeld.

6.1.2 Ergebnisse der multilingualen Experimente

In diesem Teil werden die Ergebnisse im Task „multilingual-4“ vorgestellt. In Abbildung 6.2 sind die beiden eingereichten Läufe im Vergleich zu zwei anderen Teilnehmern zu sehen. Insgesamt wurde bei 14 Teilnehmern der 8. Platz belegt. der Durchschnitt der *average Precision* lag bei 0.2752.

Der Ansatz der Universität Alicante (Llopis und Muñoz 2003 [Llopis und Muñoz 2003]) besteht darin, anstatt auf Volltexten auf Passagen zu suchen (*passage retrieval*). Die Größe einer Passage bestimmen Llopis und Muñoz durch die Anzahl der pro Passage zugelassenen Sätze. In der Regel waren dies bei CLEF 2003 sechs Sätze. Dokumente, deren Passagen nun hohe Relevanzwerte erhalten, werden als Ergebnisse ausgegeben. Zusätzlich wendet die Gruppe Anfrageerweiterung und Kompositazerlegung von deutschen Wörtern. Ersteres resultiert in bis zu +3% *average precision*, letzteres verbessert die *average precision* für deutsche Kollektionen um bis zu 5%.

Anders ist der experimentelle Aufbau der Universität Exeter (Lam-Adesina, A.M.; Jones, G.J.F.; 2003, pp. 143-151). Das zugrundeliegende IR System ist das bekannte Okapi System (cf. Kapitel 2.2.3). Der integrierte Porter Stemmer wurde benutzt, nachdem 260 Wörter über eine Stopwortliste entfernt wurden. Außerdem fand eine kleine Synonymliste Anwendung.

Bei BRF wurde ein technisch sehr ausgereifter Ansatz angewendet: Zunächst wurden aus den obersten fünf Dokumenten Zusammenfassungen aus den sechs besten Sätzen erzeugt (Methode beschrieben in [Jones und Lam-Adesina 2001]). Die Terme aus den Zusammenfassungen wurden dann über eine leicht veränderte Version des Robertson Selection Value gerankt. Die Veränderung ging dahin, dass die Auswahlmenge zwar aus den Top 5 Dokumenten bestand, die Berechnung des Ranking dann aber auf Basis der Top 20 Dokumente geschah. Einerseits wird so sichergestellt, dass nicht zu viele „schlechte“ Terme in die Auswahl geraten, andererseits haben auch nachfolgende, potentiell relevante Dokumente Einfluss auf die Erweiterung.

Die 20 besten Terme wurden schließlich zur ursprünglichen Anfrage hinzugefügt. Die Originalterme in der Anfrage wurden um Faktor 3.5 höher gewichtet als die Erweiterungsterme. Für die multilingualen Ergebnisse wurden verschiedene *merging* Strategien evaluiert. Diese Studie von Jones und Lam–Adesina hat – im Gegensatz zu anderen – gezeigt, dass *raw score merging* nicht schlechter ist als andere Verfahren. Trotz aller Versuche, die in diesem Feld durchgeführt wurden, konnte immer noch kein klarer Favorit herausgearbeitet werden.

Während der Optimierung konnte Lucene mit *precision*–Werten von mehr als 32% aufwarten, bei den offiziellen Läufen mussten Einbußen hingenommen werden. Unser bester Lauf liegt nur bei 0.3057. Verluste gab es auch beim Recall, statt um die 75% wurde nur noch eine Recallquote von 65% bei R1 bzw. 67% bei

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multil.
$\sum D_r$	1006	946	1825	2368	6145	6145
UHImlt4R1	951	816	1413	1967	5147	3944
UHImlt4R2	953	846	1478	1954	5223	4137

Tabelle 6.1: *Recall der runs 2003.*

R2 erreicht (Tabelle 6.1).

Die Analyse der multilingualen Ergebnisliste hat ergeben, dass insbesondere die Topics 164, 181 und 197 schlecht abgeschnitten haben (Tabelle 6.2). Da sehr hohe Recall–Werte erreicht werden konnten, sind hier durch nur drei Topics bereits jeweils an die 1000 Dokumente bereits verloren gegangen. Die *Precision* ist nur für Topic 181 an der unteren Grenze der Annehmbarkeit. In den nächsten Kapiteln wird das Abschneiden dieser Topics in den jeweiligen Sprachen weiter verfolgt werden.

6.2 Ergebnisse nachträglicher Runs

Einige Zeit nach der Veröffentlichung der offiziellen Ergebnisse werden die von den Juroren erstellten Relevanzurteile den Teilnehmern zur Verfügung gestellt. Mit diesen sogenannten *qrels* (oder auch *relevance assessments*) lassen sich in Verbindung dem trec_eval–Programm und Ergebnislisten in geeignetem Format weitere Experimente evaluieren.

Die überprüften Parameter waren Lucene vs. fusionierte Systeme, BRF vs. kein BRF und original Queries vs. (maschinell) übersetzte Queries. Dazu waren bei drei Faktoren mit je zwei mögliche Ausprägungen $2^3 = 8$ Läufe nötig. Die Ergebnisse dieser Läufe werden zuerst sprachspezifisch, danach sprachübergreifend untersucht. Mit einem * gekennzeichnete Läufe haben zu den offiziellen Ergebnissen beigetragen.

Topicnummer	164
Titel	European Drug Sentences
	Europäische Rauschgift-Sätze / Rauschgift-Urteile / Medikamentensätze
	La Droga europea Sentencia / Oraciones de droga europeas
	Phrases européennes de la drogue / Le Médicament européen Condamne
Recall	R1: 55 / R2: 41 von 260
Precision	R1: 0.0210 / R2: 0.0121
Topicnummer	181
Titel	French Nuclear Tests
	Französische Kerntests / Atomprüfungen
	Pruebas nucleares francesas
	aux Essais nucléaires français de Trouver
Recall	R1: 358 / R2: 382 von 814
Precision	R1: 0.2613 / R2: 0.2617
Topicnummer	197
Titel	Dayton Peace Treaty
	Friedensvertrag / Friedensabkommen von Dayton
	Tratado de paz de Dayton
	Le Traité de Paix de Dayton
Recall	R1: 247 / R2: 204 von 543
Precision	R1: 0.1495 / R2: 0.1100

Tabelle 6.2: *Analyse der Topics 2003.*

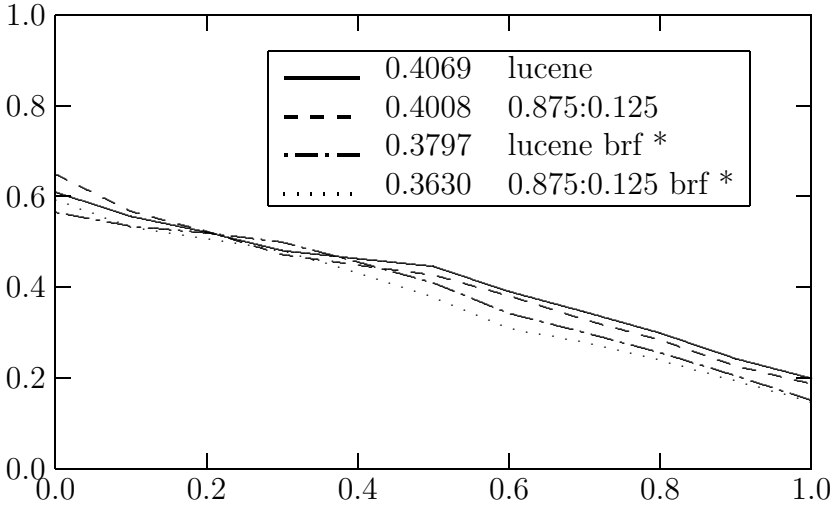

Abbildung 6.3: *Verschiedene monolingual–englisch Runs.*

6.2.1 Englisch

Sämtliche Runs beruhen auf den englischen Topics. Daher gibt es hier keine Übersetzungen und die Zahl der verstellbaren Einstellungen beträgt nur noch 2^2. In Abbildung 6.3 sind diese vier Verläufe aufgezeichnet. Der die Systeme fusionierende Run schneidet etwas schlechter ab als derjenige, der auf Lucene allein beruht. Die Anwendung von BRF hat in beiden Fällen die Ergebnisse verschlechtert.

Eine intellektuelle Überprüfung hat ergeben, dass die Erweiterungsterme dennoch in der Mehrzahl sinnvoll waren. Allerdings hatten sich ja schon bei Daedalus Probleme mit BRF für die englische Kollektion offenbart. Auch [Savoy 2003, S.185] berichtet hier von starken Verlusten, gleiches belegen [McNamee und Mayfield 2003, S.23] für 5–grams. Die englische Topic/Kollektionen–Kombination schien in diesem Jahr wenig geeignet für BRF.

Das Abschneiden der drei fraglichen Topics kann in Tabelle 6.3 abgelesen werden. Die Recall–Quoten sind in allen Fällen in Ordnung, die *Precision* ist nur für Topic 181 akzeptabel.

Topicnummer	164
Recall	R1: 20 / R2: 20 von 27
Precision	R1: 0.0471 / R2: 0.0597
Topicnummer	181
Recall	R1: 91 / R2: 91 von 92
Precision	R1: 0.5164 / R2: 0.5375
Topicnummer	197
Recall	R1: 50 / R2: 50 von 50
Precision	R1: 0.0984 / R2: 0.1388

Tabelle 6.3: *Analyse der englischen Topics 2003 in den offiziellen Runs.*

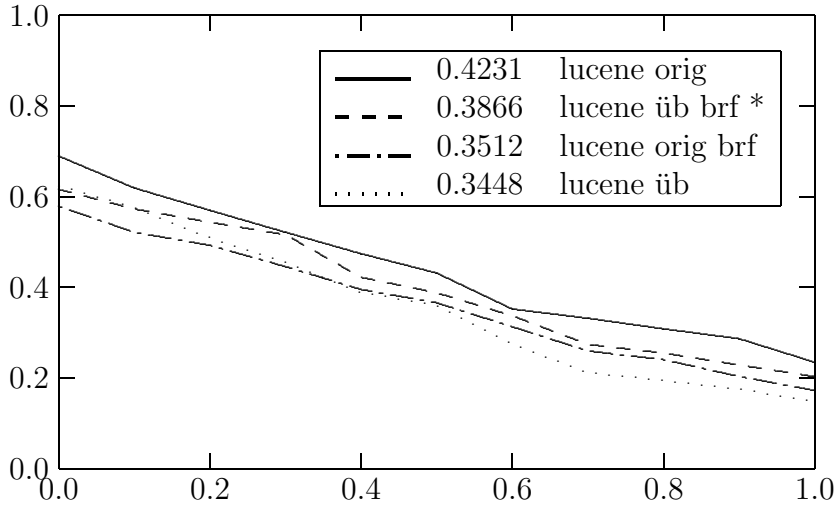

Abbildung 6.4: *Verschiedene Lucene–basierte monolingual–französisch Runs.*

6.2.2 Französisch

Die Resultate für die Experimente mit den französischen Kollektionen sind für die Lucene-basierten Experimente Abbildung 6.4 und für die fusionierten IRS Abbildung 6.5 zu entnehmen.

Die Runs mit aus Originaltopics erstellten Anfragen erweisen sich erwartungsgemäß als den übersetzten Anfragen überlegen. Auch dass Lucene allein besser als die Fusion abschneidet, ist nach der Optimierung keine große Überraschung, wenn auch die Differenz etwas hoch ist. Es fällt auf, dass die Anwendung von BRF bei den Originalanfragen die *Precision* verschlechtert, bei den übersetzten

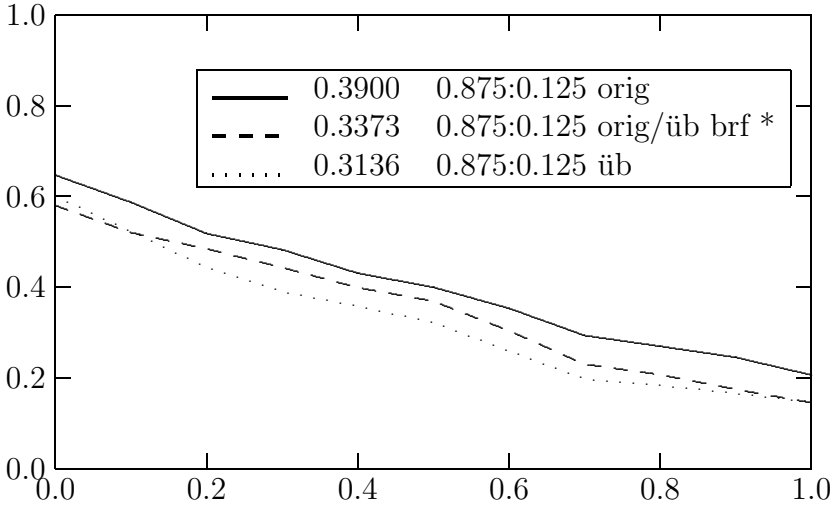

Abbildung 6.5: *Verschiedene fusionierte monolingual–französisch runs.*

Anfragen die *Precision* verbessert hat. Der fusionierte Lauf mit BRF hat für übersetzte und originale Topics seltsamerweise die gleiche *average precision*.

Im Allgemeinen hat die Termerweiterung in den französischen Kollektionen dazu geführt, dass auffällig viele schlechte Deskriptoren zu der Anfrage hinzugefügt wurden. Regelmäßig kamen Terme aus der folgenden Liste vor: l, d, a, est, y, il, ont, on, êtr. Die meisten dieser Terme oder ihrer denkbaren Herkunft waren jedoch über die Stopwortliste ausgeschlossen.

Das Abschneiden der drei Vergleichstopics ist in Tabelle 6.4 aufgeführt. Für #181 und #197 waren z.T. gute Resultate erzielt worden, #164 war vor allem in der *Precision* sehr schwach.

6.2.3 Deutsch

Die Anfragen auf die deutschen Kollektionen hatten insgesamt sehr gute Ergebnisse zur Folge. Die *average precision* betreffend konnten hier die höchsten Werte in allen Teilkollektionen erzielt werden. Ein Erfolgsfaktor war das *relevance feedback*.

Topicnummer	164
Recall	R1: 19 / R2: 38 von 89
Precision	R1: 0.0054 / R2: 0.0343
Topicnummer	181
Recall	R1: 192 / R2: 192 von 193
Precision	R1: 0.3688 / R2: 0.4422
Topicnummer	197
Recall	R1: 119 / R2: 122 von 131
Precision	R1: 0.2660 / R2: 0.4011

Tabelle 6.4: *Analyse der französischen Topics 2003 in den offiziellen Runs.*

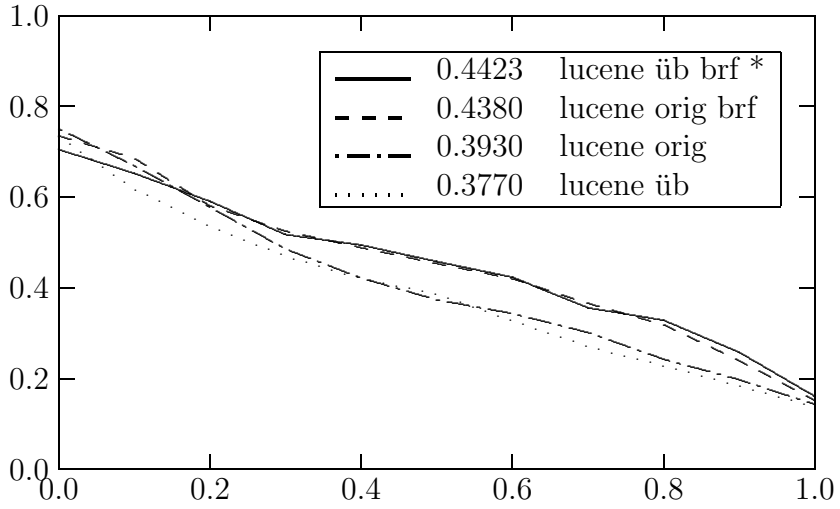

Abbildung 6.6: *Verschiedene Lucene–basierte monolingual–deutsch Runs.*

Wie aus Abbildung 6.6 bzw. Abbildung 6.7 hervorgeht, haben die Anfragen, die mit BRF weiterentwickelt wurden, deutlich bessere Werte erzielt als solche ohne Erweiterung. Bemerkenswert ist außerdem, dass die Anwendung von BRF bei den übersetzten Queries noch vorteilhafter gewirkt hat als bei den Originalanfragen.

BRF bewirkt bei „lucene üb“ einen absoluten Anstieg von +6.53% und einen relativen Anstieg von +17.32%. Bei „lucene orig“ sind dies absolut +4.50% und relativ +11.45% und bei den fusionierten Läufen sind es +4.09 und +13.64 bzw.

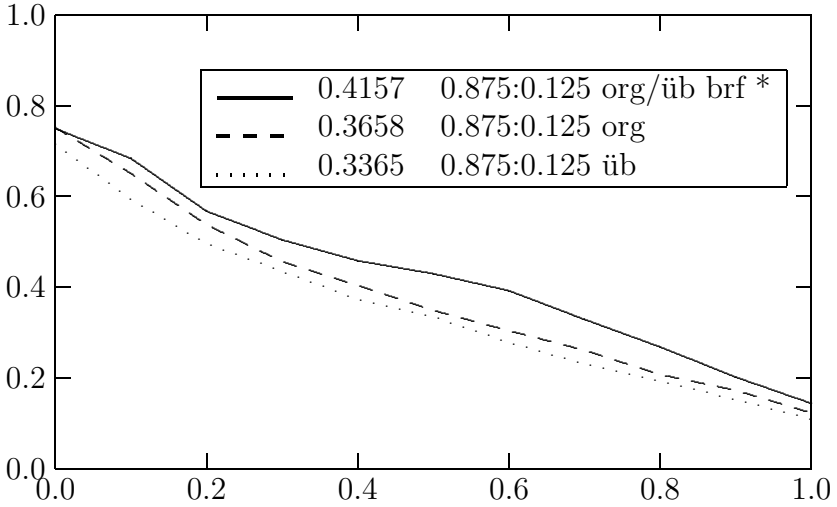

Abbildung 6.7: *Verschiedene fusionierte monolingual–deutsch Runs.*

Topicnummer	164
Recall	R1: 56 / R2: 47 von 72
Precision	R1: 0.1294 / R2: 0.1351
Topicnummer	181
Recall	R1: 25 / R2: 46 von 226
Precision	R1: 0.0027 / R2: 0.0186
Topicnummer	197
Recall	R1: 112 / R2: 119 von 122
Precision	R1: 0.3797 / R2: 0.5014

Tabelle 6.5: *Analyse der deutschen Topics 2003 in den offiziellen Runs.*

+9.92 und +23.54. Auch bei diesen Verläufen ist das Phänomen zu beobachten, dass die Graphen der übersetzten und originalen Anfragen mit BRF absolut identisch verlaufen.

Aus Tabelle 6.5 geht hervor, dass die Referenztopics eher durchwachsen abgeschnitten haben.

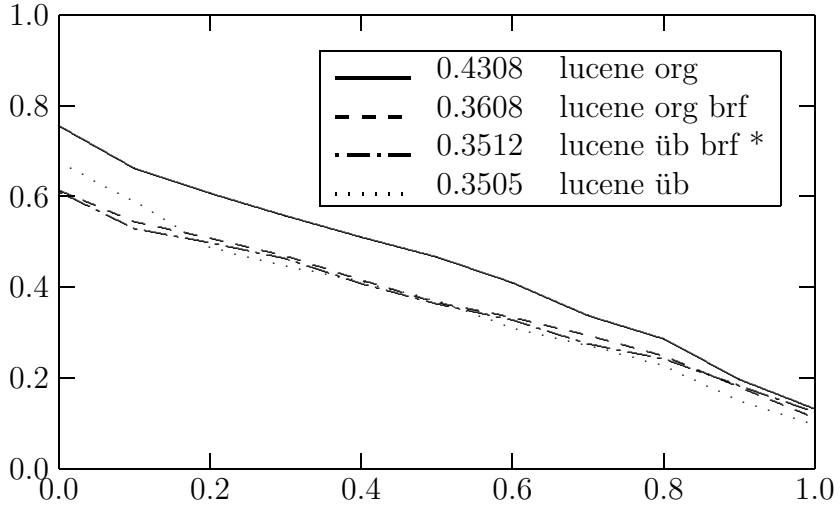

Abbildung 6.8: *Verschiedene Lucene–basierte monolingual–spanisch Runs.*

6.2.4 Spanisch

Die spanischen Subkollektionen hatten in der Optimierungsphase z.T. sehr gut abgeschnitten. Für die Anfragen von 2003 erfüllten sich die optimistischen Erwartungen allerdings nicht (vgl. Abbildungen 6.8 und 6.9).

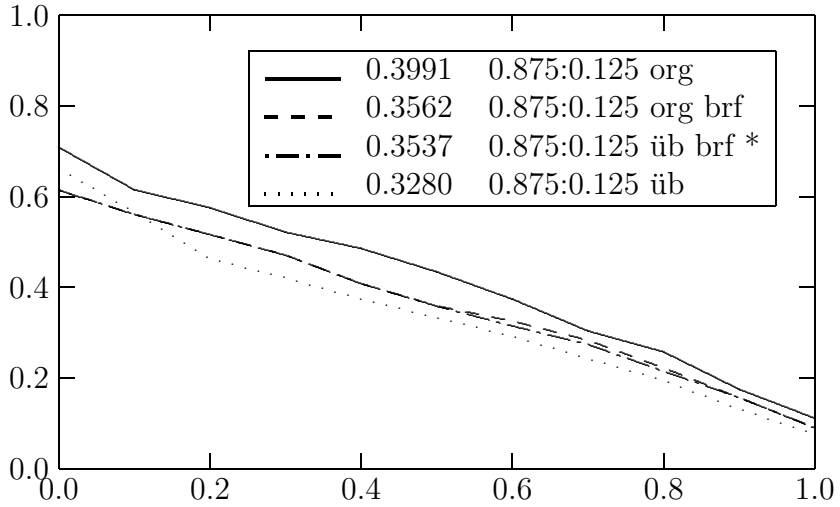

Abbildung 6.9: *Verschiedene fusionierte monolingual–spanisch Runs.*

Die originalen Anfragen kamen ohne BRF auf 43% bzw. knapp 40%, mit BRF sogar nur auf 36% bzw. 35.6%. Das maximal erreichbare Ergebnis für

übersetzte Anfragen lag bei gut 35%.

Die Lucene–Runs mit den originalen Anfragen sind den übersetzten Anfragen jedes Mal überlegen. Das gleiche Bild zeichnet sich bei den fusionierten Systemen ab. Allerdings ist die Richtung, in der BRF wirkt, entgegengesetzt. Für erstere Anfragen wirkt BRF zum Teil dramatisch verschlechternd (von 43.08% auf 36.08% bzw. von 39.91% auf 35.62%), auf letztere wirkt BRF demgegenüber verbessernd auf die *average precision* - einmal sehr gering von 35.05% auf 35.12%, das andere Mal von 32.80% auf 35.37%.

Die Wirksamkeit von BRF wurde in den spanischen Kollektionen sehr durch Autorenkürzel und Datumsangaben gebremst. Am Ende von „TEXT"–Elementen waren Angaben zu finden wie „sar-ez/ik", „lm/fv" oder „07/19/15-47/94". Da diese Angaben nicht extra gekennzeichnet waren, wurden sie als Deskriptoren in die Indizes aufgenommen und schließlich Anfragen durch BRF hinzugefügt. Eine typische, erweiterte Anfrage sah wie hier Anfrage #184 („Maternity Leave in Europe") aus:

> hoj acerc europ longitud dej disposicion provision encuentr document permis matern matern permis disfrut 06/07/10-05/95 razones 11/09/11-38/95 07/19/15-47/94 episiotom 04/21/17-55/94 05/03/05-32/95 irmtraut 122.500 gavarr pospart hums disposicion mujer socia-democrat carabañ muj

Solche Anfragen sind selbstverständlich nur noch eingeschränkt nützlich.

Die Performanz der drei verfolgten Topics ist gemischt. Es gab sehr gute Werte für #181, mittelgute Werte für #197 und geradezu „mieserable" Werte für #164 (Tabelle 6.6).

Topicnummer	164
Recall	R1: 15/ R2: 6 von 72
Precision	R1: 0.0030 / R2: 0.0005
Topicnummer	181
Recall	R1: 285 / R2: 253 von 303
Precision	R1: 0.6228 / R2: 0.5617
Topicnummer	197
Recall	R1: 184 / R2: 148 von 240
Precision	R1: 0.3755 / R2: 0.1790

Tabelle 6.6: *Analyse der spanischen Topics 2003 in den offiziellen Runs.*

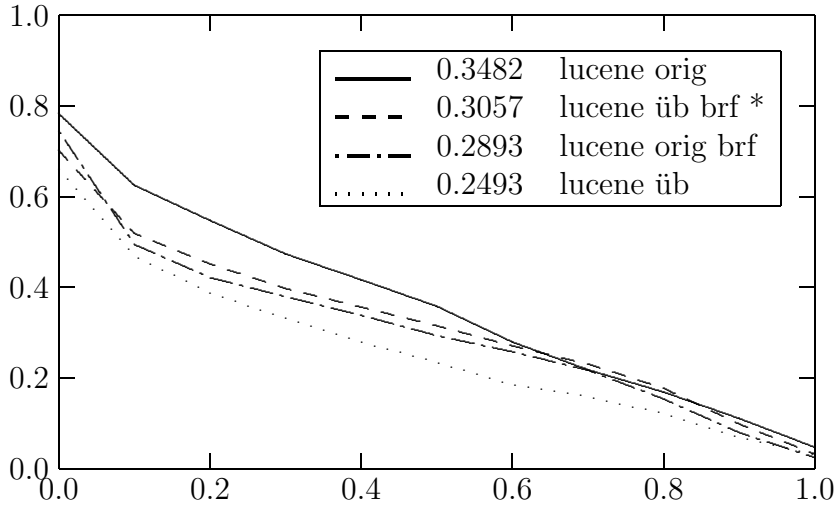

Abbildung 6.10: *Verschiedene Lucene–basierte multilinguale Runs.*

6.2.5 Multilingual

In den multilingual fusionierten Runs erweisen sich die originalen Anfragen den maschinell übersetzten als überlegen (Abbildung 6.10 und Abbildung 6.11). Die Anwendung von BRF hat auf die *Precision* von Runs mit originalen Anfragen einen negativen Effekt. Dagegen profitieren Runs mit übersetzten Anfragen von BRF. Die Trends aus den Einzelkollektionen setzen sich also in den gemeinsamen Listen fort.

Die erreichten Recall–Werte sind in Tabelle 6.7 aufgeführt. Die meisten rele-

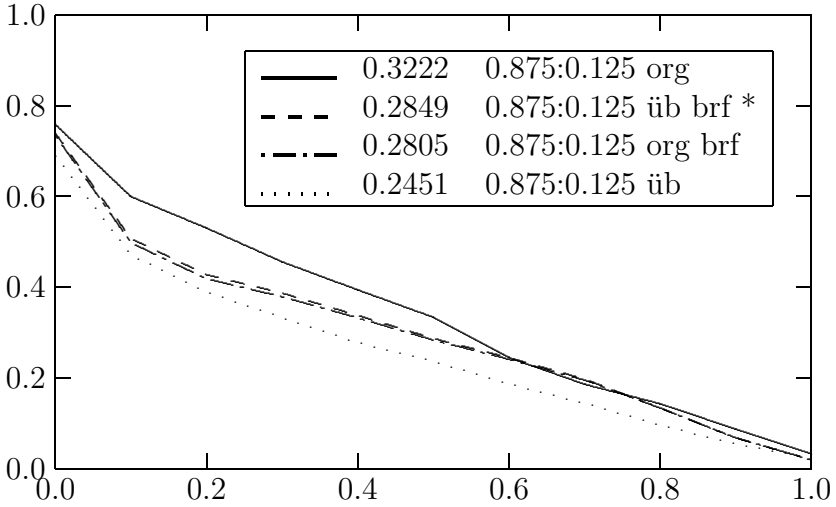

Abbildung 6.11: *Verschiedene fusionierte multilinguale Runs.*

vanten Dokumente werden in den Runs nachgewiesen, die auch schon die höchste *Precision* erzielt hatten. Trotzdem ist die beste Recallquote für den Run „lucene orig q no BRF“ mit etwa 71% mittelmäßig.

Weiterhin zeigt sich, dass die relativen Verluste immer deutlich über 20% liegen (Tabelle 6.8), für übersetzte Anfragen ohne BRF sogar um die 30%. Insbesondere die drei bislang verfolgten Topics 164, 181 und 197 verzeichnen in allen Konfigurationen überproportional hohe Verluste (Tabelle 6.9).

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multil.
$\sum D_r$	1006	946	1825	2368	6145	6145
0.875:0.125 * üb q BRF	951	816	1413	1967	5147	3944
0.875:0.125 üb q no BRF	957	830	1322	1979	5088	3607
0.875:0.125 orig q BRF	951	816	1413	1965	5145	3917
0.875:0.125 orig q no BRF	957	885	1562	2147	5551	4304
lucene * üb q BRF	953	846	1478	1954	5223	4137
lucene üb q no BRF	957	839	1366	1978	5140	3515
lucene orig q BRF	953	830	1455	1969	5207	3985
lucene orig q no BRF	957	902	1563	2147	5569	4388

Tabelle 6.7: *Recall der Runs 2003 offiziell wie auch inoffiziell.*

	absoluter Verlust	relativer Verlust
0.875:0.125 * üb q BRF	1203	23.23%
0.875:0.125 üb q no BRF	1481	29.11%
0.875:0.125 orig q BRF	1228	23.87%
0.875:0.125 orig q no BRF	1247	22.46%
lucene * üb q BRF	1086	20.79%
lucene üb q no BRF	1625	31.61%
lucene orig q BRF	1222	23.47%
lucene orig q no BRF	1181	21.21%

Tabelle 6.8: *Verluste beim Recall der Runs 2003.*

	Topic #164			Topic #181			Topic #197		
	R	$\sum D_r$	Rest	R	$\sum D_r$	Rest	R	$\sum D_r$	Rest
0.875:0.125 * mrg q BRF	260	110	55	814	593	358	543	465	247
rel. Verlust			50.00%			39.63%			46.88%
0.875:0.125 mrg q no BRF	260	117	41	814	589	318	543	509	289
rel. Verlust			64.96%			46.01%			43.22%
0.875:0.125 orig q BRF	260	110	55	814	593	358	543	465	247
rel. Verlust			50.00%			39.63%			46.88%
0.875:0.125 orig q no BRF	260	205	138	814	765	455	543	512	314
rel. Verlust			32.68%			40.52%			38.67%
lucene * mrg q BRF	260	111	41	814	572	382	543	439	204
rel. Verlust			63.06%			33.22%			53.53%
lucene mrg q no BRF	260	110	29	814	575	317	543	510	270
rel. Verlust			73.64%			44.87%			53.53%
lucene orig q BRF	260	110	50	814	591	354	543	465	244
rel. Verlust			54.54%			40.10%			47.53%
lucene orig q no BRF	260	205	141	814	765	445	543	509	346
rel. Verlust			31.22%			41.83%			32.02%

Tabelle 6.9: *Analyse der multilingualen Topics 2003.* $\sum D_r$ *bezeichnet die Summe der relevanten Dokumente aus den Einzelkollektionen; „Rest“ steht für die relevanten Dokumente, die nach dem Fusionieren in der mehrsprachigen Ergebnisliste noch nachgewiesen werden konnten.*

	Terme	Terme/Anfr.	Terme<4	ø Termlänge
Deutsch	725	12.08	24	7.58
Englisch	443	7.38	18	5.89
Französisch	838	13.97	158	5.54
Spanisch	723	12.05	71	6.01
Deutsch BRF	1925	32.08	80	8.67
Englisch BRF	1643	27.38	89	6.24
Franz. BRF	2038	33.97	512	5.44
Spanisch BRF	1923	32.05	176	6.93

Tabelle 6.10: *Termlänge der runs 2003. Die Anzahl der Terme für BRF hat um je 60×20 = 1200 Terme zugenommen, entsprechend ist die Anzahl der Terme pro Anfrage um 20 gestiegen.*

6.2.6 Fazit Ergebnisse

In diesem Fazit werden vor allem die Leistungen von MySQL und *Blind Relevance Feedback* betrachtet. Beide Faktoren hatten erheblichen Einfluss auf die Resultate.

MySQL war „out-of-the-box“ nicht konkurrenzfähig. Der wahrscheinlich einflussreichste Grund dafür war, dass bei der Volltextsuche Terme mit einer Länge von unter vier Buchstaben automatisch aus der Anfrage entfernt wurden. Da die Terme aber erst einheitlich „gestemmed“ wurden, bevor sie an die IRS übergeben wurden, sind relativ viele Wörter an dieser Grenze gescheitert.

Eine Untersuchung der Termlängen von Anfragen hat ergeben, dass vor allem französische, mit Abstrichen auch spanische, Anfragen viele kurze Terme enthalten (cf. Tabelle 6.10). Dies wird vor allem nach der Anfrageerweiterung durch BRF offensichtlich.

BRF hat insgesamt die gewünschten Performanzvorteile gebracht. Ein dabei immer wieder auftretendes Problem wird häufig salopp als „*garbage in, garbage out*“ bezeichnet. Damit soll ausgedrückt werden, dass inkorrekte Daten in den Kollektionen auch zu inkorrekten Daten bei der Termerweiterung führen. Ein

Beispiel dafür wären bei der deutschen Anfrageerweiterung für Topic #159 (*North Sea Oil Environment*) die Terme „wurdenwirkungsvolleeingriffebehindert“ und „fruchtbarkeitunderhohtesterblichkeitbeif“.

Eine weitere Fehlerquelle sind Terme, die aufgrund von Rechtschreibfehlern als „gute“ Deskriptoren identifiziert werden. So ist bei der spanischen Anfrage zu Topic #169(*Advent of the CD–Burner*) der Term „adudiocasett“ (anstatt „audiocasett“) hinzugefügt worden.

Die identisch verlaufenden Graphen für deutsche und französische Topics bei fusionierten Systemen mit BRF sind kurios. Auch bei den spanischen Topics gibt es nur einen Unterschied von 0.0025 *Precision*, der von Topic #200 herrührt. Trotz mehrfacher Kontrolle und Wiederholung der fraglichen Läufe war das Ergebnis immer das gleiche. Es scheint, dass durch BRF die Unterschiede der Übersetzungsqualität „ausgeglichen“ wurden. Dafür spräche auch, dass durch BRF originale Anfragen schlechter, maschinell übersetzte Anfragen besser wurden.

Kapitel 7

Abschlussbetrachtung

Die Teilnahme an CLEF 2003 hat die Grundannahmen über Fusion und *(Blind) Relevance Feedback* bestätigt. Diese Techniken haben sich hier als geeignet erwiesen, Retrievalleistungen zu verbessern. Gleichzeitig sind Problembereiche bei der Umsetzung aufgedeckt worden. So sind Datenbankzugriffe für das Absetzen von Anfragen merklich langsamer gewesen als Zugriffe einer Suchmaschine auf ihren Index. MySQL war außerdem nur eingeschränkt konfigurierbar. Lucene konnte dagegen in allen Bereichen verändert und den Bedürfnissen angepasst werden. Durch die spezielle Programmarchitektur ist die Software vielseitig einsetzbar. Mit überschaubarem Entwicklungsaufwand könnten Verfahren wie Kompositazerlegung, automatische Rechtstrunkierung (anstelle von *Stemming*) und Aufbau von n–gram–Indizes integriert werden.

Die Merge–Verfahren können weiter optimiert werden. In diesem Gebiet sind in der Fachwelt gegensätzliche Erfahrungen bei der Suche nach dem bestmöglichen Fusions–Algorithmus gemacht worden. Bislang hat sich über die verschiedenen Studien hinweg keine Alternative als überlegen bewiesen. Vor diesem Hintergrund sollte beachtet werden, dass es, wie in Kapitel 3 demonstriert, mehrere Fusionsschritte gibt. Hier ist auf jeder Ebene das optimale Vorgehen zu bestimmen. Durch eine geeignete Implementierung des MIMOR–Modells könnte diese Bestimmung, mit Trainingsdaten wie den CLEF–Kollektionen, in einem automa-

tisch lernenden Prozess erfolgen.

Anhang A

Wertetabellen

Recall–Stufen	Lucene	MySQL
0.0	0.8927	0.6063
0.1	0.6053	0.3722
0.2	0.5140	0.2769
0.3	0.4175	0.1536
0.4	0.3488	0.0428
0.5	0.2779	0.0014
0.6	0.1876	0.0000
0.7	0.1267	0.0000
0.8	0.0624	0.0000
0.9	0.0117	0.0000
1.0	0.0000	0.0000

Tabelle A.1: *Interpolierte Recall – Precision Werte 2001*

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multilingual
$\sum D_r$	856	1212	2130	2694		6892
Lucene	801	**1143**	1873	**2504**	**6321**	**5167**/6892
avg. P	0.4803	0.4042	0.3598	0.4752		0.2880
avg. DP	0.4941	**0.3853**	**0.4527**	**0.5318**		**0.3248**
MySQL	416	547	962	1287	3212	2873
avg. P	0.2247	0.1513	0.1366	0.2389		0.1037
avg. DP	0.2274	0.1600	0.2060	0.2540		0.1359
0.5:0.5	744	881	1561	2210	5396	3975
avg. P	0.3198	0.2517	0.2258	0.3332		0.1856
avg. DP	0.3657	0.2452	0.3164	0.3805		0.2206
0.8:0.2	803	1080	1848	2500	6231	4984
avg. P	0.4417	0.3319	0.3272	0.4444		0.2673
avg. DP	0.4787	0.3312	0.4272	0.5057		0.3094
0.9:0.1	802	1101	**1874**	2504	6281	5101
avg. P	0.4797	0.3632	0.3501	0.4676		0.2830
avg. DP	**0.4970**	0.3563	0.4493	0.5293		0.3248
0.85:0.15	**804**	1094	1865	2503	6266	5056
avg. P	0.4701	0.3465	0.3404	0.4565		0.2764
avg. DP	0.4913	0.3447	0.4405	0.5194		0.3189

Tabelle A.2: *Fusion 2001*

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multilingual
$\sum D_r$	856	1212	2130	2694		6892
Lucene	801	1143	1873	2504	6321	5167/6892
avg. P	0.4803	0.4042	0.3598	0.4752		0.2880
avg. DP	0.4941	0.3853	0.4527	0.5318		0.3248
MySQL	416	547	962	1287	3212	2873
avg. P	0.2247	0.1513	0.1366	0.2389		0.1037
avg. DP	0.2274	0.1600	0.2060	0.2540		0.1359

Tabelle A.3: *Einzelergebnisse 2001*

	Englisch	Franz.	Deutsch	Spanisch	$\sum D_r$	Multilingual
$\sum D_r$	821	1383	1938	2854		6996
Lucene	773	1197	1478	2475	5923	4454
avg. P	0.4715	0.3359	0.3374	0.4376		0.2876
avg. DP	0.4373	0.3534	0.3174	0.4913		0.2769
MySQL	371	668	731	1297	3067	2446
avg. P	0.1704	0.1506	0.1343	0.1962		0.0913
avg. DP	0.1550	0.1542	0.1160	0.2101		0.0951

Tabelle A.4: *Einzelergebnisse 2002*

Universität Neuchâtel	Universität Padua
Universität Tampere	Universität Oviedo
Johns Hopkins Universität	Universität Hagen
Xerox Research Centre Europe	MediaLab
Universität Berkeley	Universität Maryland
Universität Alicante	UNED
Carnegie Mellon Universität	DIOGENE
ITC–irst	Universität Süd–Kalifornien
Universität Amsterdam	Universität Montréal
Universität Hildesheim	DFKI Saarbrücken
LIC2M	Universität Sheffield
Universität Buffalo	Universität Surrey
SINAI, Spain	Ricoh
Universität Taipei	COLE
MIRACLE, Spain	SICS
Clairvoyance	Tagmatica
NII, Japan	Océ-Technologies
Universität Exeter	Hummingbird
Fondazione Ugo Bordoni	DLT
Universität Twente	

Tabelle A.5: *Teilnehmende Gruppen bei CLEF 2003*

Literaturverzeichnis

[Agosti et al. 2001] Agosti, Maristella; Crestani, F.; Pasi, G.(Hrsg.) (2001): *Lectures on Information Retrieval*. Berlin et al.: Springer.

[Attar und Fraenkel 1977] Attar, R.; Fraenkel, A. S. (1977): *Local feedback in full-text retrieval systems*. J. ACM 24(3). pp. 397417.

[Baeza-Yates und Ribeiro-Neto 1999] Baeza-Yates; Ribeiro-Neto (1999): *Modern Information Retrieval*. Harlow, England: Addison-Wesley Longman.

[Ballesteros und Croft 1997] Ballesteros, L.; Croft, W.B. (1997): *Phrasal Translation and Query Expansion Techniques for Cross-Language Information Retrieval*. In: Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-97), pp. 84-91, 1997.

[Belkin und Croft 1987] Belkin, N.J.; Croft, W.B. (1987): *Retrieval Techniques*. In: Williams, M. (Hrsg.) (1987): Annual Review of Information Science and Technology. New York: Elsevier Science Publishers. pp. 109-145.

[Borri und Peters 2003] Borri, Francesca; Peters, Carol (Hrsg.) (2003): *Results of the CLEF 2003 Cross-Language System Evaluation Campaign*. Working Notes for the CLEF 2003 Workshop 21-22 August, Trondheim, Norway. Band I. of the CLEF 2003 Workshop. Springer [LNCDS].

[Callan et al. 1992] Callan, J.P.; Croft, W.B.; Harding, S.M. (1992): *The inquery retrieval system*. In: DEXA 3: Proceedings of the Third International Conference on Database and Expert Systems Applications. Berlin: Springer. pp. 83-87.

[Carpineto et al. 2001] Carpineto, C.; de Mori, R.; Romano, G.; Bigi, B. (2001): *An Information-Theoretic approach to Automatic Query Expansion. ACM Transactions on Information Systems*. Volume 19, issue 1. pp. 1-27.

[Croft und Harper 1979] Croft, W.B.; Harper, D.J. (1979): *Using probabilistic models of document retrieval without relevance information*. Journal of Documentation 35, pp. 285295.

[Cuadra und Katter 1967] Cuadra, C.A.; Katter, R.V. (1967): *Experimental Studies of Relevance Judgments*. Report TM-3520, Final Report, Vol. 1. Santa Monica, California: System Development Corporation.

[Diekema 2003] Diekema, Anne: *Translation Events in cross-language IR: Lexical ambiguity, lexical holes, vocabulary mismatch, and correct translations*. Dissertation. Syracuse.

[Ferber 2003] Ferber, Reginald (2003): *Information Retrieval. Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. Heidelberg: dpunkt.verlag.

[Hölscher und Strube 1999] Hölscher, Christoph; Strube, Gerhard: *Searching on the Web: two types of expertise*. In: Proceedings of the 22nd Annual International CM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99). pp. 305-306. Berkeley: ACM.

[Fachgruppe IR 1996] Fachgruppe IR (1996): Fachgruppe Information Retrieval. http://ls6-www.informatik.uni-dortmund.de/ir/fgir/mitgliedschaft/brochure2.html

[Grossman und Frieder 1998] Grossman, D.A.; Frieder, O. (1998): *Information Retrieval - Algorithms and Heuristics*. Boston et al.:Kluwer Academic Publishers.

[Jones und Lam-Adesina 2001] Jones, G.J.F.; Lam-Adesina, A.M. (2001): *Exeter at CLEF 2001: Experiments with Machine Translation for Bilingual Retrieval*. In: [Peters 2001]

[Kluck et al. 2002] Kluck, Michael; Mandl, Thomas; Womser–Hacker, Christa (2002): *Cross–Language Evaluation Forum (CLEF): Europäische Initiative zur Bewertung sprachübergreifender Retrievalverfahren*. In: nfd Information – Wissenschaft und Praxis 53 (2). pp. 82-89.

[Korfhage 1997] Korfhage, Robert R. (1997): *Information Storage and Retrieval*. New York et al.: John Wiley & Sons, Inc.

[Kukuk 2003] Kukuk, Martin (2003): *Evaluierung kontextbasierter Retrievalsysteme im Axel Springer Verlag Infopool*. In: Schmidt, Ralph (Hrsg.): *Competence in Content. Proceedings der 25. Online–Tagung der DGI*. pp. 280-284.

[Lam-Adesina und Jones 2001] Lam-Adesina, A.M.; Jones, G.J.F. (2001): *Applying summarization techniques for term selection in relevance feedback*. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001) (New Orleans), pp. 19.

[Li 2002] Li, Qun (2002): *Anbindung von DB2 an die Implementierung des Fusionsansatzes MIMOR*. Magisterarbeit. Universität Hildesheim.

[Llopis und Muñoz 2003] Llopis, Fernando; Muñoz, Rafael (2003): *Cross-language experiments with IR-n system*. In: [Borri und Peters 2003]

[Longman 1996] Longman (1996): *Dictionary of Contemporary English*. London et al.: Longman

[Mandl 2003a] Mandl, Thomas (2003): *Web– und Multimedia–Dokumente*. In: nfd Information – Wissenschaft und Praxis 54. pp. 203-210

[Mandl und Womser-Hacker 2003b] Mandl, Thomas; Womser-Hacker, Christa (2003): *Proper Names in the Multilingual CLEF Topic Set*. In: [Borri und Peters 2003]

[McNamee und Mayfield 2003] McNamee, Paul; Mayfield, James (2003): *JHU/APL Experiments in Tokenization and Non-Word Translation*. In: [Borri und Peters 2003]

[Mayfield 2002] Mayfield, James (2002): *Information Retrieval*. http://www.clsp.jhu.edu/ws2002/preworkshop/mayfield.pdf

[Oard 1997] Oard, Doug: *Cross-Language Information Retrieval Defined*. http://raven.umd.edu/dlrg/clir/mlir_definition.html

[Peters 2001] Peters, Carol (Hrsg.) (2001): *Results of the CLEF 2001 Cross-Language System Evaluation Campaign*. Working Notes for the CLEF 2001 Workshop 3 September, Darmstadt, Germany.

[Peters 2002] Peters, Carol (Hrsg.) (2002): *Results of the CLEF 2002 Cross-Language System Evaluation Campaign*. Working Notes for the CLEF 2002 Workshop 19-20 September, Rome, Italy.

[Plödt 2003] Plödt, Alexandra (2003): *Multilinguales Information Retrieval: Linguistische Verfahren zur Anfrageoptimierung im Kontext des Cross-Language Evaluation Forum (CLEF)*. Magisterarbeit. Universität Hildesheim.

[Robertson und Sparck-Jones 1976] Robertson, S.E.; Sparck Jones, K. (1976): *Relevance weighting of search terms*. Journal of the American Society for Information Sciences, 27(3). pp. 129-146.

[Robertson 1991] Robertson, S.E. (1991): *On term selection for query expansion*. Journal of Documentation 46 4 pp. 59-364.

[Robertson et al. 1996] Robertson, S.E.; Walker, S.; Beaulieu, M.; Gatford, M.; Payne, A. (1996): *Okapi at Trec-4*. In: Harman, D. (Hrsg.): The Fourth Text REtrieval Conference (TREC-4). NIST Special Publication 500-236. pp. 182-191.

[Rocchio 1971] Rocchio, J.J. (1971): *Relevance feedback in information retrieval*. In: Salton, G. (Hrsg.) (1971): The SMART Retrieval System – Experiments in Automatic Document Processing. Englewood Cliffs, N.J.: Prentice Hall.

[Salton und McGill 1987] Salton, Gerard; McGill, Michael J. (1987): *Information Retrieval – Grundlegendes für Informationswissenschaftler*. Hamburg et al.: McGraw-Hill.

[Salton und Buckley 1988] Salton, G.; Buckley, C. (1988): *Term-weighting approaches in automatic retrieval*. Information Processing & Management, 24(5). pp. 513-523.

[Salton und Buckley 1990] Salton, G.; Buckley, C. (1990): *Improving retrieval performance by relevance feedback*. Journal of the American Society for Information Science. Band 41. pp. 288297.

[Savoy 2001] Savoy, Jacques (2001): *Report on CLEF-2001 Experiments*. In: [Peters 2001]

[Savoy 2002] Savoy, Jacques (2002): *Report on CLEF-2002 Experiments: Combining Multiple Sources of Evidence*. In: [Peters 2002]

[Savoy 2003] Savoy, Jacques (2003): *Report on CLEF-2003 Multilingual Tracks*. In: [Borri und Peters 2003]

[Sheridan und Ballerini 1996] Sheridan, Paraic; Ballerini, Jean Paul (1996): *Experiments in Multilingual Information Retrieval using the SPIDER system*. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996). pp. 58-65.

[Sheridan et al. 1997] Sheridan, Paraic; Braschler, Martin; Schäuble, Peter (1997): *Cross-Language Information Retrieval in a Multilingual Legal Domain*. In: Peters, Carol; Thanos, Costantino(Hrsg.) (1997): Research and Advanced Technology for Digital Libraries. First European Conference (ECDL'97). pp. 253-268.

[Si und Callan 2003] Si, Luo; Callan, Jamie (2003): *A Semi–Supervised Learning Method to Merge Search Engine Results*. ACM Transactions on Information Systems. Vol. 21, No 4.

[Singhal et al. 1996] Singhal, A.; Salton, G.; Mitra, M.; Buckley, C. (1996): *Document Lenght Normalization*. Information Processing and Management 32(5), pp. 619-633.

[Singhal 1997] Singhal, A.K. (1997): *Term Weighting Revisited*. Dissertation. Cornell University, Ithaca, NY, Department of Computer Sciences.

[van Rijsbergen 1979] Van Rijsbergen, K. (1979): *Information Retrieval*. London(UK): Butterworths.

[Womser–Hacker 1997] Womser–Hacker, Christa (1997): *Das MIMOR-Modell. Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval*. Habilitationsschrift. Universität Regensburg, Informationswissenschaft.

[Womser–Hacker 2002] Womser–Hacker, Christa (2002): *Multilingual Topic Generation within the CLEF 2001 Experiments*. In: [Peters 2002]

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Hilfsmittel verfasst habe.

René A. Hackl Hildesheim, 15.01.2004