Technische Universität Chemnitz

Fakultät für Informatik

Professur Medieninformatik

Diplomarbeit

von

Thomas Wilhelm

Matrikelnr. 24754

Entwurf und Implementierung eines Frameworks zur Analyse und Evaluation von Verfahren im Information Retrieval

Prüfer: Prof. Dr. Maximilian Eibl
Gutachter: Dipl. Inf. Jens Kürsten

Chemnitz, 25. April 2008

Wilhelm, Thomas
Entwurf und Implementierung eines Frameworks zur Analyse und Evaluation von Verfahren im Information Retrieval
Diplomarbeit, Fakultät für Informatik
Technische Universität Chemnitz, April 2008

Kurzfassung

Diese Diplomarbeit führt kurz in das Thema Information Retrieval mit den Schwerpunkten Evaluation und Evaluationskampagnen ein. Im Anschluss wird anhand der Nachteile eines vorhandenen Retrieval Systems ein neues Retrieval Framework zur experimentellen Evaluation von Ansätzen aus dem Information Retrieval entworfen und umgesetzt.

Die Komponenten des Frameworks sind dabei so abstrakt angelegt, dass verschiedene, bestehende Retrieval Systeme, wie zum Beispiel Apache Lucene oder Terrier, integriert werden können. Anhand einer Referenzimplementierung für den ImageCLEF Photographic Retrieval Task des ImageCLEF Tracks des Cross Language Evaluation Forums wird die Funktionsfähigkeit des Frameworks überprüft und bestätigt.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter der Angabe der Quellen kenntlich gemacht.

Chemnitz, 25. April 2008

Thomas Wilhelm

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Liste der Abkürzungen

API	Application Programming Interface (Programmierschnittstelle)
CLEF	Cross Language Evaluation Forum
FB	Relevance Feedback
GMAP	Geometric Mean Average Precision
GUI	Graphical User Interface (Grafische Benutzerschnittstelle)
IAPR	International Association for Pattern Recognition
IR	Information Retrieval
MAP	Mean Average Precision
MPEG	Moving Picture Experts Group
QE	Query Expansion
TREC	Text Retrieval Conference
UML	Unified Modelling Language
XML	Extensible Markup Language

1 Einführung

Die Produktions- und Verkaufszahlen für Digital- und Analog-Kameras der Camera & Imaging Products Association (CIPA) für das Jahr 2007 (siehe [CIPA 2008]) zeigen deutlich, dass die analoge Fotografie von der digitalen Fotografie verdrängt wurde. 100,37 Millionen Digital-Kameras stehen nur noch 0,71 Millionen analoge Kameras gegenüber. Dies hat für die Archivierung von Fotos weitreichende Folgen. In das klassische Fotoalbum kommen nur noch gelungene Fotos, die entweder gleich zu Hause ausgedruckt oder von einem Fotolabor auf professionelles Fotopapier gedruckt werden. Dagegen sammeln sich auf den digitalen Datenträgern unzählig viele Fotos. Speicher ist billig und alle Urlaubsbilder passen auf eine einzige CD oder DVD – meist inklusive der misslungenen Schnappschüsse.

Diese Datenflut macht eine effiziente Verwaltung notwendig. Entwickler und Unternehmen haben diese Notwendigkeit bereits erkannt und versuchen Lösungen zu entwickeln um diese Daten für den Nutzer zugänglich zu machen. Einfache Lösungsansätze bieten bereits die Möglichkeit, Bilder nach unterschiedlichen Kriterien, zum Beispiel nach Datum oder nach vom Nutzer eingegebenen Informationen sortiert, abzulegen. Diese Werkzeuge können die in den Bildern bereits eingebetteten Metadaten (zum Beispiel „Exif“) auslesen und anzeigen. Darüber hinaus kann der Nutzer eigene Metadaten für die Bilder erfassen und speichern, vergleichbar mit den Bildbeschriftungen in Fotoalben.

1.1 Information Retrieval

Die Autoren Salton und McGill verstehen unter dem Begriff Information Retrieval folgendes:

> “Information Retrieval (IR) is concerned with the representation, storage, organization, and accessing of information items. In principle no restriction is placed on the type of item handled in information retrieval.” [Salton&McGill 1983]

Im Buch *An Introduction to Information Retrieval* von Manning, Raghavan und Schütze wird Information Retrieval wie folgt definiert:

> “Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfy an information need from within large collections (usually stored on computers).” [Manning 2008]

Dies zeigt, wie unterschiedlich der Begriff des Information Retrieval verstanden werden kann: Salton und McGill konzentrieren sich auf die Repräsentation und den Zugriff auf Informationen, wogegen Manning et. al. sich insbesondere mit der Befriedigung eines Informationsbedürfnisses befassen. Beide wollen aber die Art der Information nicht einschränken. Das schließt neben reinen Texten auch jegliche Form von multimedialen Informationen ein, wie unter anderem Bilder, Tonaufzeichnungen und Videos.

Um diese Informationen für den Nutzer zugänglich zu machen, benötigt dieser ein Information Retrieval System. Ein Information Retrieval System ist nach [Rijsbergen 1979] eine Software, die Datenbestände auf der Grundlage von unscharfen, vom Benutzer aufgrund seines Informationsbedürfnisses angegebenen Suchkriterien durchsucht und daraus die Dokumente, die den Suchkriterien am besten oder auch nur teilweise entsprechen, anzeigt. Dabei ist es unsicher, ob die gefundenen Dokumente die vom Benutzer gewünschte Information tatsächlich enthalten, weil einerseits die Suchkriterien das Informationsbedürfnis nicht vollständig beschreiben und andererseits nicht der gesamte Inhalt der Dokumente für die Suche zugänglich ist.

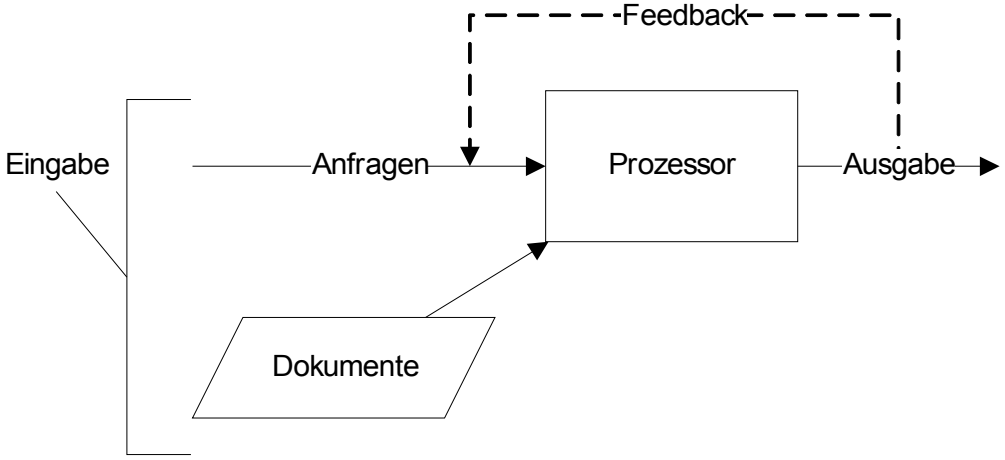


Abbildung 1: Typisches Information Retrieval System nach [Rijsbergen 1979]

Nach [Rijsbergen 1979] besteht ein Information Retrieval System vereinfacht aus einer Eingabe, einer Verarbeitung und einer Ausgabe. Die Eingabe umfasst die Dokumente, die vom System in eine interne Repräsentation umgewandelt werden, und die Anfragen, die vom Nutzer in einer vom System geforderten Art und Weise formuliert werden müssen. Bei der Verarbeitung werden die Anfragen und die Dokumente miteinander verglichen und nach ihrer Ähnlichkeit sortiert in eine Ergebnisliste eingetragen. Nach Ausgabe der Ergebnisse kann der Nut-

zer anhand dieser seine Anfrage reformulieren, um sein Informationsbedürfnis für das System besser zu beschreiben, indem er zum Beispiel Suchkriterien hinzufügt um die Ergebnisse zu reduzieren oder Suchkriterien entfernt um die Anzahl der Ergebnisse zu erhöhen.

In den folgenden Abschnitten werden einige zentrale Retrievalmodelle kurz vorgestellt.

1.1.1 Boolesches Retrieval

Beim Booleschen Retrievalmodell wird jedes Dokument durch eine Menge von Termen repräsentiert. Der Schritt der Anfrageformulierung geschieht durch das Verknüpfen dieser Terme mit den booleschen Operatoren AND (Konjunktion), OR (Disjunktion) und NOT (Negation). Ein Matching findet statt, indem die Dokumente bzw. die Menge der Terme, die ein Dokument repräsentieren, in die Anfrage eingesetzt werden und auf den Wahrheitsgehalt hin untersucht werden. Dokumente, die die gesuchten Terme in der angegebenen Art und Weise enthalten, werden in die Ergebnisliste aufgenommen.

Das booleschen Retrievalmodell ist ein einfaches Retrievalmodell. Es unterstützt keine direkte Bewertung der einzelnen Ergebnisse, so dass keine nach Relevanz sortierte Ergebnisliste erstellt werden kann. Unscharfe Formulierungen durch den Benutzer sind nicht möglich.

1.1.2 Vektorraum-basiertes Retrieval

Das Vektorraummodell wurde von G. Salton im Rahmen des SMART-Projektes [Salton 1971] entwickelt und von Raghavan und Wong [Raghavan 1986] weiterentwickelt. Die grundlegende Idee ist, die Dokumente und die Abfragen (Queries) als Punkte in einem Vektorraum zu beschreiben, wobei die Term-Vektoren orthogonal und normiert sind. Beim Retrieval werden die Dokument-Vektoren und die Abfrage-Vektoren über ein Ähnlichkeitsmaß mit einander verglichen. Als Ähnlichkeitsmaß werden Metriken, wie zum Beispiel das Skalarprodukt, das Cosinus-Maß oder der Dice-Koeffizient [Rijsbergen 1979], eingesetzt. Da alle Term-Vektoren normiert sind, kann über den Betrag des Vektors eine Gewichtung statt finden.

Eine oft genutzte Gewichtung ist die Term-Frequenz – Inverse Dokument-Frequenz (TF-IDF). Die Term-Frequenz für ein Dokument gibt die Häufigkeit für das Auftreten des Terms

in diesem Dokument an. Die Inverse Dokument-Frequenz ist ein Maß für die Spezifität eines Terms. [Manning 2008]

$$idf_t = \log \frac{N}{df_t}$$

$$tfidf_{t,d} = tf_{t,d} \times idf_t$$

$$\text{Score}(q, d) = \sum_{t \in q} tfidf_{t,d}$$

d ist ein Dokument, t ist ein Term, q ist eine Query bzw. Suchanfrage

df_t ist Dokumentfrequenz von t bzw. die Anzahl Dokumente, die t enthalten

$tf_{t,d}$ ist die Termfrequenz von t in d bzw. die Häufigkeit von t in d

1.1.3 Probabilistisches Retrieval

Unter dem Begriff des probabilistischen Retrieval wird eine Menge von Modellen zusammengefasst, die das Ranking der Dokumente anhand von verschiedenen Wahrscheinlichkeiten vornehmen.

Prominente Beispiele für diese Klasse von Modellen sind unter anderem naive Bayes-Klassifikatoren, der PageRank und Okapi BM25. Diese werden in den folgenden Unterkapiteln näher betrachtet.

Weitere Bereiche des probabilistischen Retrievals sind:

- Unsicheres Schlussfolgern (siehe [Wong 1995])
- Divergence from Randomness (siehe [Amati 2002])
- Latent Dirichlet Allocation (siehe [Blei 2003])

1.1.3.1 Naive Bayes-Klassifikatoren

Ein Bayes-Klassifikator ordnet einen Term t einer Klasse C_i aus einer Menge von Klassen zu [Lewis 1998]. Die Grundlage dafür ist das Bayestheorem für Rechnungen mit bedingten Wahrscheinlichkeiten:

$$P(C_i|t)=\frac{P(t|C_i)\cdot P(C_i)}{P(t)}$$

$P(C_i)$ ist die Wahrscheinlichkeit für die Klasse C_i

$P(t)$ ist die Wahrscheinlichkeit für den Term t

$P(C_i|t)$ ist die Wahrscheinlichkeit für die Klasse C_i, wenn der Term t gegeben ist

$P(t|C_i)$ ist die Wahrscheinlichkeit für den Term t, wenn die Klasse C_i gegeben ist

Wenn die Annahme getroffen wird, das alle Terme t unabhängig von einander auftreten, spricht man von einem *naiven* Bayes-Klassifikator. Dies hat den Vorteil, dass der Bayes-Klassifikator schnell bestimmt werden kann, auch für sehr große Datenbestände.

Trotz des naiven Ansatzes liefern die Bayes-Klassifikatoren gute Ergebnisse, da besonders im Bereich des Information Retrieval die absoluten Werte für die Wahrscheinlichkeiten nicht von Bedeutung sind, sondern ihre Reihenfolge.

Die Wahrscheinlichkeit, dass ein Dokument D zur Klasse C_i gehört, ergibt sich aus den Einzelwahrscheinlichkeiten der im Dokument enthaltenen Terme:

$P(D|C_i)=\prod_{t\in D} P(t|C_i)$ ist die Wahrscheinlichkeit, dass, wenn wir uns in C_i befinden, das Dokument D auftritt bzw. vorhanden ist.

Die Klasse C_D, zu der Dokument D am wahrscheinlichsten gehört, wird nach der Maximum-A-Posteriori-Methode bestimmt:

$$C_D=\text{argmax}_c P(C_i=c)\prod_{t\in D} P(t|C_i=c)$$

C_D ist die Klasse, die Dokument D am wahrscheinlichsten enthält.

1.1.3.2 PageRank

Der von den Google-Gründern Lawrence Page und Sergey Brin vorgestellte PageRank [Page 1998] basiert auf der Wahrscheinlichkeit, dass ein beliebiger Nutzer des Internets eine bestimmte Seite über einen Hyperlink besucht.

Es wird ein gerichteter Graph modelliert, dessen Knoten Webseiten repräsentieren und dessen Kanten für Links zwischen diesen Webseiten stehen. Bei einer Erhebung im Januar 2005 [Gulli 2005] wurde die Anzahl der indizierbaren Dokumente im World Wide Web auf 11,5 Milliarden Seiten geschätzt. Die Anzahl der Hyperlinks zwischen diesen Dokumenten ist um ein Vielfaches höher.

Ein Knoten (Webseite) in diesem gerichteten Graphen besitzt eine bestimmte Anzahl von Eingangs- und Ausgangskanten. Die Eingangkanten werden als Backlinks und die Ausgangskanten als Forwardlinks bezeichent. Einfach formuliert ergibt sich die Relevanz einer Webseite aus der Summe der Relevanzen der Backlinks. Es gibt jedoch keine Möglichkeit die Backlinks einer Webseite direkt aus dieser zu bestimmen. Im Gegensatz dazu ist eine Extraktion der Forwardlinks trivial. Deshalb wird bei der Berechnung des PageRanks das Modell eines sich zufällig durch das World Wide Web bewegenden Nutzers verwendet. Dieses Modell (Random Surfer Model) basiert auf Random Walks in Graphen mit zusätzlichem Dämpfungsfaktor.

Die vereinfachte Formel ohne Dämpfungsfaktor für den PageRank lautet:

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{|F_v|} \quad \text{(vereinfachter PageRank)}$$

B_x ist die Menge der Backlinks für die Webseite x

F_x ist die Menge der Forwardlinks für die Webseite x

c ist ein Faktor zur Normalisierung

Eine Webseite, die von außen referenziert ist, jedoch selbst keine Referenz auf externe Webseite besitzt führt zu einem sich wiederholenden Verfolgen der gleichen, internen Links. In diesem Fall würde der Random Walk sich in einer Endlosschleife auf dieser Webseite bewegen. Um dies zu verhindern wird ein in näher [Page 1998] beschriebener Dämpfungsfaktor eingesetzt.

Die Bedeutung des PageRank wird von Google selbst wie folgt beschrieben:

> „The heart of our software is PageRank™, a system for ranking web pages developed by our founders Larry Page and Sergey Brin at Stanford University. [...] PageRank continues to play a central role in many of our web search tools.“ [Google 2008]

1.1.3.3 Okapi BM25

Okapi ist ein experimentelles Text Retrieval System. Im Rahmen dieses Systems entstanden verschiedene Termgewichtungsfunktionen mit dem Prefix „BM“. BM steht dabei für „Best Match“. Zu diesen Termgewichtungsfunktionen gehören BM0 (alle Terme sind gleich gewichtet), BM1 (ähnlich TF/IDF), BM11, BM15 und BM25. Die Termgewichtungsfunktionen mit höheren Nummern stellen dabei Weiterentwicklungen dar, die meist eine höhere Komplexität als die zuvor entwickelten Gewichtungsfunktionen aufweisen.

$$\text{Score}(q,d)=\sum_{t\in q} w^{(1)}\frac{(k_1+1)tf_{t,d}}{K+tf_{t,d}}\frac{(k_3+1)tf_{t,q}}{k_3+tf_{t,q}}+k_2\cdot|Q|\cdot\frac{avdl-dl}{avdl+dl} \quad \text{(BM25)}$$

d ist ein Dokument, t ist ein Term, q ist eine Query bzw. Suchanfrage

$K=k_1((1-b)+b\cdot dl/avdl)$

k_1, b, k_2, k_3 und k_4 sind Parameter, die von der Art der Queries und der Datenbank abhängen

$tf_{t,d}$ ist die Termfrequenz von t im Dokument d

$tf_{t,q}$ ist die Termfrequenz von t in Query q

dl ist die Dokumentenlänge

$avdl$ ist die durchschnittliche Dokumentenlänge über alle Dokumente

$w^{(1)}$ ist das Robertson/Sparck Jones Gewicht:

$$w^{(1)}=\log\frac{(r_t+0.5)(N-n_t-R+r_t+0.5)}{(n_t-r_t+0.5)(R-R+0.5)} \quad \text{(siehe [Robertson 1995])}$$

r_t ist die Anzahl relevanter Dokumente für den Term t

R ist die Gesamtanzahl relevanter Dokumente

n_t ist die Anzahl indizierter Dokumente für den Term t

N ist die Gesamtanzahl indizierter Dokumente

Die Termgewichtungsfunktion BM25 hat bei mehreren Gelegenheiten, unter anderem bei mehreren TREC-Kampagnen, ihre Überlegenheit gegenüber der Standard-Termfrequenz/In-

verse-Dokumenzfrequenz-Gewichtung gezeigt und wurde deshalb von mehreren Gruppen in ihre Experimente aufgenommen.

1.2 Evaluation

Unter Evaluation ist die Bewertung der Leistungsfähigkeit und/oder des Wertes eines Systems, eines Prozesses oder eines Produktes zu verstehen. Für Information Retrieval Systeme bedeutet das, dass die Leistungsfähigkeit verschiedener Systeme auf der Basis von Kennzahlen unter gleichen Voraussetzungen zu beurteilen ist. Dies ist notwendig, da der Nutzer eines Information Retrieval Systems nur vage Suchkriterien angibt, die Objekte nicht vollständig beschreiben sind und als Folge dessen verschiedene Systeme verschiedene Treffer liefern.

Um ein Information Retrieval System evaluieren zu können benötigt man

- eine Dokumentensammlung (Korpus),
- eine Zusammenstellung von Suchanfragen, die die Informationsbedürfnisse von potentiellen Nutzern widerspiegeln und
- eine Aufstellung von relevanten und nicht-relevanten Suchanfrage-Dokument-Paaren.

Um den Evaluationsprozess zu unterstützen wurden verschiedene Evaluationskampagnen ins Leben gerufen, die den Teilnehmern die benötigten Daten zur Verfügung stellen.

In den folgenden Abschnitten werden zuerst die gebräuchlichsten Kennzahlen beschrieben und im Anschluss einige wichtige Evaluationskampagnen vorgestellt.

1.2.1 Kennzahlen

Zur Evaluation von Information Retrieval Systemen gibt es eine Vielzahl von Kennzahlen zur Beschreibung der Leistungsfähigkeit. Die meisten dieser Kennzahlen basieren auf den von [Kent 1955] beschriebenen Maßen Precision (Genauigkeit) und „Relevance“, was später in Recall (Vollständigkeit) umbenannt wurde.

Zur genauen Beschreibung von Precision und Recall werden folgende Mengen definiert:

- R ist die Menge aller relevanten Dokumente und $\bar{R}$ die komplementäre Menge.
- S ist die Menge der gefundenen Dokumente und $\bar{S}$ die komplementäre Menge.

- $t = R \cap S$ ist die Menge gefundener, relevanter Dokumente.

Die Precision gibt Auskunft über die Genauigkeit des Suchergebnisses, hat einen Wertebereich von 0 bis 1 bzw. 0% bis 100% und ist definiert durch:

$$\text{Precision} = \frac{|t|}{|S|}$$

Der Recall gibt die Vollständigkeit des Suchergebnisses an, also wie viele der relevanten Dokumente gefunden wurden, hat den gleichen Wertebereich wie die Precision und wird durch folgende Formel beschrieben:

$$\text{Recall} = \frac{|t|}{|R|}$$

Es ist offensichtlich, das ein Recall von 1 einfach durch die Betrachtung aller Dokumente eines Korpus erreicht werden kann, dabei sinkt jedoch die Precision gegen 0. Deshalb werden Precision und Recall stets in Verbindung betrachtet. Bei steigendem Recall (Anzahl gefundener, relevanter Dokumente) sinkt meist die Precision (Anteil relevanter Dokumente am Ergebnis). Dabei gibt es zwei Extreme, die eintreten können: Wenn Recall = 0 und Precision = 0 sind, so ist keines der gefundenen Dokumente relevant. Wenn Recall = 1 und Precision = 1 sind, so sind alle gefundenen Dokumente relevant und alle relevanten Dokumente wurden gefunden.

Diese Beziehung wird im Recall-Precision-Diagramm veranschaulicht. Im Recall-Precision-Diagramm wird auf der Y-Achse die Precision eingetragen, auf der X-Achse der Recall. Die einzelnen Punkte werden mit den oben angegebenen Formeln berechnet, wobei die Suchergebnisse einzeln betrachtet werden. Das heißt, die Menge der zu betrachtenden Suchergebnisse besteht zunächst aus nur einem Element, dem am besten bewerteten Dokument. Nun werden Recall und Precision berechnet. Bei jedem weiteren Schritt wird ein weiteres Dokument aus der Ergebnis-Liste zu dieser Menge hinzugefügt, Recall und Precision werden neu berechnet und falls sich der Recall zum vorherigen Schritt geändert hat (Recall-Punkt), in das Diagramm eingetragen.

Aus Precision und Recall können nun weitere Kennzahlen abgeleitet werden.

- Die „Precision at n“ (P n) gibt die Precision bei einer Suchergebnismenge mit n gefundenen Dokumenten an, also $|S|=n$.

  Bevorzugte Werte für n sind:

  - $n=20$ (P20), in Anlehnung an das Nutzerverhalten und die Ergebnis-Präsentation von Google[1], da auf der einer Ergebnis-Seite bei Standardkonfiguration 10 Ergebnisse angezeigt werden und der Nutzer im Allgemeinen nach Recherche der ersten beiden Seiten entweder seine Suche abbricht oder die Suchanfrage neu formuliert.
  - $n=|R|$, bei der so genannten R-Precision wird die Anzahl der relevanten Dokumente als n verwendet.

- Die „Average Precision“ (AP) ist der Durchschnitt der Precision über alle Recall-Werte. Wenn nicht alle relevanten Dokumente in der Menge der gefundenen Dokumente vorhanden sind, wird für diese Recall-Werte eine Precision von 0 angenommen.

$$\mathrm{AP}=\frac{\sum_{s\in S}\text{Precision at } s}{|R|}$$

Bei der Evaluierung von Information Retrieval Systemen werden meist mehrere Suchanfragen untersucht. Um diese Systeme besser vergleichen zu können werden die einzelnen Ergebnisse der Suchanfragen zu weiteren Kennzahlen kombiniert, die das Retrieval System bzw. dessen Konfigurationen charakterisieren.

Für die meisten Kennzahlen ist es ausreichend, das arithmetische Mittel über die einzelnen Suchanfragen zu bestimmen. So ist zum Beispiel das arithmetische Mittel der Average Precision als „Mean Average Precision“ (MAP) ein Standard-Performance-Maß für Retrieval Systeme.

$$\mathrm{MAP}=\frac{1}{|Q|}\cdot\sum_{q\in Q}\mathrm{AP}(q) \quad \text{mit } Q = \text{Menge der Suchanfragen}$$

1 http://www.google.com/ - Google

Das arithmetische Mittel wird häufig auch für die anderen Kennzahlen, wie zum Beispiel die Precision at 20 oder die R-Precision bestimmt.

Durch die arithmetische Mittlung fallen einzelne, schlechte Suchanfragen kaum ins Gewicht. Deshalb ist neben der MAP eine weitere Kennzahl nötig, um die Robustheit von Retrieval Systemen zu beschreiben - die so genannte Geometric Mean Average Precision (GMAP). Sie ist das geometrische Mittel der AP der einzelnen Suchanfragen und misst den sehr schlechten Suchanfragen eine höhere Bedeutung zu, als die MAP. Es reicht aus, wenn eine Suchanfrage kein relevantes Dokument liefert, damit die GMAP = 0 ist.

$$\mathrm{GMAP} = \sqrt[|Q|]{\prod_{q \in Q} \mathrm{AP}(q)} \quad \text{mit } Q = \text{Menge der Suchanfragen}$$

Eine weitere Möglichkeit zur übersichtlichen Darstellung von numerischen Daten innerhalb eines Intervalls ist der Box-Whisker-Plot (auch kurz Box-Plot) nach [Tukey 1977]. Ein Box-Plot gibt eine grafische Übersicht über die aufgetretenen Werte, deren Mittelwert und Median und die Verteilung im Intervall.

Die Darstellung eines Box-Plots ergibt sich aus den Kennzahlen, die anhand der gegebenen Werte bestimmt werden. Zu Beginn sortiert man die Werte und bestimmt dann die folgenden Kennzahlen:

Median (Zentralwert) – halbiert die Stichprobe bzw. Zahlenreihe

Mittelwert (Durchschnitt) – der arithmetische Mittelwert

Erste Quartile (Q1) – das erste Viertel der in 4 Teilen zerlegten Stichprobe

Dritte Quartile (Q3) – das dritte Viertel der in 4 Teilen zerlegten Stichprobe

Interquartilsabstand (IQR) – das Intervall zwischen Q1 und Q3

Das Intervall Q1 bis Q3 wird grafisch als Rechteck (Box) dargestellt. Diese Box wird durch den Median als senkrechte Linie in zwei verbundene Rechtecke unterteilt. Die links und rechts aus der Box ragenden Linien sind die Whisker. Sie repräsentieren die Werte, die außerhalb des Intervalls von Q1 bis Q3 liegen. Die maximale Länge der Whisker ist das Anderthalbfache des Interquartilsabstandes, das heißt des Abstandes zwischen Q1 und Q3. Außerhalb der Whisker liegen die als Kreuz gekennzeichneten Ausreißer.

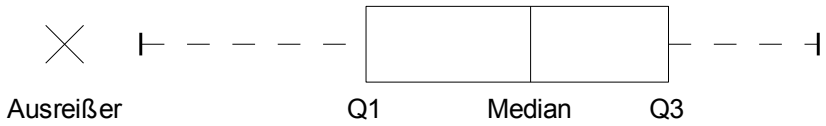


Abbildung 2: Skizze eines Box-Whisker-Plots

1.2.2 Evaluationskampagnen

Das Vorhandensein von Kennzahlen und Datensammlungen führt nicht automatisch zu einer Vergleichbarkeit von verschiedenen Systemen. Dafür wird eine gemeinsame Datenbasis benötigt, die durch die verschiedenen Evaluationskampagnen bereit gestellt wird. Die Datenbasis im Rahmen einer Kampagne umfasst neben einem Korpus auch die Suchanfragen und die Relevanzbewertungen.

Den Anfang machte in den 60er Jahren das Cranfield Projekt. In diesem Projekt wurde untersucht, welche Verfahren am besten zur Evaluation von Information Retrieval Systemen geeignet sind und welche Methode am besten geeignet ist um die nötigen Relevanzbewertungen zu erheben. [Cleverdon 1970]

Nach Abschluss des Cranfield Projektes sind im Zuge des SMART (Salton's Magic Automatic Retriever of Text) Information Retrieval Systems [Salton 1971] und anderer Projekte Test-Kollektionen entstanden, bis 1992 durch das National Institute of Standards and Technology (NIST) die Text Retrieval Conference (TREC) ins Leben gerufen wurde. [Saracevic 1995] Die TREC wird bis heute mit stets wachsenden Teilnehmerzahlen durchgeführt.

Für den asiatischen Raum wurde 1997 das NTCIR Test Collection Project gegründet, welches ähnlich wie TREC Test-Kollektionen zur Verfügung stellt und Workshops abhält. Neben Englisch liegt hier der Schwerpunkt auf den asiatischen Sprachen Chinesisch und Japanisch. [Kando 2001]

Das Cross-Language Evaluation Forum (CLEF) entstand 2001 als ein europäischer Ableger der TREC, der sich speziell mit sprachübergreifendem Information Retrieval beschäftigt.

Andere Kampagnen sind zum Beispiel die Music Information Retrieval Evaluation Exchange[2] (MIREX), das Forum for Information Retrieval Evaluation[3] (FIRE) und das Russion Information Retrieval Evaluation Semiar[4]

1.2.2.1 Text Retrieval Conference (TREC)

Organisiert vom National Institute of Standards and Technology (NIST) und gesponsort vom Ministerium für Verteidung der Vereinigten Staaten, ist die Text Retrieval Conference eine der bedeutendsten Evaluationskampagnen. [Fuhr 2004]

Zentrale Ziele der Text Retrieval Conference [TREC 2008] sind

- die Forschung im Bereich des Information Retrieval für großen Datenbeständen zu fördern,
- die Kommunikation zwischen Forschungseinrichtungen, Industrie und staatlichen Einrichtungen zu verbessern,
- die Entwicklung kommerzieller Produkte auf Basis von Forschungsergebnissen zu beschleunigen und
- einen Schwerpunkt auf die praktische Retrieval-Probleme des Alltags zu legen.

Die Text Retrieval Conference unterteilt sich in verschiedene Aufgabenbereiche, den so genannten Tracks. Die Tracks variieren von Jahr zu Jahr. Entsprechend der Zielstellung von TREC können neue Tracks entstehen und bestehende Tracks, die ihr Ziel erreicht haben, entfallen. Unter Umständen werden auch einzelne Trecks, wie zum Beispiel im Fall von TRECVID[5] oder CLEF, in eigenständige Kampagnen ausgelagert.

Im Jahr 2008 stehen fünf Tracks zur Verfügung:

- **Blog Track** (Erforschung des Suchverhaltens im Umfeld von Internet Blogs)
- **Enterprise Track** (Information Retrievalaufgaben in Unternehmen)

2 http://www.music-ir.org/mirexwiki/index.php/Main_Page - Main Page MIREX2008
3 http://www.isical.ac.in/~clia/ - Forum for Information Retrieval Evaluation (FIRE)
4 http://romip.narod.ru/en/ - ROMIP: Russian Information Retrieval Evaluation Seminar
5 http://www.itl.nist.gov/iaui/894.02/projects/trecvid/ - TREC Video Retrieval Evaluation Home Page

- **Legal Track** (Retrieval-Technologien zur Unterstützung von Anwälten)
- **Million Query Track** (Untersuchung der Hypothese, das viele, unvollständig bewertete Topics besser als Pooling ist)
- **Relevance Feedback Track** (Soll die Rahmenbedingungen zur besseren Untersuchung von Relevance-Feedback-Methoden schaffen)

1.2.2.2 Cross-Language Evaluation Forum (CLEF)

Im Rahmen des Cross-Language Evaluation Forums (CLEF) werden jährlich neue Test-Szenarien für unterschiedliche Aufgabebereiche, so genannte Tracks, angeboten. Das Cross-Language Evaluation Forum ist aus dem Cross-Language Track der Text Retrieval Conference (TREC) hervorgegangen und wird seit dem Jahr 2000 eigenständig organisiert und durchgeführt.

Zu jeder CLEF Kampagne wird ein Workshop abgehalten, auf dem die Teilnehmer ihre Ergebnisse vergleichen und diskutieren können. Außerdem werden Anregungen und Verbesserungen der Kampagne für das nächste Jahr gesammelt und besprochen.

Für die einzelnen Tracks werden die Korpora, die Themenstellungen und nach Durchführung der Evaluation die Relevanz-Bewertungen zur Verfügung gestellt. Im Jahr 2007 wurden insgesamt 8 unterschiedliche Tracks durchgeführt:

- Multilingual Document Retrieval on News Collections (Ad-Hoc)
- Scientific Data Retrieval (Domain-Specific)
- Interactive Cross-Language Information Retrieval (iCLEF)
- Multiple Language Question Answering (QA@CLEF)
- Cross-Language Image Retrieval (ImageCLEF)
- Cross-Language Speech Retrieval (CL-SR)
- CLEF Web Track (WebCLEF)
- Cross-Language Geographical Information Retrieval (GeoCLEF)

Jeder dieser Tracks ist in weitere Unterszenarien unterteilt, die so genannten Tasks. Für den ImageCLEF Track sind das:

- Allgemeine Fotografien
  - Ad-hoc photographic retrieval task (ImageCLEFphoto, 20.000 Fotografien mit semi-strukturierten Annotationen)
  - Object retrieval task[6]
- ImageCLEFmed
  - Medical image retrieval[7] (ca. 70.000 Aufnahmen mit mehrsprachigen Notizen zu den Krankheitsfällen)
  - Automatic annotation task for medical images[8] (hierarchische Annotationen von medizinischen Röntgenbildern in Englisch und Deutsch)

Der ImageCLEFphoto-Task wird im Kapitel 4.2 näher vorgestellt, da er als Grundlage zur Evaluation der Anwendungsfälle des Frameworks dient.

1.2.3 Ablauf einer Kampagne

Im folgenden Abschnitt wird der allgemeine Ablauf einer Evaluationskampagne beschrieben.

Zu Beginn einer Kampagne, in der Registrierungsphase, melden sich alle Gruppen, die an der Kampagne teilnehmen wollen, beim Veranstalter an. Dieser gewährt daraufhin den Teilnehmern den Zugriff auf die Ressourcen der Kampagne. Dazu gehören meist Dokumente und Datenbestände aus vorhergehenden Jahren und die aktuellen Korpora, die für die Durchführung der Experimente nötig sind.

Falls noch keine Topics aus vorhergehenden Kampagnen verfügbar sind oder neue Experimente neue Topics nötig machen, werden diese vom Veranstalter selbst unter zu Hilfenahme verschiedener Quellen erstellt. Zu den möglichen Quellen zählen Analysen von Protokollen

6 http://www-i6.informatik.rwth-aachen.de/~deselaers/imageclef07/objret.html - ImageCLEF 2007 – Object Retrieval Task
7 http://ir.ohsu.edu/image/ - ImageCLEFmed - Medical Image Retrieval Challenge Evaluation
8 http://www-i6.informatik.rwth-aachen.de/~deselaers/imageclef07/medaat.html - ImageCLEF 2007 - Medical Automatic Annotation Task

relevanter Anwendungen, das Wissen um den Inhalt des verwendeten Korpus oder die Ideen und Anmerkungen der Teilnehmer.

Nach der Veröffentlichung der Topics haben die Teilnehmer bis zu einem vorher festgelegten Termin Zeit, ihre Experimente durchzuführen und die Ergebnisse entsprechend den Anforderungen des Veranstalters einzureichen.

Nach dem Eingang der Ergebnisse wird die Relevanzbewertung durchgeführt. Dies stellt, angesichts der Menge der Ergebnisse, keine triviale Aufgabe dar. Hier schafft das „Pooling“ Abhilfe. Beim Pooling von [Gilbert 1979] erstellt ein spezieller Algorithmus aus den ursprünglichen Ergebnislisten für eine Topic eine einzige Ergebnisliste. Im Allgemeinen erfahren mehrfach auftretende Treffer oder besonders hoch eingestufte Treffer besondere Beachtung für die neue Ergebnisliste. Die Länge der neuen Ergebnisliste wird auf einen Bruchteil der ursprünglichen Listen reduziert, deshalb sind nicht alle Treffer der ursprünglichen Listen in der neuen Ergebnisliste vorhanden. Anhand dieser neuen, reduzierten Ergebnisliste führen nun Juroren die Relevanzbewertungen durch. Da verschiedene Juroren die Relevanz eines Ergebnisses unterschiedlich bewerten können, ist im Nachgang eine Verknüpfung alle Relevanzbewertungen zu einer Relevanzbewertungsliste nötig. Hierfür stehen wieder verschiedene Algorithmen zur Verfügung. Wenn zum Beispiel mindestens zwei Juroren ein Ergebnis als relevant bewerten, wird es in der endgültigen Liste auch als relevant bewertet.

Nach der Freigabe der Relevanzbewertungen wird jeder Gruppe die Möglichkeit gegeben, ihren Forschungsschwerpunkt bei der Durchführung der Experimente als Working Note oder als wissenschaftliche Veröffentlichung im Rahmen einer Buchreihe der Kampagne einzureichen.

Am Ende einer Kampagne steht meist ein Workshop, auf dem sich die Teilnehmer untereinander über ihre Fortschritte und auch über Probleme austauschen können.

2 Entwurf des Frameworks

In diesem Kapitel werden die Ausgangssituation für die Entwicklung des neuen Retrieval Frameworks sowie die Ziele und die Anwendungsszenarien, die sich auch den neuen Möglichkeiten ergeben, beschrieben.

2.1 Ausgangssituation

Es existiert bereits ein Retrieval System auf Basis von Apache Lucene (siehe [Lucene 2008]), das im Rahmen des Seminars „Cross-Language Retrieval“ an der Technischen Universität Chemnitz entwickelt und eingesetzt wurde. Es dient zum Ausführen und direkten Vergleichen von Suchläufen. Außerdem können eigene Relvanzbewertungen erfasst und gespeichert werden. Bereits implementiert sind Funktionen zur Erstellung eines Index, zur Suche in einem Index und zur Bewertung der Relevanz von Ergebnissen. Es wird bereits ein Diagramm für jeden Suchlauf generiert, das den entsprechenden Recall-Precision-Graphen darstellt.

Diese Lösung hat jedoch eine Reihe von Nachteilen, die durch eine Neuentwicklung behoben werden sollen:

(1) Ausschließlich Lucene-basiertes Retrieval

Die Lösung ist zu eng an Apache Lucene gekoppelt und erschwert damit die Verwendung anderer Such-Engines. Alle Daten müssen in einem Lucene-Index hinterlegt werden können. Die Klassen zum Einlesen der Korpora liefern ausschließlich Lucene-Dokumente zurück, die im Indizierungsprozess abgelegt werden.

(2) Fehlende Trennung zwischen Daten und Präsentation

Es gibt keine logische Trennung zwischen den Daten bzw. Datenstrukturen, der Logik und der Benutzeroberfläche. Funktionen zur Datenverarbeitung sind teilweise direkt in die Funktionen der Oberfläche eingebunden.

Sowohl die Klasse Indexer als auch die Klasse Searcher arbeiten direkt mit der Oberfläche zusammen, um ihren Status anzuzeigen.

(3) Begrenzte Erweiterungmöglichkeiten

Es gibt keine definierte API und keine Abstraktionsschicht. Alle Funktionen werden direkt verwendet. Das hat zum einen den Vorteil, das es auf viele Funktionen einen direkten Zugriff gibt, aber zum anderen den Nachteil, das alle Programme sehr speziell ausgeprägt sind und praktisch nicht wiederverwendet werden können.

Zum Beispiel besteht die Funktion der Klasse Indexer des Alt-Systems darin, Lucene-Dokumente, die durch einen XML-Parser aus XML-Dokumenten extrahiert werden, in einen neuen Lucene-Index einzufügen. Nicht-XML-Dokumente können nicht indiziert werden. Außerdem gibt es keine Möglichkeiten, Dokumente zu einem bestehenden Index hinzuzufügen oder Dokumente aus einem Index zu entfernen.

(4) Fehlende Flexibilität

Wie bereits unter Punkt (3) kurz beschrieben, können ausschließlich XML-basierte Korpora indiziert werden. Dies schließt zum Beispiel einfache Text-Dateien aus.

Außerdem sind die Datenstrukturen nicht flexibel genug, um den wechselnden Bedingungen unterschiedlicher Aufgabenstellungen zu genügen. Als Beispiel hierfür seien die Topics genannt, die im Alt-System aus den Feldern Topic-Nummer, Titel, Sprachecode, Beschreibung und einer detaillierten Beschreibung des Informationsbedürfnisses bestehen. Weitere Felder, die für Beispielbilder nötig wären, sind nicht vorhanden.

(5) Eingeschränkte Konfigurationsmöglichkeiten

Die Konfiguration der einzelnen Komponenten ist nur über speziell für die Komponenten angepasste Dialoge möglich.

2.2 Ziele

Das neue Retrieval System soll mindestens den Funktionsumfang des alten Systems besitzen, jedoch dessen Nachteile weitestgehend beseitigen. Daraus ergeben sich folgende Anforderungen an das neue System:

(1) Freie Wahl des Retrieval-Systems

Im Gegensatz zum Alt-System, das nur ein Lucene-basiertes Retrieval ermöglicht[9], soll das neue System die Wahl des Retrieval-Systems nicht einschränken. Daher soll es möglich sein, unterschiedliche Retrieval-Systeme zu verwenden und gegebenenfalls die Ergebnisse unterschiedlicher Retrieval-Systeme zu kombinieren.

Neben Lucene ist eine Unterstützung für Terrier[10] und Lemur[11] geplant.

(2) Unabhängigkeit zwischen Korpus und Indizierung

Die Indizierung soll vollständig unabhängig vom Korpus implementiert werden können, so dass mit dem gleichen Indizierungsverfahren unterschiedliche Korpora (GIRT4, IAPR, usw.) indizierbar sind. Dadurch soll die fehlende Flexibilität des Alt-Systems[12] bezüglich der verwendbaren Korpora aufgehoben werden.

(3) Verbesserte Unterstützung multilingualer Experimente

Das System muss Mehrsprachigkeit unterstützen, da es im besonderen für Cross-Language Retrieval Experimente vorgesehen ist.

(4) Evaluation

Wie auch beim Alt-System muss eine Evaluierung von Suchläufen möglich sein und gegebenenfalls sind die Möglichkeiten zur Evaluierung des Alt-Systems auszubauen. Dies Umfasst neben Funktionen zum Berechnen und Anzeigen von Recall-Precision-Graphen auch Funktionen zum Laden und Speichern von Relevanzbewertungen im QRELS-Format.

(5) Einfache Konfiguration

Es wird angestrebt eine einheitliche Konfigurationsschnittstelle zu schaffen, die den Nutzer mit einer grafischen Benutzeroberfläche bei der Konfiguration der Parameter der Klassen unterstützt. Dadurch steht die Möglichkeit zur Verfügung, die Parameter einer Klasse zur Laufzeit zu ändern, ohne den Quellcode der Klasse verändern zu müssen.

9 siehe Kapitel 2.1 Stichpunkt (1)
10 siehe [Ounis 2007]
11 siehe [Lemur 2007]
12 siehe Kapitel 2.1 Stichpunkte (3) und (4)

(6) Sicherung von Experimenten

Durchgeführte Experimente sollen gespeichert werden können, so dass die verwendeten Konfigurationen der einzelnen Komponenten nachgeschlagen werden können. Dies soll helfen einzelne Einstellungen und ihre Auswirkungen auf das Ergebnis besser analysieren zu können.

2.3 Anwendungsszenarien

Primär für die experimentelle Evaluation entworfen, haben sich durch die gewonnene Flexibilität des Retrieval Frameworks zusätzliche Anwendungsszenarien ergeben.

(1) Experimentelle Evaluation

Das zentrale Anwendungsszenario des Retrieval Framework ist die experimentelle Evaluation von Retrieval-Prozessen.

Der modulare Aufbau des Systems ermöglicht es, einzelne Komponenten auszutauschen und so neue Verfahren für die einzelnen Aufgaben zu evaluieren.

Zur Vereinfachung dieses Prozesses gibt es zusätzlich zum Framework zwei konkrete Implementierungen: einmal eine Implementierung für reines Text-Retrieval, die im Rahmen des Domain-Specific Tracks der CLEF-Kampagne entstand – und eine Implementierung basierend auf dem gleichen Text-Retrieval mit zusätzlichem Content-Based Image Retrieval für den ImageCLEFphoto-Track der gleichen Kampagne.

(2) Durchführung von Relevanzbewertungen

Für die Durchführung von Relevanzbewertungen können Ergebnislisten im TREC-Standard importiert und angezeigt werden. Dabei kann mit der richtigen DataCollection-Klasse sogar auf den Korpus zugegriffen werden und beispielsweise können für den IAPR-TC12 Korpus die Bilder und Annotationen angezeigt werden.

(3) Klassisches Suchmaschinenszenario

Im Rahmen dieser Diplomarbeit nicht geplant, aber durchaus denkbar ist ein Einsatz als klassische Suchmaschine. Das heißt, ein Nutzer formuliert eine Informationsbedürfnis, das durch das System entweder direkt in Form einer Query oder über den Um-

weg einer temporären Topic für den Retrieval-Prozess genutzt wird. Als Ergebnis wird dem Nutzer die Ergebnisliste präsentiert. Anhand der Ergebnisliste kann er entweder weitere Verfeinerungen seiner Query vornehmen oder ein Feedback für einen erneuten Suchlauf eingeben.

3 Aufbau des Frameworks

Das Framework besteht aus zwei Hauptkomponenten: der Kernkomponente und der grafischen Benutzeroberfläche (GUI).

Die Kernkomponente stellt grundlegende Funktionen für das Information Retrieval zur Verfügung und definiert die Schnittstellen (Interfaces) für alle zusätzlichen Funktionen. Es enthält keine spezielle Implementierung einer Indizierung oder einer Retrieval Methode.

Die grafische Benutzeroberfläche stellt dem Benutzer die Funktionen des Frameworks zur Verfügung und erzeugt aus den von der Kernkomponente gelieferten Kennzahlen ansprechende und übersichtliche Grafiken und Diagramme.

Die soeben genannten Hauptkomponenten haben keine weiteren externen Abhängigkeiten von Apache Lucene oder irgendeiner anderen Retrieval-Engine, um Punkt (1) aus Kapitel 2.2 zu entsprechen.

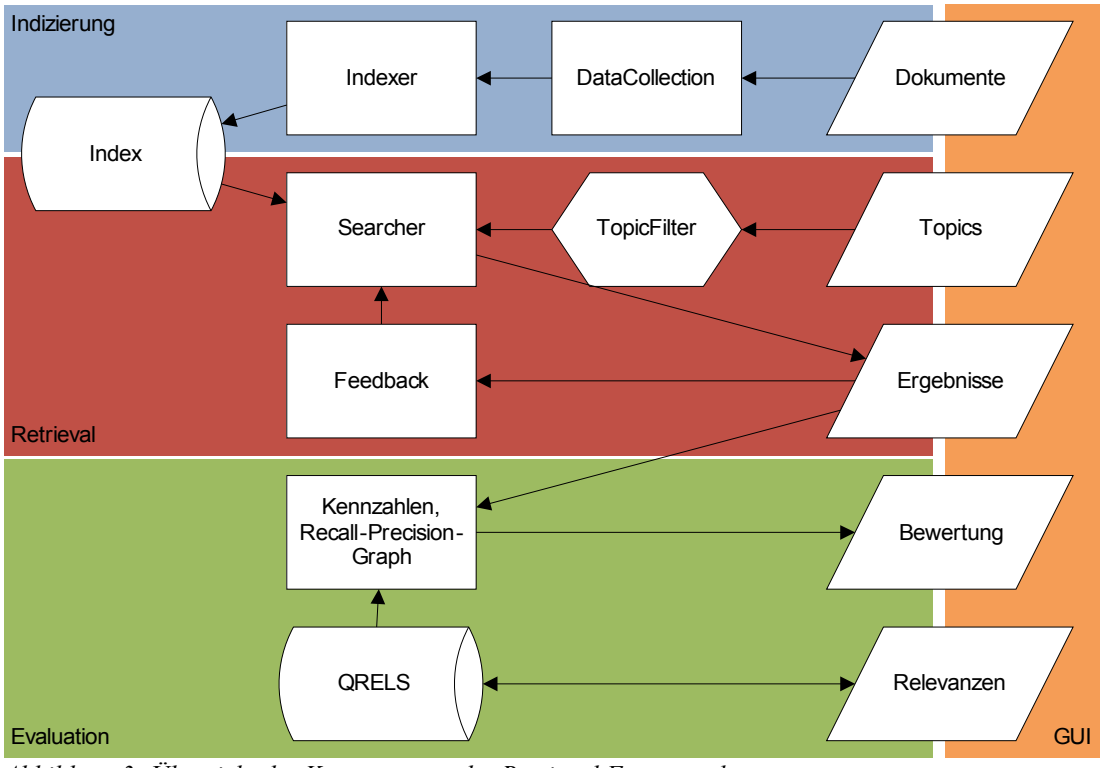


Abbildung 3: Übersicht der Komponenten des Retrieval Frameworks

Zusätzlich zu den zwei Hauptkomponenten existieren zwei anwendungsfallbasierte Komponenten. Erst bei diesen Komponenten tritt die Verbindung zu Lucene oder einem anderen Retrieval System auf. Das beiden anwendungsfallbasierten Komponenten sind:

- Eine Implementierung von Indizierung und Suche des GIRT4-Korpus durch Verwendung von Lucene. In diesem Paket enthalten ist eine speziell auf Lucene zugeschnittene Feedback-Implementierung. Außerdem sind einige spezielle TopicFilter zur automatischen Übersetzung durch einen Online-Übersetzungsdienst und zur Query Expansion auf Basis eines Thesaurus vorhanden.
- Eine Implementierung von Indizierung und Suche des IAPR TC-12 Korpus unter Verwendung von Lucene (unter Verwendung der Klasse aus der vorherigen Komponente) und Caliph&Emir für das Content-Based Image Retrieval. Zur Unterstützung der erweiterten Topics mit Beispielbildern entstand ein darauf zugeschnittener TopicLoader.

3.1 Kernkomponente

Die Kernkomponente des Frameworks ist in drei funktionale Bereiche[13] unterteilt:

(1) Zur **Indizierung** gehören die Klassen: DataCollection, DataDocument, DataField, DataFieldMapping, DataFieldType, Index und Indexer.

(2) Das **Retrieval** umfasst: Hit, HitSet, Merger, MergeSearcher, QueryExpansion, Searcher, Topic, TopicField, TopicFilter und TopicLoader

(3) Zum Bereich **Evaluation** gehören: Relevance, Run, BoxPlot und RecallPrecisionGraph.

13 siehe auch Abbildung 3

3.1.1 Indizierung

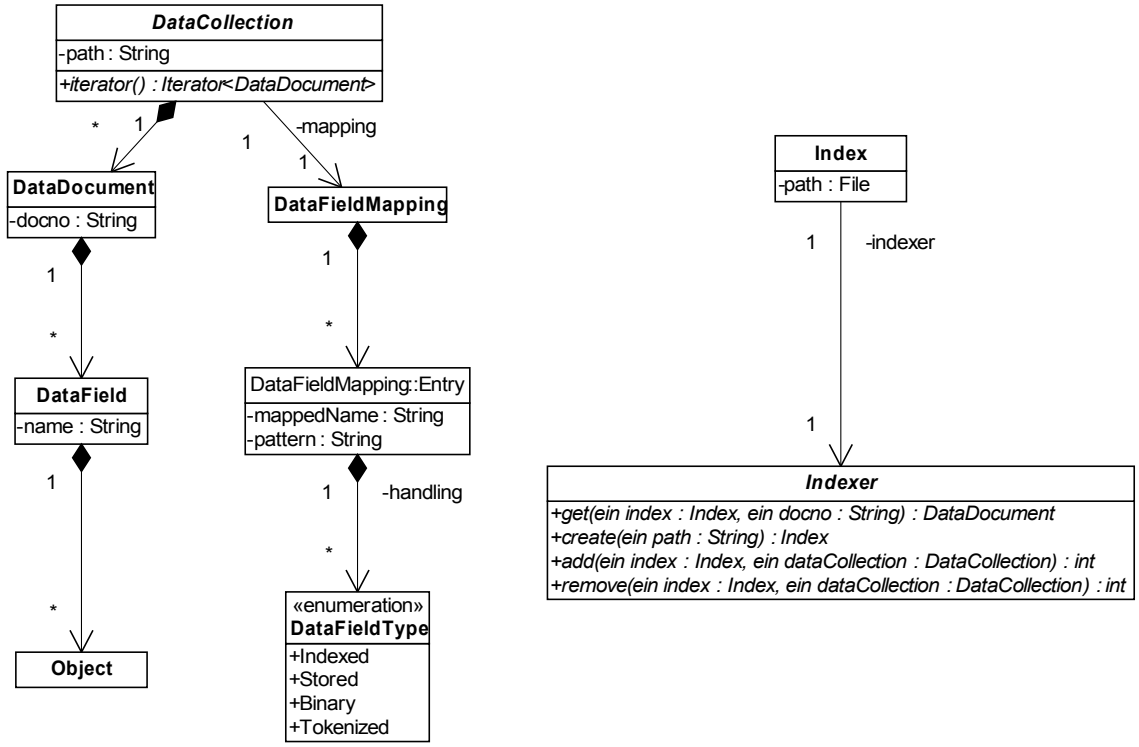


Abbildung 4: UML-Diagramm der Komponenten für die Indizierung

3.1.1.1 Abstrakte Klasse: DataCollection

Die abstrakte Klasse DataCollection ist eine Repräsentation eines Korpus. Sie besteht aus mehreren Elementen der Klasse DataDocument und einem DataFieldMapping-Objekt. In der Kernkomponente werden nur Methoden zum Auslesen und Setzen des Korpus und zum Auslesen des DataFieldMappings implementiert. Weitere Methoden sind entweder nur Funktionsrümpfe, das heißt sie führen keine Operationen aus und liefern keine Ergebnisse oder sie liefern abstrakte Funktionen, die von der erbenden Klasse implementiert werden müssen. Zu den zu implementierenden Methoden gehört:

- `Iterator<DataDocument> iterator()`

  Mit Hilfe des Iterators können alle Dokumente des Korpus als DataDocument ausgelesen werden.

3.1.1.2 Klasse: DataDocument

Die Klasse DataDocument repräsentiert die einzelnen Dokumente eines Korpus. Jedes Dokument besitzt eine eindeutige Identifizierung in Form einer Zeichenkette, der Dokumenten-Nummer, und besteht aus mindestens einem DataField.

3.1.1.3 Klasse: DataField

Jedes Objekt der Klasse DataField besitzt einen eindeutigen Namen, über den es in der DataDocument-Klasse angesprochen werden kann, und mehrere Objekte eines nicht weiter spezifizierten Typs. Dass heißt ein Feld kann sowohl eine einfache Zeichenkette repräsentieren als auch ein Bild oder eine Tonspur. Die Art und Weise der Behandlung der einzelnen Typen wird dabei vom verwendeten Indexer bestimmt.

3.1.1.4 Klasse: DataFieldMapping

Mit Hilfe der Klasse DataFieldMapping werden bestimmte Parameter für die Indizierung der einzelnen Felder definiert. Dies geschieht auf DataCollection-Ebene, so dass der Indexer keine direkte Kenntnis über den Aufbau des Korpus besitzen muss.

3.1.1.5 Klasse: DataFieldType

Im Aufzählungstyp DataFieldType sind folgende Datenfeldtypen definiert:

- `Indexed` – Das Datenfeld wird durch den Indexer indiziert.
- `Stored` – Der Inhalt des Datenfeldes wird im Index gespeichert.
- `Binary` – Der Inhalt des Datenfeldes ist binärer Natur.
- `Tokenized` – Der Inhalt des Datenfeldes wird vor der Indizierung normalisiert.

Alle vier Datenfeldtypen können einzeln, aber auch in Kombination in einem EnumSet<DataFieldType> auftreten.

3.1.1.6 Klasse: Index

Der Index repräsentiert einen erstellten Index. Er speichert Informationen zum verwendeten Indexer und dessen Konfiguration, zur verwendeten DataCollection und deren Konfiguration und zum Speicherort des erstellten Indexes.

3.1.1.7 Abstrakte Klasse: Indexer

Der Indexer ist wieder eine abstrakte Klasse. Die Aufgabe eines Indexers besteht darin, die Dokumente des Korpus zu indizieren und den erzeugten Index zurückzugeben. Folgende Methoden müssen von einem Indexer implementiert werden:

- `Index create( String path )`

  legt im angegebenen Pfad einen neuen, leeren Index an und gibt diesen zurück.

- `int add( Index index, DataCollection dataCollection )`

  fügt den Korpus zum übergebenen Index hinzu und gibt die Anzahl der hinzugefügten Dokumente zurück

- `int remove( Index index, DataCollection dataCollection )`

  entfernt alle Dokumente des Korpus aus dem Index und gibt die Anzahl der entfernten Dokumente zurück

- `DataDocument get( Index index, String docno )`

  rekonstruiert, wenn möglich, das ursprüngliche DataDocument.

3.1.2 Retrieval

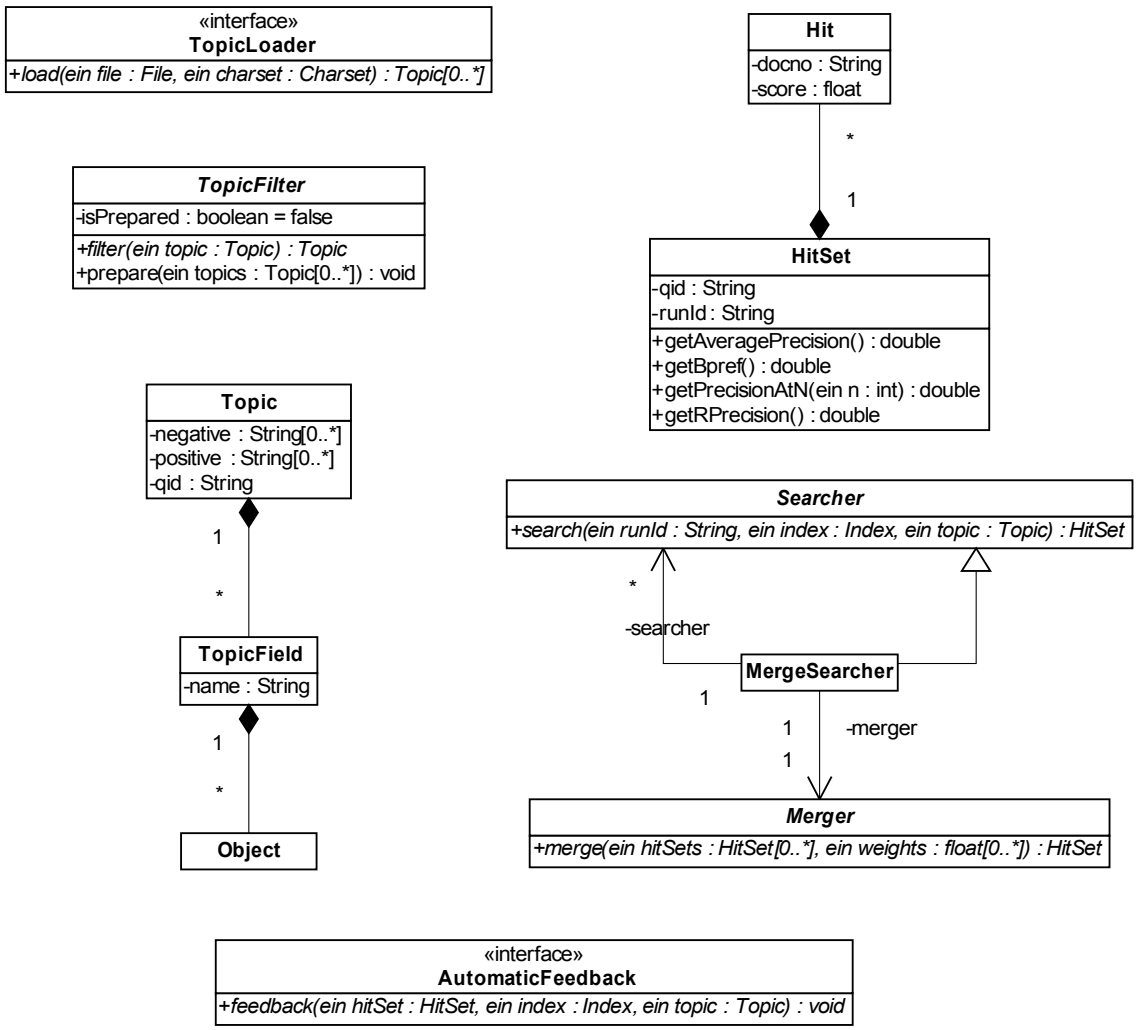


Abbildung 5: UML-Diagramm der Komponenten für das Retrieval

3.1.2.1 Klasse: Topic

Die Klasse Topic ist eine Repräsentation einer Suchanfrage bzw. eines Informationsbedürfnisses. Eine Topic besteht aus einer eindeutigen Identifikation („qid“) in Form einer Zeichenkette und aus einer Menge von Feldern, die Informationen zur Suchanfrage enthalten.

Jede Topic kann außerdem je eine Listen mit relevanten und irrelevanten Dokumenten enthalten, die vom Searcher ausgewertet werden können und somit für ein Feedback nutzbar sind.

3.1.2.2 Klasse: TopicField

Ein TopicField hat eine eindeutige Bezeichnung, über die es in der Topic identifiziert werden kann. Es besteht aus einer Menge von Objekten, deren Typ bzw. Klasse nicht festgelegt ist. Es können also Objekte beliebigen Typs in einem TopicField abgelegt werden.

3.1.2.3 Interface: TopicLoader

Der TopicLoader ist ein Interface (Schnittstelle), um Topics aus einer Datei in einem bestimmten Format auszulesen. Das Interface definiert folgende Methode:

- `Topic[] load( File file, Charset charset )`

  Diese Funktion erzeugt aus einer Datei einen Array von Topics unter Berücksichtigung des Zeichensatzes, wenn dieser angegeben ist.

3.1.2.4 Abstrakte Klasse: TopicFilter

TopicFilter sind Klassen zur Vorverarbeitung von Topics vor der tatsächlichen Suche. Dieser Schritt der Vorverarbeitung kann zum Beispiel zur Query Expansion genutzt werden. Die abstrakte Klasse TopicFilter der Kernkomponente liefert dafür den Rahmen und definiert die zu implementierende Methode:

- `Topic filter( Topic topic )`

  Es wird eine neue Topic mit den verarbeiteten Daten der übergebenen Topic zurückgegeben.

3.1.2.5 Klasse: Hit

Ein Hit ist ein einzelnes Ergebnis (auch Treffer). Es setzt sich zusammen aus der Dokumenten-Nummer „docno“ und einer Relevanzbewertung, die vom Searcher vergeben wird, dem sogenannten Score.

3.1.2.6 Klasse: HitSet

Objekte vom Typ HitSet stellen eine Ergebnisliste dar. Ein HitSet wird von einem Searcher angelegt und enthält neben der Ergebnisliste den Identifikator der verwendeten Topic und eine Suchlauf-Identifikation.

3.1.2.7 Abstrakte Klasse: Searcher

Die Klasse Searcher der Kernkomponente ist eine abstrakte Klasse, die folgende Methode zur Implementierung in erbenden Klassen vorgibt:

- `HitSet search( String runId, Index index, Topic topic )`

  Diese Methode führt die Suche auf dem angebenen Index mit der übergebenen Topic aus und liefert eine Ergebnisliste mit der Suchlauf-Identifikation „runId“ zurück.

3.1.2.8 Abstrakte Klasse: Merger

Merger sollen zum Zusammenführen von zwei oder mehreren Ergebnislisten dienen. Da es für diesen Prozess viele verschiedene Methoden und Ansätze gibt, ist die Merger-Klasse der Kernkomponente wieder eine abstrakte Klasse, die eine Implementierung folgender Methode erfordert:

- `HitSet merge( HitSet[] hitSets, float[] weights )`

  Es wird eine Ergebnisliste zurückgegeben, die aus den übergebenen Ergebnislisten mit den angegebenen Gewichtungen generiert wird.

3.1.2.9 Klasse: MergeSearcher

Der MergeSearcher verknüpft die Ergebnislisten mehrerer Searcher mit Hilfe eines Mergers zu einer Ergebnisliste. Er implementiert alle durch die Searcher-Klasse geforderten Methoden und stellt sich damit im System selbst als Searcher dar.

3.1.2.10 Interface: AutomaticFeedback

Das Interface AutomaticFeedback erfordert die Implementierung einer Funktion, die automatisch das Feedback für die Ergebnisse der Suche einer Topic ermittelt.

- `void feedback( HitSet hitSet, Index index, Topic topic )`

  Bestimmt anhand der Ergebnisliste „hitSet“ das Feedback, das in die Topic eingetragen wird. Das heißt, Dokumente werden entsprechen ihrer Relevanz in die Listen *positive* und *negative* der Topic eingetragen und können später von einem Searcher ausgewertet werden.

3.1.3 Evaluation

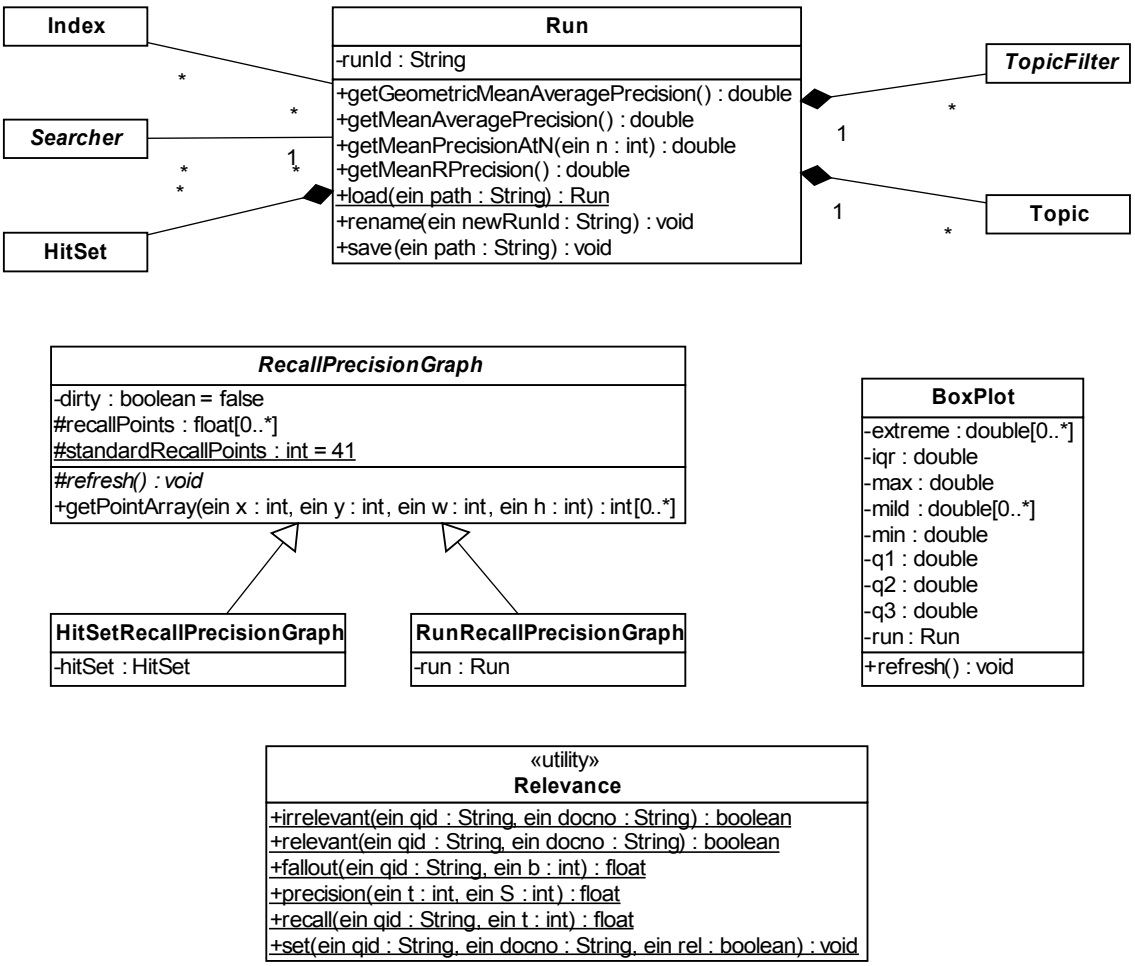


Abbildung 6: UML-Diagramm der Komponenten für die Evaluation

3.1.3.1 Klasse: Run

Die zentrale Klasse des Frameworks und die Grundlage der Evaluation ist die Klasse Run. Sie speichert sämtliche Informationen, die zum Ausführen und Wiederholen einer Suche benötigt werden. Dies umfasst:

- die Topics, die mit Hilfe von TopicFiltern vorverarbeitet werden können (z.B. durch eine Übersetzung),
- den Index, der durch einen Indexer aus einer DataCollection erstellt wurde,
- die Such-Methode, die im Searcher implementiert ist und
- die Ergebnislisten, die durch HitSets repräsentiert werden.

Die Kennzahlen stehen durch die Funktionen *getGeometricMeanAveragePrecision*, *getMeanAveragePrecision*, *getMeanPrecisionAtN* und *getMeanRPrecision* zur Verfügung.

Die Funktionen *save* und *load* speichern oder laden einen Run in Form einer XML-Datei mit allen vorhandenen Informationen, inklusive der Konfiguration der DataCollection, des Indexers und des Searchers.

3.1.3.2 Statische Klasse: Relevance

Die Relevanzbewertungen verwaltet das System mit Hilfe der statischen Klasse Relevance. Die Klasse enthält Funktionen zum Laden und Speichern von Relevanzbewertungen im QRELS-Format (siehe [TREC 2005]), zur Bestimmung der Relevanz einzelner Dokumente und zur Berechnung der Kennzahlen Precision und Recall.

3.1.3.3 Klasse: BoxPlot

Diese Klasse dient der Berechnung der Kennzahlen, die zur Darstellung eines Box-Whisker-Plots, wie von [Tukey 1977] beschrieben, nötig sind. Über die Funktionen *getQ1*, *getQ2* und *getQ3* sind das erste Quartile, der Median und das dritte Quartile abrufbar. Die Länge der Whiskers wird mittels *getMin* und *getMax* bestimmt. Die Funktionen *getMild* und *getExtreme* liefern Listen mit den „milden“ und den „extremen“ Ausreißern, die außerhalb der Box und der Whisker liegen. Der Interquartilsabstand (IQR) wird von der Funktion *getIQR* zurückgegeben. Durch den Aufruf der Funktion *refresh* werden alle Werte auf Grundlage des zugewiesenen Runs neu berechnet. Die Funktion *refreshAll* berechnet die Werte aller Box-Whisker-Plots neu.

3.1.3.4 Abstrakte Klasse: RecallPrecisionGraph

Alle Recall-Precision-Graphen basieren auf dieser abstrakten Klasse. Sie stellt die Datenstrukturen und Berechnungsgrundlagen für die abgeleiteten Klassen zur Verfügung. Dazu gehört die Funktion *getPointArray*, die die vorhandenen Daten für die Darstellung in einem ganzzahligen Koordinatensystem umrechnet.

- `void refresh()`

  Ein Aufruf dieser Funktion führt zu einer Neuberechnung der internen Daten.

3.1.3.5 Klassen: HitSetRecallPrecisionGraph und RunRecallPrecisionGraph

Beide Klassen erben von der abstrakten Klasse RecallPrecisionGraph und setzen die Funktion *refresh* entsprechend der zugewiesenen Objekte um. Die Klasse HitSetRecallPrecisionGraph arbeitet mit Objekten der Klasse HitSet, stellt also die Daten für den Recall-Precision-Graph eine Ergebnisliste zur Verfügung. Die Klasse RunRecallPrecisionGraph arbeitet mit Objekten der Klasse Run und stellt die Daten des Durchschnitts-Recall-Precision-Graph aller im Run enthaltenen Ergebnislisten zur Verfügung.

3.2 Grafische Benutzeroberfläche

Für die Evaluation stehen im Retrieval Framework folgende Funktionen zur Verfügung:

- **Erfassen von Relevanz-Bewertungen**

  Im Dialogfenster zur Anzeige des Inhaltes eine Dokumentes kann die Relevanz des Dokumentes im Bezug auf die ausgewählte Topic erfasst werden.

  Relevante Dokumente werden in der Ergebnistabelle durch eine grüne Schriftfarbe hervorgehoben, nicht-relevante Dokumente haben eine rote Schriftfarbe und nicht bewertete Dokumente besitzen eine schwarze Schriftfarbe.

  Im Hauptmenü sind die Funktionen zum Laden und Speichern der Relevanz-Bewertungen im QREL-Format (siehe [TREC 2005]) zu finden. Außerdem können alle geladenen oder erfassten Bewertungen aus dem Speicher gelöscht werden.

- **Anzeige des Recall-Precision-Diagramms**

  Es werden sowohl das durchschnittliche Recall-Precision-Diagramm aller Suchanfragen eines Suchlaufs (blau) als auch das Diagramm der aktuell ausgewählten Topic/Suchanfrage (rot) angezeigt. Dies erlaubt einen Vergleiche zwischen den Ergebnis jeder einzelnen Suchanfrage mit dem Gesamtergebnis des Experiments, um gute und auch schlechte Ergebnisse einzelner Suchanfragen identifiziert zu können.

- **Anzeige der Kennzahlen**

  Für jeden Suchlauf sind die Mean Average Precision (MAP) und die Geometric Mean Average Precision (GMAP), und für jede Suchanfrage sind die Average Precision

(AP), die R-Precision (RP) und die Precision at 20 (P20) in die Anzeige des Recall-Precision-Diagramms integriert.

- **Direkter Vergleich unterschiedlicher Suchläufe**

  Im so genannten „Vergleichsgraph“ (siehe Kapitel 3.2.1.2) können alle geöffneten Suchläufe zusammen in einem Diagramm angezeigt werden. Es besteht auch die Möglichkeit, die Ergebnisse einzelner Suchanfragen direkt miteinander zu vergleichen. Die Graphen zusammengehörender Suchläufe und Suchanfragen besitzen die gleiche Farbe. Die Linien der Suchläufe sind durchgehend, die der Suchanfragen unterbrochen.

- **Box-Whisker-Plot unterschiedlicher Suchläufe**

  Für einen Überblick über die Verteilung der Topic-Ergebnisse sorgt die Box-Plot-Ansicht (siehe Kapitel 3.2.1.3). Es wird die Verteilung der Average Precision der einzelnen Suchanfragen für alle geöffneten Experimente untereinander angezeigt. Dies soll das Ermitteln von Ausreißern bei den Ergebnissen der Suchanfragen für unterschiedlichen Experimentanordnungen vereinfachen.

3.2.1 Hauptfenster

Das Hauptfenster besteht aus einer Reiter-Leiste, in der alle geöffneten Reiter zu sehen sind und ausgewählt oder geschlossen werden können. Auf der rechten Seite dieser Reiter-Leiste befinden sich die Schalter zum Öffnen des Vergleichsgraphen-Reiters (3.2.1.2) und des Boxplot-Reiters (3.2.1.3) und das Hauptmenü.

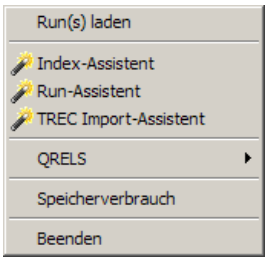


Abbildung 7: Hauptmenü der grafischen Benutzeroberfläche

Über das Hauptmenü erhält man Zugriff auf die Assistenten zur Erzeugung eines neuen Index, zur Durchführung eines Experiments (Run) und zum Import von TREC-Ergebnislisten.

Es besteht die Möglichkeit zuvor gespeicherte Experimente zu laden und QRELS, die im Rahmen der TREC-Kampagnen standardisierten Relevanzbewertungen, sowohl zu laden als auch zu speichern.

Darüber hinaus kann der aktuelle Speicherverbrauch angezeigt werden. Diese Funktion dient der Detektion von Speicherlecks, die zwar in Java unüblich sind, aber gegebenenfalls durch schlechtes Objektverweis-Management auftreten können.

3.2.1.1 Experiment-Reiter

Hinter dem Experiment-Reiter verbirgt sich die Hauptansicht des Frameworks. Alle Informationen zu einem Experiment sind hier entweder direkt oder über das Menü abrufbar.

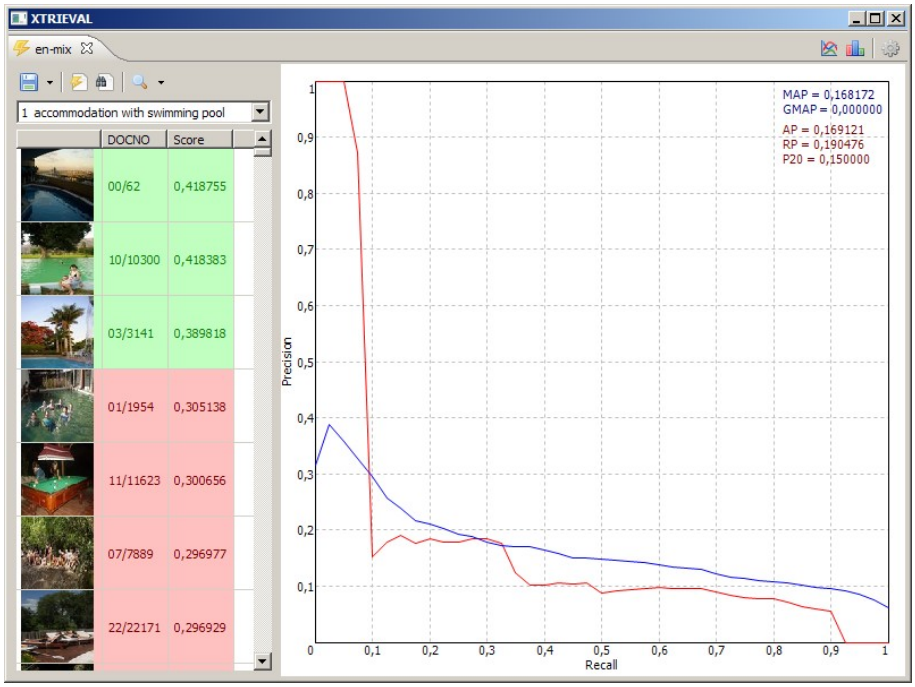


Abbildung 8: Experiment-Reiter mit ausgewählter Topic, Relevanzbewertungen und Feedback

Die Ansicht des Experiment-Reiters unterteilt sich in zwei Hauptbereiche:

- Auf der *linken Seite* befinden sich eine Werkzeugleiste, eine Dropdown-Liste mit allen geladenen Topics und die Ergebnisliste der aktuell ausgewählten Topic.

  In der Werkzeugleiste stehen folgende Befehle zur Verfügung (von links nach rechts): Run speichern/exportieren, Informationen zum Run anzeigen, Informationen zur Topic anzeigen, Suchoptionen für den Run.

Wenn Relevanzbewertungen zur Verfügung stehen, werden die betroffenen Ergebnisse im Falle von relevanten Treffern mit grüner Schriftfarbe und im Falle von nicht-relevanten Treffern mit roter Schriftfarbe in der Ergebnisliste dargestellt. Durch einen Rechtsklick auf einen Tabelleneintrag erscheint ein Kontext-Menü, das es dem Nutzer ermöglicht für den ausgewählten Eintrag eine Relevanzbewertung vorzunehmen oder ihn für das Feedback entsprechend zu markieren. Einträge, die mit einem positiven Feedback versehen sind, haben einen hellgrünen Hintergrund und Einträge, die mit einem negativen Nutzer-Feedback versehen sind, haben einen hellroten Hintergrund.

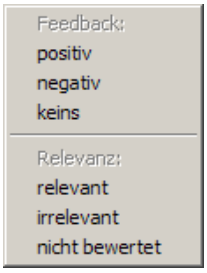


Abbildung 9: Kontext-Menü für einen Eintrag in der Ergebnisliste

Bei einem Doppelklick auf einen Eintrag in der Ergebnisliste werden weitere Details zu dem entsprechenden Dokument angezeigt.

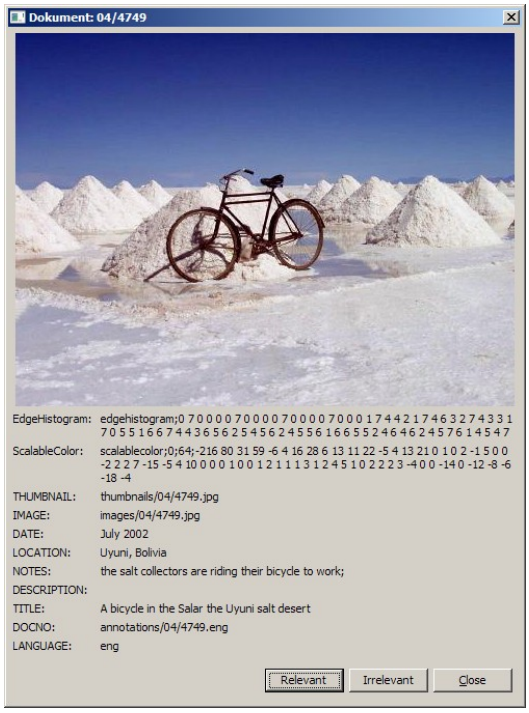


Abbildung 10: Detailansicht für einen Eintrag aus der Ergebnisliste

- Auf der rechten Seite befindet sich der Recall-Precision-Graph. Der blaue Graph und die blauen Kennzahlen repräsentieren das gesamte Experiment. Wenn auf der linken Seite eine Topic ausgewählt ist, dann erscheinen der Graph und die Kennzahlen für die gewählte Topic in roter Farbe im Diagramm.

3.2.1.2 Vergleichsgraph-Reiter

Der Vergleichsgraph-Reiter zeigt die Graphen aller geöffneten Experiment-Reiter als durchgehende Linie an. Wenn in der oberen rechten Ecke zusätzlich eine Topic ausgewählt ist, wird zusätzlich der Recall-Precision-Graph für die Ergebnisse zur gewählten Topic als unterbrochene Linie angezeigt. Außerdem befinden sich unter der Dropdown-Liste zur Auswahl der Topic die wichtigsten Kennzahlen aller geöffneten Experimente. So ist ein direkter Vergleich verschiedener Experimente möglich.

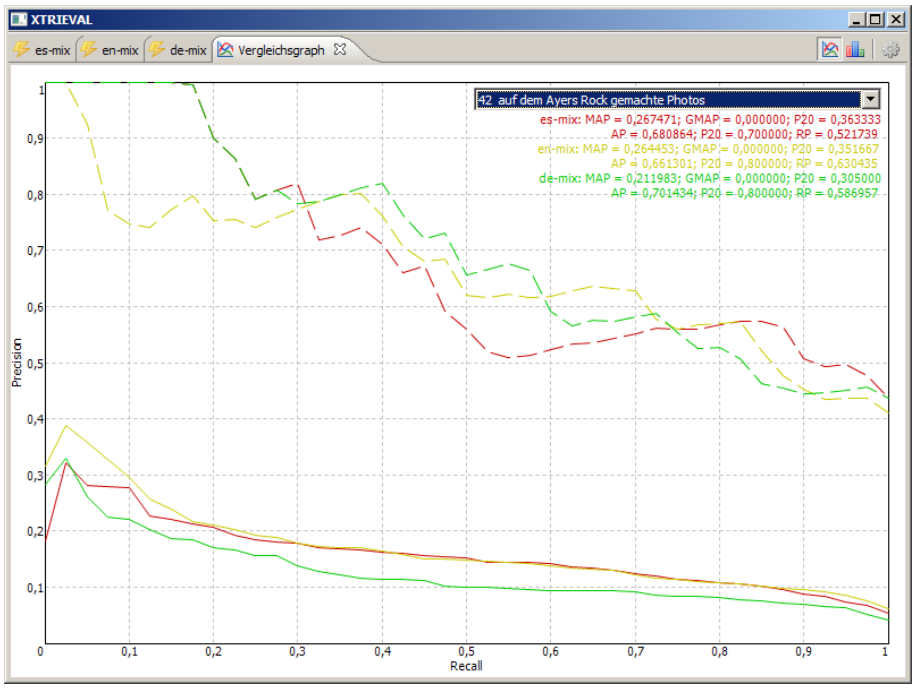


Abbildung 11: Vergleichsgraph-Reiter mit 3 Experimenten

3.2.1.3 Box-Plot-Reiter

Im Box-Plot-Reiter ist pro Experiment ein Box-Plot über der Verteilung der Average Precison (AP) pro Topic sichtbar. Alle Box-Plots werden untereinander angeordnet, so dass die Ergebnisse miteinander verglichen werden können.

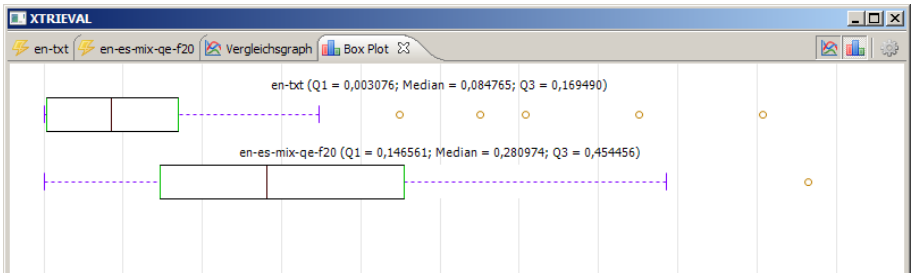


Abbildung 12: Box-Plot-Reiter mit 2 Experimenten

3.2.2 Assistenten

Verschiedene Assistenten unterstützen den Nutzer bei der Durchführung von Aufgaben. Alle Assistenten haben den gleichen Grundaufbau:

- Im *oberen Teil* steht der Name der aktuellen Seite. Direkt darunter kann der Assistent zusätzliche Meldungen für den Benutzer anzeigen. Dazu zählen Informationen, Warnungen und Fehlermeldungen bezüglich der aktuellen Eingabe.
- Im *mittleren Teil* kann der Nutzer die erforderlichen Daten eingeben oder bearbeiten.
- Im *unteren Teil* befinden sich immer mindestens vier Schalter, zwei davon für die Navigation im Assistenten. Sie ermöglichen es dem Nutzer zur vorangegangenen Seite zu wechseln und zuvor eingegebene Daten zu ändern oder zur nächsten Seite zu wechseln, um weitere benötigte Daten einzugeben.

  Nach Betätigen des Schalters „Finish“ bzw. „Fertigstellen“ führt der Assistent die konfigurierten Operationen aus und zeigt gegebenenfalls das Ergebnis an. „Cancel“ bzw. „Abbrechen“ beendet den Assistenten, ohne eine Operation auszuführen.

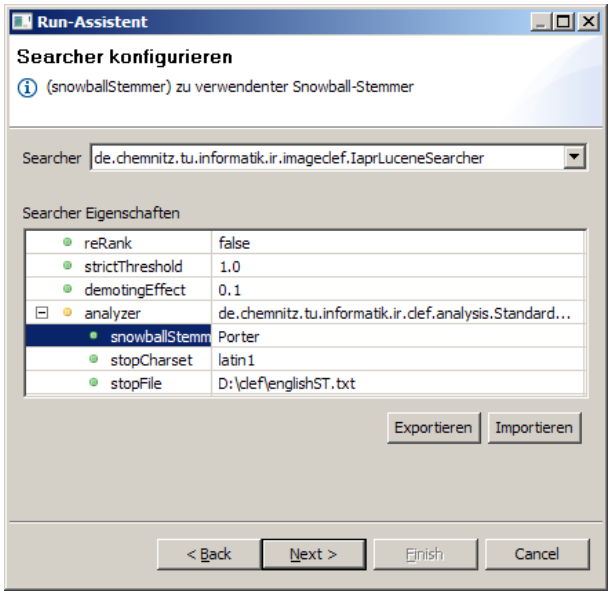


Abbildung 13: Run-Assistent mit aktiver Seite zur Konfiguration der Searcher-Klasse

3.3 Anwendungsfälle

Um die Leistungsfähigkeit des Retrieval Frameworks bewerten zu können, sind zwei Anwendungsfälle ausgewählt und implementiert worden.

Der erste Anwendungsfall basiert auf dem Domain-Specific-Track der CLEF-Kampagne (siehe Kapitel 1.2.2.2) und umfasst die Implementierung eines Text Retrieval Verfahrens auf der Basis von Apache Lucene (siehe [Lucene 2008]).

Der zweite Anwendungsfall basiert auf dem ImageCLEFphoto-Task als Teil des ImageCLEF-Tracks der CLEF-Kampagne. Hier wird auf das Text Retrieval Verfahren aus dem ersten Anwendungsfall zurückgegriffen und dieser um zusätzliche Funktionen zur Bilddatenverarbeitung mit MPEG-7 Deskriptoren erweitert. Weiterführende Informationen zum ImageCLEF-photo-Task sind Kapitel 4.1 zu entnehmen.

3.3.1 CLEF Domain-Specific (GIRT4)

Der CLEF Domain-Specific-Track untersucht mono-, bi- und multilinguale Retrieval-Aufgaben in strukturierten, wissenschaftlichen Daten, dem GIRT4-Korpus. Der GIRT4-Korpus setzt sich aus Teilen der SOLIS und SOFIS Datenbank der Gesellschaft Sozialwissenschaftli-

cher Infrastruktureinrichtungen[14] (GESIS) zusammen. Insgesamt enthält der GIRT4-Korpus 151.319 Dokumente. Neben dem GIRT4-Korpus steht ein intellektuell gepflegter Thesaurus zur Verfügung.

3.3.1.1 Klasse: Girt4DataCollection

Diese Klasse implementiert die Klasse DataCollection für den Zugriff auf den GIRT4-Korpus. Der Zugriff auf die Daten erfolgt mittels DOM-Parser. Da im GIRT4-Korpus vereinzelt fehlerhafte XML-Notationen vorliegen ist eine Vorverarbeitung notwendig, die von dieser Klasse automatisch durchgeführt wird.

3.3.1.2 Klasse: LuceneIndexer

Der LuceneIndexer stellt die Schnittstelle zu Apache Lucene dar. Der volle Umfang des Indexer-Interfaces ist implementiert. Das heißt es können ein neuer Index angelegt, Dokumente zu einem Index hinzugefügt und Dokumente aus einem Index entfernt werden. Alle vier in der Kernkomponente definierten DataFieldTypes werden unterstützt: Indexed, Stored, Binary, Tokenized.

Apache Lucene verwendet zur Normalisierung so genannte Analyzer. Der LuceneIndexer unterstützt dieses Verfahren, indem der zu verwendende Analyzer übergeben werden kann. Als Standard-Analyzer wird der von Lucene mitgelieferte StandardAnalyzer aus dem Package org.apache.lucene.analysis.standard eingesetzt.

3.3.1.3 Abstrakte Klasse: LuceneSearcher

In der abstrakten Klasse LuceneSearcher sind grundlegende Funktionen zur Suche mit Lucene definiert. Sie enthält die Definition eines Lucene-Analyzers, der von den erbenden Klassen genutzt werden kann. Außerdem stellt sie Funktionen für den Zugriff auf einen Index zur Verfügung. Erbende Klassen müssen folgende Funktion implementieren:

- `Query buildQuery( Index index, Topic topic )`

14 http://www.gesis.org/ - GESIS Homepage

Diese Funktion gibt eine Lucene-Query für die übergebene Topic und den übergebenen Index zurück.

3.3.1.4 Abstrakte Klasse: LuceneFeedbackSearcher

Der LuceneFeedbackSearcher erweitert die Klasse LuceneSearcher um die Funktion, das in einer Topic hinterlegte Feedback auszuwerten. Die Original-Suchanfrage (Query) wird wie beim LuceneSearcher durch die Funktion buildQuery erzeugt, die von der erbenden Klasse zu implementieren ist. Die Feedback-Funktion arbeitet sehr eng mit dem Lucene-Index zusammen und läuft in folgenden Schritten ab:

(1) Alle indizierten Terme werden hinsichtlich ihre Häufigkeit in den als relevant eingestuften Dokumenten untersucht und in einer Datenstruktur mit zusätzlichen Daten der Häufigkeit betreffend abgelegt.

(2) Die Datenstrukturen werden sortiert abgearbeitet. Besonders häufig vorkommende Terme werden als relevant eingestuft und zur Suchanfrage (Query) hinzugefügt.

(3) Die neue Suchanfrage wird an den LuceneSearcher übergeben und verarbeitet.

Als Besonderheit dieses Ansatzes ist anzumerken, das die Query Expansion auf Feedbackbasis nicht an spezielle Feldnamen gebunden ist. Das heißt, sie funktioniert ohne Anpassung mit jedem beliebigen Lucene-Index, unabhängig von dessen Struktur oder Inhalt. Dadurch können auch weniger offensichtliche Verbindungen zwischen bekannten, relevanten Dokumenten entdeckt und zum Auffinden neuer Dokumente genutzt werden.

3.3.1.5 Klasse: TranslationTopicFilter

Nicht im Rahmen dieser Diplomarbeit entwickelt, jedoch von Bedeutung ist der TranslationTopicFilter. Er ermöglicht es, den textuellen Inhalt einer Topic mit Hilfe verschiedener Online-Übersetzungsdienste in eine andere Sprache zu übersetzen. Erste Experimente mit dem TranslationTopicFilter sind in [Kürsten 2007] beschrieben.

3.3.2 ImageCLEFphoto (IAPR TC-12)

Für die Teilnahme am ImageCLEFphoto-Task (siehe Kapitel 4.1) wurden die folgende Klassen entworfen und implementiert.

3.3.2.1 Klasse: IaprDataCollection

Die IaprDataCollection stellt den Zugang zum IAPR TC-12 Benchmark Korpus bereit. Es besteht die Möglichkeit, den Korpus sowohl in komprimierter Form (als ZIP-Archiv) als auch vollständig entpackt zu nutzen. Neben der textuellen Beschreibung der Fotografien stehen auch die Bilder selbst in Originalgröße und als Vorschaubild (Thumbnail) zur Verfügung.

3.3.2.2 Klasse: IaprTopicLoader

Die Topics im ImageCLEFphoto-Task beinhalten außer einer textuellen Beschreibung der Suchanfrage auch Beispielbilder, die den gesuchten Bildern entsprechen oder ähnliche Motive zeigen. Deshalb fügt der IaprTopicLoader neben den Texten der Topic auch Informationen zu den Beispielbildern in der Topic an, so dass im späteren Retrieval-Prozess der IaprLuceneSearcher auf die Beispielbilder zugreifen kann.

3.3.2.3 Klasse: IaprLuceneSearcher

Die Klasse IaprLuceneSearcher ist eine Erweiterung der Klasse LuceneSearcher aus dem Package de.chemnitz.tu.informatik.ir.clef, die bereits für den CLEF Domain-Specific Track genutzt wurde. Zusätzlich zur Text-Suche mit Hilfe von Apache Lucene nutzt der IaprLuceneSearcher das Caliph&Emir Projekt (siehe [Lux 2004]) um die durch Apache Lucene gefundenen Bilder mit den Beispielbildern aus der Topic zu vergleichen und eine Anpassung des Rankings vorzunehmen.

3.3.2.4 Klasse: OOoThesaurusTopicFilter

Der OOoThesaurusTopicFilter dient zur Query Expansion. Er erweitert die Terme in den Feldern einer Topic um die Wörter, die in einem Thesaurus mit diesen Termen verbunden sind. Es wird der Standard-Thesaurus des Open-Source-Programms OpenOffice ab Version 2 verwendet. Dadurch stehen Thesauri in einer Vielzahl unterschiedlicher Sprachen zur Verfügung und können ohne Einschränkungen genutzt werden.

Da die Suche in einem Thesaurus eine Retrieval darstellt, wird für diese Suche ein Lucene-Index erstellt. Als Resultat ist die Suche, wenn einmal der Index erstellt ist, schneller und es findet eine Normalisierung der Terme statt. Außerdem ist eine unscharfe Suche mit Hilfe von

Wildcards oder Fuzzy-Suche möglich. Nach der Suche im Lucene-Index wird ein Schwellwert auf die Ergebnisse angewendet, so dass nur Terme mit einer größtmöglichen Relevanz für die Query Expansion verwendet werden.

4 Evaluation

Anhand des in Kapitel 3.3.2 beschriebenen Anwendungsfalles wird sowohl die Funktionalität das Framework, als auch die Kombination von Text Retrieval und Content-Based Image Retrieval Methoden durch die Teilnahme am ImageCLEFphoto-Tasks untersucht. Im Folgenden wird die Aufgabenstellung des ImageCLEFphoto-Tasks vorgestellt, der Versuchsaufbau und die verwendeten Komponenten dargelegt und die erzielten Ergebnisse präsentiert und analysiert.

4.1 ImageCLEFphoto

Der ImageCLEFphoto-Tasks entstand bereits 2003 als eigene Aufgabenstellung im Rahmen des ImageCLEF-Tracks der CLEF-Kampagne. Damals noch mit einem anderen Korpus, wird er seit 2006 in seiner jetzigen Form und mit dem IAPR-TC12-Korpus durchgeführt. Ziel des ImageCLEFphoto-Tasks ist es, die Möglichkeiten des mehrsprachigen Text-Retrievals mit Content-based Image Retrieval Methoden zu verbinden, was sich mit dem Ziel des Retrieval Frameworks, medienübergreifend arbeiten zu können, deckt.

4.1.1 Korpus

Der zur Verfügung gestellte Korpus ist der IAPR TC-12 Benchmark [Grubinger 2006] mit 20.000 Fotografien, die sowohl Menschen als auch Gebäude in einem touristischen oder sportlichen Kontext zeigen. Zu jedem Bild sind Annotationen in den Sprachen Deutsch, Englisch und Spanisch vorhanden.

Die Annotationen umfassen

- `<DOCNO>`: eine Dokument-Identifikation,
- `<TITLE>`: einen Titel,
- `<DESCRIPTION>`: einen beschreibenden Text, der jedoch im ImageCLEFphoto-Task im Jahr 2007 entfernt wurde und für die Experimente nicht mehr zur Verfügung stand,
- `<LOCATION>`: eine ungefähre Ortsangabe, wo das Foto aufgenommen wurde,
- `<DATE>`: das Aufnahmedatum und

- zusätzliche Angaben zur Ablage des assoziierten Bildes und des Thumbnails.

Inhalt der englischen Annotation mit dem Dokumentidentifikator „00/60“:

```
<DOC>
<DOCNO>annotations/00/60.eng</DOCNO>
<TITLE>Palma </TITLE>
<DESCRIPTION>two lane street with large shops on the
right and smaller shops on the left; people are walking
on the sidewalk, some are crossing the street; cars are
parked along the left side of the street as well;
</DESCRIPTION>
<NOTES>The main shopping street in Paraguay; </NOTES>
<LOCATION>Asunción, Paraguay </LOCATION>
<DATE>March 2002 </DATE>
<IMAGE>images/00/60.jpg </IMAGE>
<THUMBNAIL>thumbnails/00/60.jpg </THUMBNAIL>
</DOC>
```

Abbildung 14: Foto mit Dokument-ID "00/60"

Zusätzlich wurden Annotationen aus allen drei Sprachen zufällig gemischt, um so einen gemischtsprachigen Korpus zu erhalten, wobei jedoch keine direkte Information zur gewählten Sprache vorhanden ist.

4.1.2 Topics

Die Suchanfragen stehen in den Sprachen Chinesisch (vereinfacht und traditionell), Dänisch, Niederländisch, Englisch, Finnisch, Französisch, Deutsch, Italienisch, Japanisch, Norwegisch, Polnisch, Portugiesisch, Russisch, Spanisch und Schwedisch zur Verfügung.

Es gibt insgesamt 60 verschiedene Topics, die nach vorher festgelegten Kriterien erstellt und ausgewählt wurden (vgl. Kapitel 1.2.3). Für jede Topic wurden weiterhin drei Beispielbilder ausgewählt, die nicht Teil des Korpus sind. [Grubinger 2007]

Inhalt der Topic Nr. 55:

```
<top>
<num> Number: 55 </num>
<title> drawings in Peruvian deserts </title>
<narr> </narr>
<image> topics/55/11189.jpg </image>
<image> topics/55/14097.jpg </image>
<image> topics/55/16275.jpg </image>
</top>
```

Abbildung 15: Eines von 3 Beispielbildern für Topic Nr. 55

Durch den Retrieval-Prozess zeigt sich, dass sich die Topics in die beiden Klassen „schwierige“ und „einfache“ Topics unterteilen lassen. Unter die Klasse der schwierigen Topics fallen alle Topics, die sehr wenige oder keine relevanten Dokumente liefern. Im Gegensatz dazu ste-

hen die einfachen Topics, die außerordentlich viele oder alle relevanten Dokumente unter den am höchsten bewerteten Dokumenten ermitteln.

Zu den schwierigen Topics zählen unter anderem:

- **Nr. 2: Kirche mit mehr als zwei Türmen**

  Diese Topic besitzt durch die Formulierung „mehr als zwei“ einen erhöhten Schwierigkeitsgrad. Zum einen besitzen die meisten Kirchen einen oder zwei Türme, zum anderen ist für die korrekte Übersetzung in eine Query das Erkennen und Interpretieren dieser Formulierung nötig. Das setzt ein gewisses Sprachverständnis auf der Seite des Parsers voraus, was im aktuellen QueryParser von Apache Lucene nicht gegeben ist.

- **Nr. 18: Sportstation außerhalb Australiens**

  Hier stellt die Formulierung „außerhalb Australiens“ ähnlich wie die Formulierung „mehr als zwei“ in Topic Nr. 2 das Problem dar. Es werden durch die einfache Analyse der Topic hauptsächlich Stadien in Australien gefunden.

- **Nr. 49: Bilder von typisch australischen Tieren**

  Das Retrieval System besitzt nicht das nötige Allgemeinwissen, um aus australischen Tieren die konkreten Terme, die in den Annotationen verwendet wurden, zu generieren. Dies wäre unter Umständen mit einem Thesaurus, der diese Informationen enthält, möglich.

- **Nr. 54: Berühmte Fernseh- und Funktürme**

  Diese Topic liefert nur in deutscher Sprache außerordentlich schlechte Ergebnisse. Dies lässt sich eventuell durch die Unvollständigkeit des Wortes „Fernsehturm“ erklären. Da der Stemmer für Deutsch keine zusammengesetzten Wörter zerlegt, sind im Index relevante Dokumente nicht mit „Fernseh“ indiziert, sondern mit „Fernsehturm“.

Zu den einfachen Topics gehören unter anderem:

- **Nr. 10: Reiseziele in Venezuela**

  Der Topic-Titel ist sehr allgemein formuliert. Daraus ergibt sich ein breites Feld an möglicherweise relevanten Dokumenten, die in diesem konkreten Fall hauptsächlich durch den Term „Venezuela“ identifiziert werden können.

- **Nr. 11: Schwarzweißphotos von Russland**

  Auch hier ist ein sehr allgemeines Informationsbedürfnis klar formuliert. Der Term Russland schränkt die Auswahl der Dokumente bereits in besonderem Maße ein. Weiterhin beschreibt der Scalable Color Descriptor nach MPEG-7 auch die Farbsättigung, die bei den relevanten Bildern besonders niedrig ausfällt und damit ein markantes Kriterium für die Distanz zwischen den Beispielbildern und den relevanten Bildern darstellt.

- **Nr. 27: Motorradfahrer während des Australischen Motorrad Grand Prix**

  Die Wortgruppe „Australischen Motorrad Grand Prix“ verfügt, da es sich um einen Eigennamen eines Ereignisses handelt, über ein hohes Maß an Spezifität.

- **Nr. 36: Photos mit Machu Picchu im Hintergrund**

  Diese Topic ist auf der einen Seite wieder sehr allgemein formuliert, das heißt dass jedes Bild, auf dem der Machu Picchu unabhängig vom Kontext zu sehen ist, relevant ist. Auf der anderen Seite stellt der Eigenname „Machu Picchu“ ein diskriminierendes Merkmal dar.

4.1.3 Experimente

Die Retrieval-Experimente führen die Teilnehmer in Eigenregie durch. Es gibt hinsichtlich der anzuwendenden Verfahren keine Einschränkungen. Iterierende Prozesse mit Relevance Feedback oder verschiedene Methoden der Query Expansion sind erlaubt. Um eine Vergleichbarkeit zu gewährleisten werden alle Experimente mit Hilfe verschiedener, gemeinsamer Merkmale (siehe auch [Grubinger 2007]) beschrieben:

- **Sprache der Topics**

  Mögliche Werte: DA (Dänisch), DE (Deutsch), EN (Englisch), ES (Spanisch), FI (Finnisch), FR (Französisch), IT (Italienisch), JA (Japanisch), NL (Niederländisch), NO (Norwegisch), PL (Polnisch), PT (Portugiesisch), RU (Russisch), SV (Schwedisch), ZHS (Chinesisch - vereinfacht), ZHT (Chinesisch – traditionell)

- **Sprache der Annotationen**

  Mögliche Werte: DE (Deutsch), EN (Englisch), ES (Spanisch), RND (Zufällig), ALL (Alle Sprachen)

- **Experiment-Art**

  Alle Experimente fallen in eine von zwei Kategorien: automatisch oder manuell. Automatisch sind alle Experimente, die ohne Nutzerinteraktion oder sonstige von Menschen abhängigen Daten arbeiten. Alle anderen Experimente werden als manuelle Experimente eingeordnet. Das heißt, wenn das Feedback von einem Nutzer erfasst wird, handelt es sich um ein manuelles Experiment.

  Mögliche Werte: AUTO (Automatisch), MAN (Manuell)

- **Feedback / Query Expansion**

  Dieser Wert gibt an, ob ein Feedback und/oder eine Query Expansion genutzt wurde. Experimente ohne Feedback und Query Expansion werden im allgemeinen als Baseline bezeichnet.

  Mögliche Werte: FB (Feedback), QE (Query Expansion), FBQE (Feedback und Query Expansion), NOFB (kein Feedback und keine Query Expansion)

- **Modalität**

  Die Modalität beschreibt die für das Retrieval verwendeten Daten. Das Retrieval kann sich zum Beispiel nur auf die Textinformationen eines Dokuments stützen (TXT) oder nur auf die Bildinformationen (IMG).

  Mögliche Werte: TXT (nur Text), IMG (nur Bild), TXTIMG (Text und Bild)

4.1.4 Relevanzbewertung

Die Bewertung wurde mit einer ternären Klassifkation (relevant, teilweise relevant und nicht relevant) durchgeführt. Die zusätzliche dritte Klasse (teilweise relevant) dient zur besseren Bewertbarkeit von Bildern, in denen zum Beispiel das gewünschte Motiv im Bild vorhanden ist, aber im falschen Kontext steht. Die unterschiedlich kombinierten Bewertungen der Juro-

ren ergeben die endgültigen Relevanzbewertungen, die für die Bewertung der Experimente verwendet und im QRELS-Format zur Verfügung gestellt wurden.

4.2 Basiskonfiguration

Um den veränderten Anforderungen des ImageCLEFphoto-Tasks im Jahr 2007 Rechnung zu tragen wurden folgende Erweiterungen gegenüber den Versuchen beim ImageCLEFphoto-Task im Jahr 2006 vorgenommen und deren Auswirkungen auf das Ergebnis untersucht:

- Manuelles Feedback (siehe 3.2.1.1)
- Query Expansion mit Hilfe eines Thesaurus (siehe 3.3.2.4)
- Verwendung von MPEG-7 Deskriptoren aus Caliph&Emir [Lux 2004] zum Bildvergleich (siehe 3.3.2.3)

Die Untersuchung der Auswirkungen wird in zwei Aufgabenbereichen vorgenommen: dem monolingualen Retrieval, bei dem für die Suchanfragen die gleiche Sprache wie für den Index verwendet wird, und dem bilingualen Retrieval, bei dem die Suchanfragen in einer anderen Sprache, als der des zugrunde liegenden Korpus, vorliegen.

Als Index wird für jede Korpus-Sprache, also Englisch, Deutsch und Spanisch, ein Index mit Lucene angelegt, in dem die Annotationen indiziert und die durch Caliph&Emir erzeugten MPEG-7 Deskriptoren gespeichert werden. Als MPEG-7 Deskriptoren kommen der Scalable Color Descriptor und das Edge Histogram zum Einsatz.

Der Lucene-Analyzer zur Normalisierung der Terme basiert auf dem StandardAnalyzer, der im Lieferumfang von Lucene enthalten ist. Er wurde erweitert um einen Stopwort-Filter, der die Stopwörter aus einer anzugebenden Textdatei lädt. Als Stemmer kommt der Snowball-Filter mit konfigurierbarer Sprach-Einstellung zur Anwendung. Damit steht ein sehr flexibler und wiederverwendbarer Lucene-Analyzer zur Verfügung, der bei allen Versuchen wiederverwendet wurde.

Die verwendeten Stopwortlisten stammen von Jacques Savoy von der Universität Neuchatel[15]. Es stehen Stopwortlisten für alle eingesetzten Sprachen zur Verfügung.

15 http://members.unine.ch/jacques.savoy/clef/index.html - CLEF and Multilingual information retrieval

Zur Query Expansion kommt der OOoThesaurusTopicFilter zum Einsatz. Als Thesaurus wurden die entsprechenden Thesauri des Lingucomponent Projects[16] im Rahmen von OpenOffice.org verwendet. Diese Thesauri wurden mit Hilfe des zuvor beschriebenen Lucene-Analyzers indiziert und durchsucht (vgl. Kapitel 3.3.2.4).

Die Bezeichner der Experimente setzen sich aus der Topic-Sprache, der Index-Sprache (wenn von Topic-Sprache abweichend) und der Modalität zusammen. Experimente mit Query Expansion erhalten den Zusatz „qe“. Experimente mit Relevance Feedback erhalten den Zusatz „f“ und die Anzahl der für das Feedback betrachteten Treffer. Damit steht „en-de-qe-f20“ für ein Experiment in dem englischen Topics in einem deutschen Index unter Verwendung der Query Expansion und den ersten 20 Treffern für das Relevance Feedback gesucht wurden.

16 http://wiki.services.openoffice.org/wiki/Dictionaries - Dictionaries - OpenOffice.org Wiki

4.3 Monolinguale Versuche

Im ImageCLEFphoto-Task stehen die Annotationen der Bilder in drei unterschiedlichen Sprachen zur Verfügung: Englisch, Deutsch und Spanisch (siehe 1.2.2). Für alle drei Sprachen sind alle Kombinationen der Erweiterungen auf ihre Verbesserung gegenüber der Baseline (ohne Erweiterungen) untersucht worden. In den folgenden Abschnitten werden die Ergebnisse vorgestellt und analysiert.

4.3.1 Englisch

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
en-txt	Text	Nein	-	0,1357	0,0000
en-txt-fb20	Text	Nein	20	0,2058 (+52%)	0,0000
en-txt-qe	Text	Ja	-	0,1295 (-5%)	0,0315
en-txt-qe-fb20	Text	Ja	20	0,1962 (+45%)	0,0607 (+93%)
en-mix	Mixed	Nein	-	0,1682 (+24%)	0,0000
en-mix-fb20	Mixed	Nein	20	0,2645 (+95%)	0,0000
en-mix-qe	Mixed	Ja	-	0,1653 (+22%)	0,0500 (+59%)
en-mix-qe-fb20	Mixed	Ja	20	0,2908 (+114%)	0,1077 (+232%)

Tabelle 1: Ergebnisse für monolingual Englisch

Das Experiment mit der Bezeichnung „en-txt“ stellt die Baseline dar und bildet die Grundlage für den Vergleich mit den anderen Versuchen. Als Erstes fällt auf, dass für alle Experimente ohne Query Expansion die Geometic Mean Average Precision (GMAP) gleich 0 (Null) ist, was ein Hinweis darauf ist, dass zu mindestens einer Topic keine relevanten Treffer gefunden wurden. Mit Hilfe der Query Expansion wird die GMAP im einfachsten Fall – also für nur Text und ohne Feedback – auf 0,0315 verbessert, jedoch sinkt die Mean Average Precision (MAP) um 5% auf 0,1295. In Verbindung mit dem manuellen Feedback über die ersten 20 angezeigten Dokumente steigt die MAP ohne Query Expansion um 52% auf 0,2058, dagegen steigt die MAP bei der Verwendung der Query Expansion gegenüber der Baseline nur um 45% auf 0,1962 und damit niedriger als ohne Query Expansion.

Wenn anhand der MPEG7-Deskriptoren ein Reranking der Ergebnisse vorgenommen wird, dann verbessert sich das Ergebnis um ca. 20% gegenüber den Experimenten ohne Bildinfor-

mationen. Auch hier verschlechtert die Query Expansion das Ergebnis im ersten Schritt, jedoch liegt das Ergebnis nach dem manuellen Feedback höher, als das Ergebnis ohne Query Expansion.

Das am schlechtesten ausgefallene Experiment ist die Baseline mit einer Mean Average Precision (MAP) von 0,1357 und einer Geometric Mean Average Precision (GMAP) von 0,0000 und das beste Experiment ist erwartungsgemäß das, bei dem alle Erweiterungen zum Einsatz kamen, mit einer MAP von 0,2908, was eine Verbesserung gegenüber der Baseline von 114% ist, und einer GMAP von 0,1077.

4.3.2 Deutsch

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
de-txt	Text	Nein	-	0,0901	0,0000
de-txt-fb20	Text	Nein	20	0,1600 (+78%)	0,0000
de-txt-qe	Text	Ja	-	0,0935 (+4%)	0,0000
de-txt-qe-fb20	Text	Ja	20	0,1688 (+87%)	0,0000
de-mix	Mixed	Nein	-	0,1271 (+41%)	0,0000
de-mix-fb20	Mixed	Nein	20	0,2120 (+135%)	0,0000
de-mix-qe	Mixed	Ja	-	0,1241 (+38%)	0,0000
de-mix-qe-fb20	Mixed	Ja	20	0,2288 (+154%)	0,0000

Tabelle 2: Ergebnisse für monolingual Deutsch

Der Unterschied zwischen Baseline (de-txt) und dem Experiment unter Verwendung aller Erweiterungen (de-mix-qe-fb20) ist mit einer Steigerung von 154% größer als bei der englischer Sprache (nur 114%). Im Allgemeinen sind die Ergebnisse aber schlechter.

4.3.3 Spanisch

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
es-txt	Text	Nein	-	0,1208	0,0000
es-txt-fb20	Text	Nein	20	0,1815 (+50%)	0,0000
es-txt-qe	Text	Ja	-	0,1255 (+4%)	0,0000
es-txt-qe-fb20	Text	Ja	20	0,1904 (+58%)	0,0000
es-mix	Mixed	Nein	-	0,1601 (+33%)	0,0000
es-mix-fb20	Mixed	Nein	20	0,2675 (+121%)	0,0000
es-mix-qe	Mixed	Ja	-	0,1600 (+33%)	0,0000
es-mix-qe-fb20	Mixed	Ja	20	0,3059 (+153%)	0,0000

Tabelle 3: Ergebnisse für monolingual Spanisch

Die Experimente zum monolingualen Retrieval in spanischer Sprache ergaben ein ähnliches Ergebnis wie die Versuche in deutscher Sprache. Das Ergebnis des Experiments „es-mix-qe-fb20“ liegt über dem Ergebnis des Experiments „en-mix-qe-fb20“ und ist sogar das beste Ergebnis der gesamten Experimentierreihe.

4.4 Bilinguale Versuche

Wie in Kapitel 4.1.2 beschrieben stehen im Rahmen des ImageCLEFphoto-Tasks die Topics in folgenden Sprachen zur Verfügung: Chinesisch (vereinfacht), Chinesisch (traditionell), Dänisch, Niederländisch, Englisch, Finnisch, Französisch, Deutsch, Italienisch, Norwegisch, Polnisch, Portugiesisch, Russisch, Spanisch und Schwedisch.

Da zur Übersetzung des Online-Übersetzungsdienst von Google verwendet wurde, konnten nur die dort zur Verfügung stehenden Sprachkombinationen getestet werden, die in den folgenden Abschnitten untersucht werden.

4.4.1 Deutsche Topics im englischen Index

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
de-en-txt	Text	Nein	-	0,1159	0,0000
de-en-txt-fb20	Text	Nein	20	0,1964 (+69%)	0,0000
de-en-txt-qe	Text	Ja	-	0,1097 (-5%)	0,0000
de-en-txt-qe-fb20	Text	Ja	20	0,1859 (+60%)	0,0000
de-en-mix	Mixed	Nein	-	0,1542 (+33%)	0,0000
de-en-mix-fb20	Mixed	Nein	20	0,2538 (+119%)	0,0000
de-en-mix-qe	Mixed	Ja	-	0,1506 (+30%)	0,0000
de-en-mix-qe-fb20	Mixed	Ja	20	0,2779 (+140%)	0,0000

Tabelle 4: Ergebnisse für deutsche Topics im englischen Index

Die Ergebnisse dieser Reihe sind, wider Erwarten, durchgehend besser als die der monolingualen Experimente für die deutsche Sprache.

4.4.2 Spanische Topics im englischen Index

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
es-en-txt	Text	Nein	-	0,1197	0,0000
es-en-txt-fb20	Text	Nein	20	0,1904 (+59%)	0,0000
es-en-txt-qe	Text	Ja	-	0,1217 (+2%)	0,0000
es-en-txt-qe-fb20	Text	Ja	20	0,1914 (+60%)	0,0000
es-en-mix	Mixed	Nein	-	0,1618 (+35%)	0,0000
es-en-mix-fb20	Mixed	Nein	20	0,2571 (+115%)	0,0000
es-en-mix-qe	Mixed	Ja	-	0,1606 (+34%)	0,0000
es-en-mix-qe-fb20	Mixed	Ja	20	0,2975 (+149%)	0,0000

Tabelle 5: Ergebnisse für spanische Topics im englischen Index

Das Ergebnis der maximalen Konfiguration ist hier besser als das Ergebnis der maximalen Konfiguration im monolingual Englisch Experiment. Als mögliche Ursache kommt der Thesaurus in Frage, der für die aus dem Spanischen ins Englische übersetzten Wörter passendere Synonyme liefert, als für die originalen englischen Wörter.

4.4.3 Französische Topics im englischen Index

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
fr-en-txt	Text	Nein	-	0,1222	0,0000
fr-en-txt-fb20	Text	Nein	20	0,1874 (+53%)	0,0000
fr-en-txt-qe	Text	Ja	-	0,1272 (+4%)	0,0000
fr-en-txt-qe-fb20	Text	Ja	20	0,1945 (+59%)	0,0000
fr-en-mix	Mixed	Nein	-	0,1671 (+37%)	0,0000
fr-en-mix-fb20	Mixed	Nein	20	0,2305 (+89%)	0,0000
fr-en-mix-qe	Mixed	Ja	-	0,1535 (+26%)	0,0000
fr-en-mix-qe-fb20	Mixed	Ja	20	0,2644 (+116%)	0,0000

Tabelle 6: Ergebnisse für französische Topics im englischen Index

Bei den Experimenten mit französischen Topics und einem englischen Index traten keine Besonderheiten auf. Alle Werte bewegen sich im erwarteten Bereich.

4.4.4 Italienische Topics im englischen Index

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
it-en-txt	Text	Nein	-	0,0998	0,0000
it-en-txt-fb20	Text	Nein	20	0,1677 (+68%)	0,0000
it-en-txt-qe	Text	Ja	-	0,1032 (+3%)	0,0000
it-en-txt-qe-fb20	Text	Ja	20	0,1630 (+63%)	0,0000
it-en-mix	Mixed	Nein	-	0,1315 (+32%)	0,0000
it-en-mix-fb20	Mixed	Nein	20	0,2321 (+133%)	0,0000
it-en-mix-qe	Mixed	Ja	-	0,1287 (+29%)	0,0000
it-en-mix-qe-fb20	Mixed	Ja	20	0,2595 (+160%)	0,0000

Tabelle 7: Ergebnisse für italienische Topic im englischen Index

Auch bei den Experimenten mit italienischen Topics und dem englischen Index sind keine Besonderheiten zu erkennen. Es ist lediglich anzumerken, das sich um das schlechteste Ergebnis für den englischen Index handelt.

4.4.5 Chinesische (vereinfacht) Topics im englischen Index

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
zhs-en-txt	Text	Nein	-	0,1122	0,0000
zhs-en-txt-fb20	Text	Nein	20	0,1772 (+58%)	0,0000
zhs-en-txt-qe	Text	Ja	-	0,1094 (-2%)	0,0000
zhs-en-txt-qe-fb20	Text	Ja	20	0,1697 (+51%)	0,0000
zhs-en-mix	Mixed	Nein	-	0,1377 (+23%)	0,0000
zhs-en-mix-fb20	Mixed	Nein	20	0,2405 (+114%)	0,0000
zhs-en-mix-qe	Mixed	Ja	-	0,1307 (+16%)	0,0000
zhs-en-mix-qe-fb20	Mixed	Ja	20	0,2863 (+155%)	0,0000

Tabelle 8: Ergebnisse für chinesische (vereinfacht) Topics im englischen Index

Trotz der Andersartigkeit der chinesischen Schrift im Vergleich zu den Schreibformen der anderen hier verwendeten Sprachen sind die erzielten guten Ergebnisse hervorzuheben. Die Baseline übertrifft die Experimente mit spanischen Topics im englischen Index und die maximale Konfiguration übertrifft alle nicht-spanischen, bilingualen Experimente.

4.4.6 Englische Topics im deutschen Index

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
en-de-txt	Text	Nein	-	0,0651	0,0000
en-de-txt-fb20	Text	Nein	20	0,1323 (+103%)	0,0000
en-de-txt-qe	Text	Ja	-	0,0678 (+4%)	0,0000
en-de-txt-qe-fb20	Text	Ja	20	0,1337 (+105%)	0,0000
en-de-mix	Mixed	Nein	-	0,0972 (+49%)	0,0000
en-de-mix-fb20	Mixed	Nein	20	0,1820 (+180%)	0,0000
en-de-mix-qe	Mixed	Ja	-	0,0939 (+44%)	0,0000
en-de-mix-qe-fb20	Mixed	Ja	20	0,1943 (+198%)	0,0000

Tabelle 9: Ergebnisse für englische Topics im deutschen Index

Die Ergebnisse für die Experimente mit englischen Topics und dem deutschen Index sind die zweitschlechtesten aller Experimente.

4.4.7 Französische Topics im deutschen Index

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
fr-de-txt	Text	Nein	-	0,0557	0,0000
fr-de-txt-fb20	Text	Nein	20	0,1131 (+103%)	0,0000
fr-de-txt-qe	Text	Ja	-	0,0550 (-1%)	0,0000
fr-de-txt-qe-fb20	Text	Ja	20	0,0987 (+77%)	0,0000
fr-de-mix	Mixed	Nein	-	0,0911 (+64%)	0,0000
fr-de-mix-fb20	Mixed	Nein	20	0,1592 (+186%)	0,0000
fr-de-mix-qe	Mixed	Ja	-	0,0915 (+64%)	0,0000
fr-de-mix-qe-fb20	Mixed	Ja	20	0,1697 (+205%)	0,0000

Tabelle 10: Ergebnisse für französische Topics im deutschen Index

Die Experimente, in denen französische Topics im deutschen Index gesucht wurden, sind die schlechtesten der gesamten Experimentierreihe. Es fällt aber auf, das die Verbesserungen durch das Feedback hier besonders hoch ausgefallen sind.

4.4.8 Englische Topics im spanischen Index

Bezeichnung	*Modalität*	*Query Expansion*	*Feedback*	*MAP*	*GMAP*
en-es-txt	Text	Nein	-	0,1199	0,0000
en-es-txt-fb20	Text	Nein	20	0,1825 (+52%)	0,0000
en-es-txt-qe	Text	Ja	-	0,1267 (+6%)	0,0000
en-es-txt-qe-fb20	Text	Ja	20	0,1993 (+66%)	0,0000
en-es-mix	Mixed	Nein	-	0,1582 (+32%)	0,0000
en-es-mix-fb20	Mixed	Nein	20	0,2686 (+124%)	0,0000
en-es-mix-qe	Mixed	Ja	-	0,1592 (+33%)	0,0000
en-es-mix-qe-fb20	Mixed	Ja	20	0,3057 (+155%)	0,0000

Tabelle 11: Ergebnisse für englische Topics im spanischen Index

Das Besondere an diesem Versuchen ist, dass der Versuch „en-es-mix-qe-fb20“ mit einem MAP von 0,3057 fast den MAP des besten monolingualen Versuchs „es-mix-qe-fb20“ mit 0,3059 erreicht. Versuch „en-es-mix-fb20“ (MAP = 0,2686) erreicht sogar ein besseres Ergebnis, als der beste vergleichbare monolinguale Versuch „es-mix-fb20“ (MAP = 0,2675).

4.5 Auswertung

Die Versuche haben gezeigt, dass die Summe der Verbesserungen das Ergebnis entscheidend verbessern kann. Das größte Potential geht dabei vom manuellen Feedback aus. An zweiter Stelle folgt das Reranking anhand der MPEG7-Deskriptoren und an letzter Stelle die Query Expansion mit Hilfe eines Thesaurus.

Das manuelle Feedback verbessert das Ergebnis um so mehr, je niedriger das Ausgangsergebnis ist. Dies lässt sich gut am direkten Vergleich des besten und des schlechtesten Experiments erkennen:

Bezeichnung	*es-...*	*fr-de-...*
...-txt	0,1208	0,0557
...-txt-fb20	0,1815 (+50%)	0,1131 (+103%)
...-mix-qe	0,1600 (+33%)	0,0915 (+64%)
...-mix-qe-fb20	0,3059 (+153%)	0,1697 (+205%)

Tabelle 12: Vergleich Monolingual Spanisch - Französchische Topic in deutschem Index

Die Verwendung der Bildinformationen in Form von MPEG7-Deskriptoren für ein Reranking der Treffer führt stets zu einer Verbesserung der Ergebnisse.

Bei Experimenten ohne manuelles Feedback kann die Thesaurus-basierte Query Expansion zu einer Aufwertung der Ergebnisse beitragen, jedoch nicht in jedem Fall. Beispiele für den gegenteiligen Effekt sind die Experimente für „monolingual Englisch“ und für die französischen Topics im deutschen Index. Für Experimente mit Feedback erreicht die Query Expansion immer eine Verbesserung des Ergebnisses.

5 Zusammenfassung und Ausblick

In der Zusammenfassung werden die praktischen Ergebnisse dieser Arbeit den Zielen gegenübergestellt. Der Ausblick zeigt Ideen für Weiterentwicklungen und weitere Anwendungen des Frameworks auf.

5.1 Zusammenfassung

In der vorliegenden Arbeit wurden auf Grundlage der Schwachstellen eines Vorgängersystems die Anforderungen an ein neues Retrieval Framework formuliert. In der Folge entstand das in Kapitel 3 beschriebene Retrieval Framework, das die für zukünftige Projekte notwendige Flexibilität gewährleistet.

Da ein Framework selbst kein fertiges Programm ist, das durch einen Anwender getestet werden kann, wurden zwei weitere Komponenten entwickelt, die im Rahmen der CLEF-Kampagne getestet werden konnten: eine Komponente für den Domain-Specific-Track und eine Komponente für den ImageCLEF Photographic Retrieval Task.

Beide Komponenten arbeiten zwar nach wie vor mit Apache Lucene, jedoch ist Lucene nur noch für die Indizierung und die Suche im erstellten Index zuständig. Beim Suchen selbst bildet das Formulieren der Query die einzige Lucene-Abhängigkeit. Damit ist Punkt 1 der Ziele aus Kapitel 2.2, die freie Wahl des Retrieval Systems, erfüllt.

Zwischen der Klasse IaprDataCollection und der Klasse LuceneIndexer gibt es keine direkte Beziehung. Damit geschieht das Auslesen des Korpus völlig unabhängig vom verwendeten Retrieval System. Es besteht eine Unabhängigkeit zwischen Korpus und Indizierung, siehe Punkt 2 Kapitel 2.2.

Das Framework bietet keine besondere Unterstützung für multilinguale Experimente, wie es in Punkt 3 der Ziele formuliert ist, gleicht dies aber durch den hohen Grad an Flexibilität aus. Die Klasse TranslationTopicFilter (Kapitel 3.3.1.5) bietet die Möglichkeit Topics vor der Suche zu übersetzen. Außerdem besitzen Topics keine festgelegten Felder mehr, damit kann für jede Sprache ein eigenes Feld in einer Topic angelegt werden.

Die Funktionen zur Evaluation, Punkt 4 der Ziele, wurden gegenüber dem Altsystem weiter ausgebaut, so sind nun mehrere Experimente mit Hilfe des Vergleichsgraphen oder des Box-Plot direkt miteinander vergleichbar.

Die Nutzung intelligente Kontrollelement der Benutzeroberfläche und der Reflection-API von Java konnte den Konfigurationaufwand erheblich reduzieren (siehe Punkt 5 der Ziele). Alle Komponenten sind nun über die gleiche Oberfläche konfigurierbar.

Funktionen zum direkten Laden und Speichern im XML-Format, aber auch die Funktionen zum Importieren und Exportieren von Ergebnislisten im TREC-Standard ermöglichen eine Sicherung von durchgeführten Experimenten, wie im Punkt 6 der Ziele beschrieben ist.

Alle im Kapitel 2.2 definierten Ziele sind somit weitestgehend erreicht, im Fall der verbesserten Unterstützung für multilinguale Experimente (Punkt 4) jedoch mit Einschränkungen.

Die Evaluation der Komponenten für den ImageCLEFphoto-Task hat gezeigt, dass das auf Lucene basierende Feedback, die auf dem OpenOffice-Thesaurus basierende Query Expansion und das Reranking der Ergebnisse anhand der MPEG-7 Deskriptoren eine Verbesserung der Ergebnisse erzielen.

5.2 Ausblick

Neben Lucene als Retrieval System existieren weitere Systeme, die in das Framework eingebunden und evaluiert werdern könnten. Zu diesen Systemen zählen Terrier [Ounis 2007] und Lemur [Lemur 2007]. Weiterhin ist ein Ausbau der durch Caliph&Emir [Lux 2004] im Ansatz vorhandenen Multimedia-Fähigkeiten möglich, zum Beispiel durch die Anbindung oder Integration eines Multimedia-Analyseframeworks, das unter anderem Funktionen zur Audio- und Video-Analyse, Sprechererkennung, Spracherkennung oder Objekterkennung bereitstellt.

Die Implementierung von Schnittstellen für weitere Retrieval Systeme macht eine Restrukturierung der Pakete aus Kapitel 3.3.1 notwendig. Die Schnittstelle zu Lucene, wie auch die der anderen Retrieval Systeme, wird als eigenständiges Package ausgelagert, das sowohl von CLEF als auch von ImageCLEF genutzt werden kann. Das führt zu einer einfacheren Einbindung anderer Retrieval Systeme in die einzelnen Projekte und verringert die Redundanz.

Es besteht bereits die Möglichkeit mit der grafischen Benutzeroberfläche oder dem automatischen Feedback ein negatives Feedback zu erfassen, jedoch fehlt der Klasse LuceneFeedbackSearcher noch die Funktionalität, dies sinnvoll in eine Query umzusetzen. Eine praktische Umsetzung könnte für Topics, die keine relevanten Ergebnisse liefern, unter Umständen zu vereinzelten relevanten Ergebnissen führen, die dann wieder für das gut funktionierende positive Feedback genutzt werden können.

Zusammengesetzte Wörter der deutschen Sprache stellen für alle verwendeten Stemmer ein Problem dar. Es wäre zu untersuchen, ob einfache oder spezielle n-Gram-Stemmer [McNamee 2004] hier Abhilfe schaffen können oder ob auf ein wörterbuchbasiertes Stemming zurückgegriffen werden sollte.

TopicFilter haben sich als besonders nützliches Werkzeug erwiesen. Die Entwicklung von Dokumentenfiltern, die ähnlich der TopicFilter die Dokumente vorverarbeiten, könnte eine sinnvolle Erweiterung des Frameworks sein.

Eine Relevanzbewertung ist bereits durch die grafische Oberfläche möglich. Es fehlt jedoch ein Assistent, der das Pooling und das Zusammenführen der Relevanzbewertungen mit den entsprechenden Algorithmen ermöglicht. Außerdem könnten die Möglichkeiten der Oberfläche entsprechend neuer Anwendungsszenarien erweitert werden.

Literaturverzeichnis

Amati, G.; van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems 20(4)*, S. 357-389.

Blei, D. M.; Ng, A. Y.; Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research 3*, S. 993-1022.

CIPA (2008). *2008-2010 Outlook on the Shipment Forecast by Product-Type Concerning Cameras and Related Goods*. Abgerufen am 22. April 2008 von http://www.cipa.jp/english/pdf/press080129_e.pdf.

Cleverdon, C. (1970). *The effect of variations in relevance assessments in comparative experimental tests of index languages*. Cranfield Library report No. 3. Cranfield University; Aslib.

Fuhr, N. (2004). *Information Retrieval - Skriptum zur Vorlesung im SS 04*. Abgerufen am 13. 3. 2008 von http://www.is.informatik.uni-duisburg.de/courses/ir_ss04/folien/irskall.pdf.

Gilbert, G.; Sparck-Jones, K. (1979). *Statistical bases of relevance assessment for the 'Ideal' information retrievaltest collection*. BL R&D Report 5481. University of Cambridge.

Google (2008). *Google Technology*. Abgerufen am 9. April 2008 von http://www.google.com/technology/.

Grubinger, M. (2007). *ImageCLEF 2007 photographic retrieval task*. Abgerufen am 14. 3. 2008 von http://eureka.vu.edu.au/~grubinger/ImageCLEFphoto2007/adhoc.htm.

Grubinger, M.; Clough, P.; Müller, H.; Deselears, T. (2006). The IAPR TC-12 Benchmark - A New Evaluation Resource for Visual Information Systems. In: *Proceedings of the International Workshop OntoImage'2006 Language Resources for Content-Based Image Retrieval*, S. 13-23.

Gulli, A.; Signorini, A. (2005). *The Indexable Web is more than 11.5 billion pages*. Abgerufen am 8. April 2008 von http://www.cs.uiowa.edu/~asignori/web-size/.

Kando, N. (2001). *NII NACSIS-Test Collection for IR Home Page*. Abgerufen am 13. 3. 2008 von http://research.nii.ac.jp/ntcir/outline/prop-en.html.

Kent, A.; Berry, M.; Leuhrs, F. U.; Perry, J. W. (1955). Operational criteria for designing information retrieval systems. *Machine literature searching VIII*, S. 93-101.

Kürsten, J.; Eibl, M. (2007). *Domain-Specific Cross Language Retrieval: Comparing and Merging Structured and Unstructured Indices*. Abgerufen am 23. April 2008 von http://www.clef-campaign.org/2007/working_notes/kuerstenCLEF2007.pdf.

Lemur (2007). *The Lemur Toolkit for Language Modeling and Information Retrieval*. Abgerufen am 23. April 2008 von http://www.lemurproject.org/.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In: *Machine Learning: ECML-98*, S. 4-15.

Lux, M.; Klieber, W.; Granitzer, M. (2004). Caliph & Emir: Semantics in MultimediaRetrieval and Annotation. In: *Proceedings of the 19th International CODATA Conference2004: The Information Society: New Horizons for Science*, S. 64-75.

Manning, C. D.; Raghavan, P.; Schütze, H. (2008). *An Introduction to Information Retrieval*. Cambridge University Press.

McNamee, P.; Mayfield, J. (2004). Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval 7(1-2)*, S. 73-97.

Ounis, I.; Lioma, C.; Macdonald, C.; Plachouras, V. (2007). Research Directions in Terrier: a Search Engine for Advanced Retrieval on the Web. *Novatica/UPGRADE Special Issue on Next Generation Web Search, 8(1)*, S. 49-56.

Page, L.; Brin, S.; Motwani, R.; Winograd, T. (1998). *The PageRank Citation Ranking: Bringing Order to the Web*.

Raghavan, V.; Wong, S. (1986). A Critical Analysis of Vector Space Model for Information Retrieval. *Journal of the American Society for Information Science 37(5)*, S. 279-287.

Robertson, S. E.; Walker, S.; Hancock-Beaulieu, M. M. (1995). Large test collection experiments on an operational, interactive system: Okapi at TREC. *Information Processing & Management*, S. 345-360.

Salton, G. (Hrsg.) (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall.

Salton, G.; McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Saracevic, T. (1995). Evaluation of evaluation in information retrieval. In: *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, S. 138-146.

TREC (2005). *Text REtrieval Conference (TREC) Data - English Relevance Judgements Files List*. Abgerufen am 23. April 2008 von http://trec.nist.gov/data/qrels_eng/.

TREC (2008). *Text REtrieval Conference (TREC) Overview*. Abgerufen am 23. April 2008 von http://trec.nist.gov/overview.html.

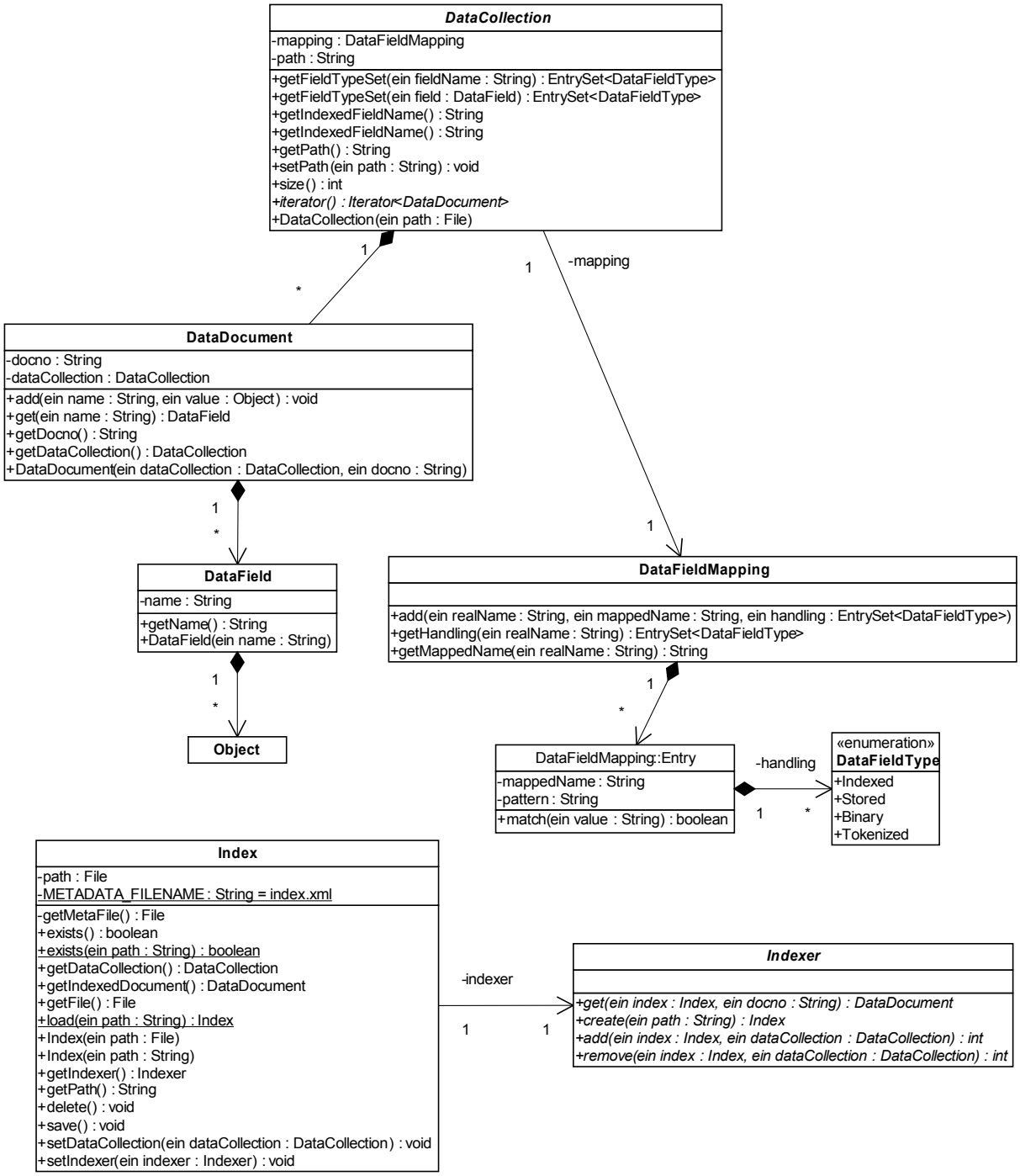
Tukey, J. W. (1977). Box-and-whisker plots. In: *Exploratory Data Analysis*, S. 39-43. Addison Wesley.

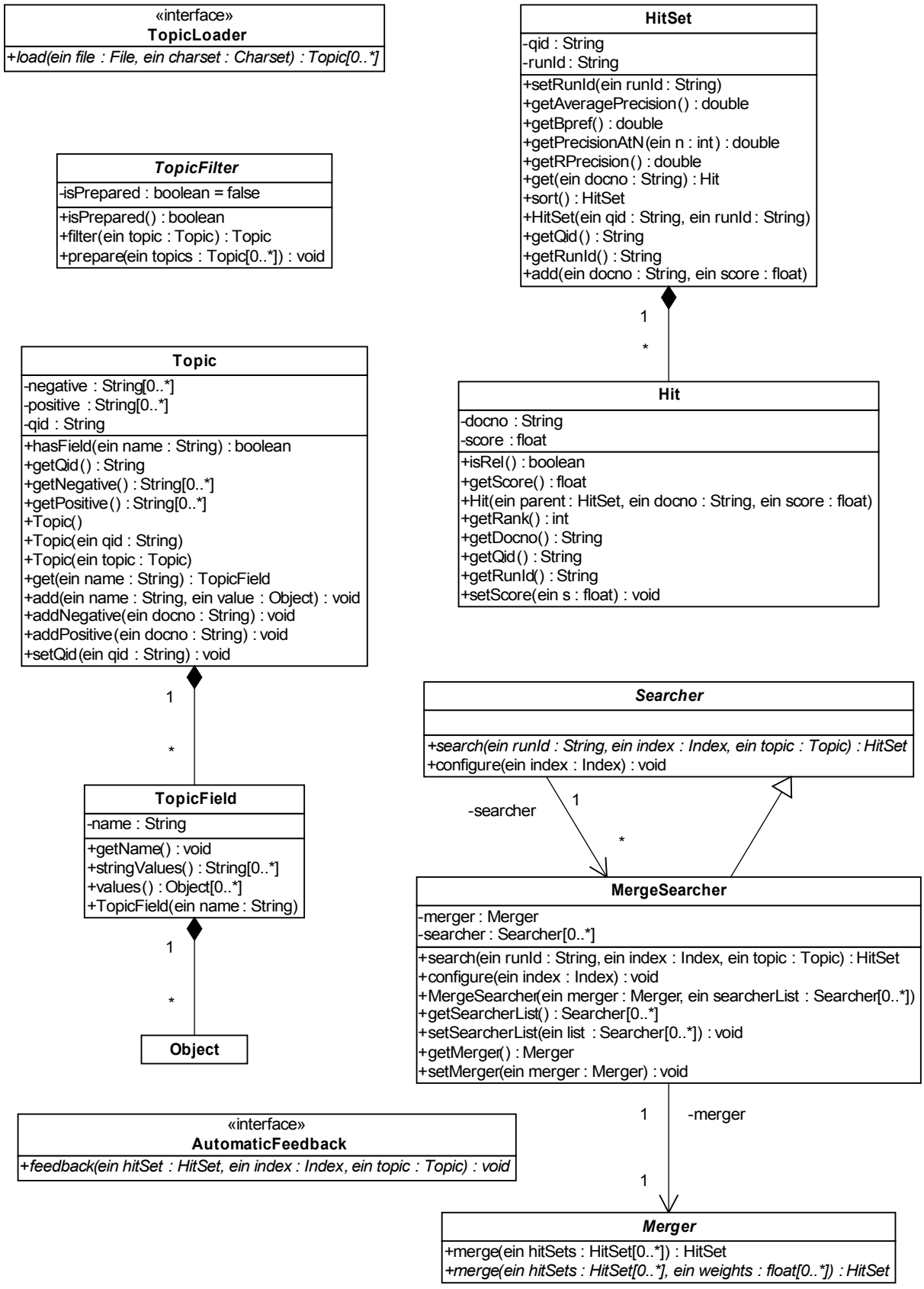
van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths.

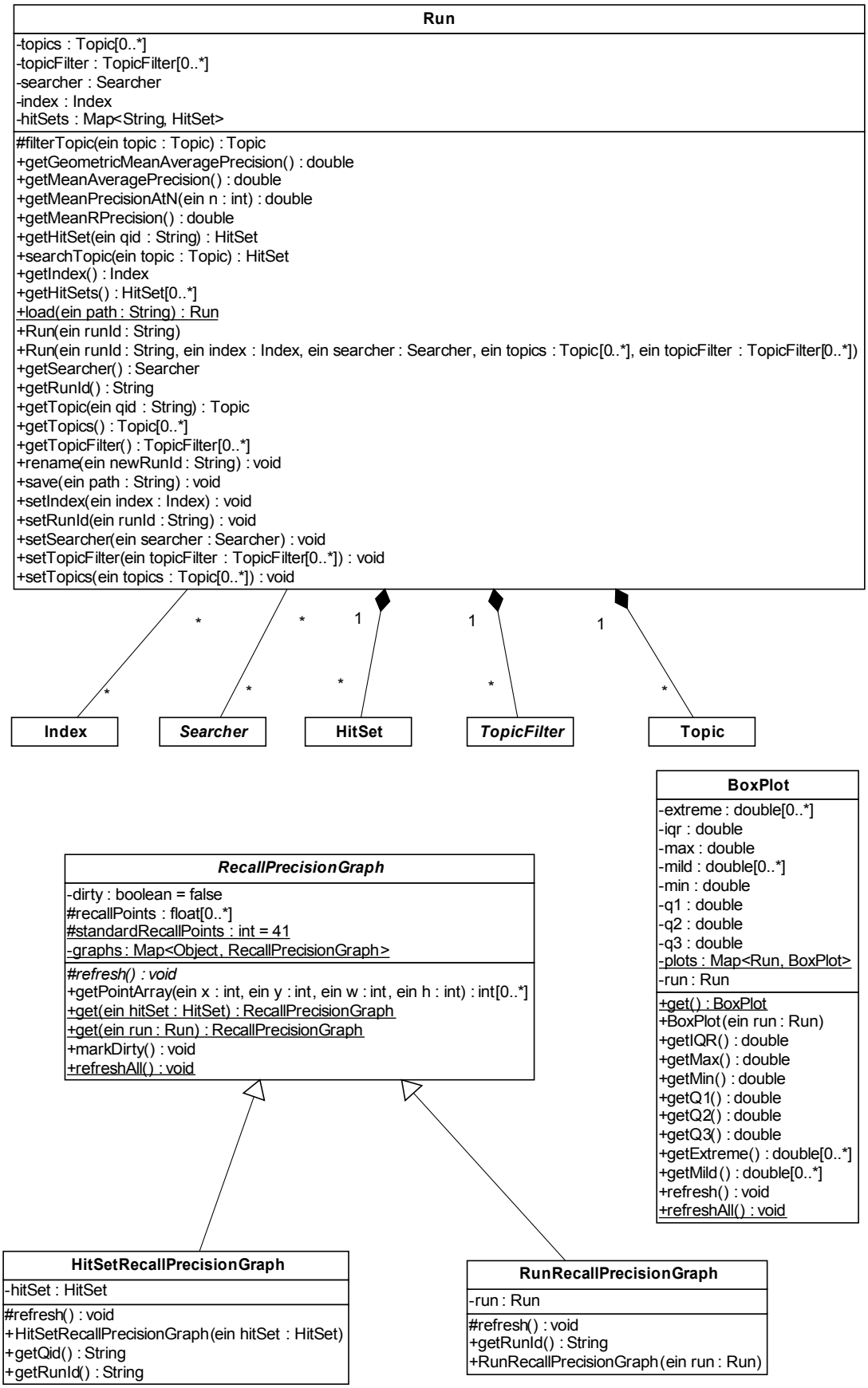
Vasiliev, A. (2008). *Frontpage - Lucene-java Wiki*. Abgerufen am 23. April 2008 von http://wiki.apache.org/lucene-java/FrontPage.

Wong, S. K. M.; Yao, Y. Y. (1995). On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems 13(1)*, S. 38-68.

Anhang A – Vollständige UML-Klassendiagramme des Frameworks




«interface»
TopicLoader
+load(ein file : File, ein charset : Charset) : Topic[0..*]
TopicFilter
-isPrepared : boolean = false
+isPrepared() : boolean
+filter(ein topic : Topic) : Topic
+prepare(ein topics : Topic[0..*]) : void
HitSet
-qid : String
-runId : String
+setRunId(ein runId : String)
+getAveragePrecision() : double
+getBpref() : double
+getPrecisionAtN(ein n : int) : double
+getRPrecision() : double
+get(ein docno : String) : Hit
+sort() : HitSet
+HitSet(ein qid : String, ein runId : String)
+getQid() : String
+getRunId() : String
+add(ein docno : String, ein score : float)
1
*
Hit
-docno : String
-score : float
+isRel() : boolean
+getScore() : float
+Hit(ein parent : HitSet, ein docno : String, ein score : float)
+getRank() : int
+getDocno() : String
+getQid() : String
+getRunId() : String
+setScore(ein s : float) : void
Topic
-negative : String[0..*]
-positive : String[0..*]
-qid : String
+hasField(ein name : String) : boolean
+getQid() : String
+getNegative() : String[0..*]
+getPositive() : String[0..*]
+Topic()
+Topic(ein qid : String)
+Topic(ein topic : Topic)
+get(ein name : String) : TopicField
+add(ein name : String, ein value : Object) : void
+addNegative(ein docno : String) : void
+addPositive(ein docno : String) : void
+setQid(ein qid : String) : void
1
*
TopicField
-name : String
+getName() : void
+stringValues() : String[0..*]
+values() : Object[0..*]
+TopicField(ein name : String)
1
*
Object
Searcher
+search(ein runId : String, ein index : Index, ein topic : Topic) : HitSet
+configure(ein index : Index) : void
-searcher
1
*
MergeSearcher
-merger : Merger
-searcher : Searcher[0..*]
+search(ein runId : String, ein index : Index, ein topic : Topic) : HitSet
+configure(ein index : Index) : void
+MergeSearcher(ein merger : Merger, ein searcherList : Searcher[0..*])
+getSearcherList() : Searcher[0..*]
+setSearcherList(ein list : Searcher[0..*]) : void
+getMerger() : Merger
+setMerger(ein merger : Merger) : void
1
-merger
1
«interface»
AutomaticFeedback
+feedback(ein hitSet : HitSet, ein index : Index, ein topic : Topic) : void
Merger
+merge(ein hitSets : HitSet[0..*]) : HitSet
+merge(ein hitSets : HitSet[0..*], ein weights : float[0..*]) : HitSet


Run
-topics : Topic[0..*]
-topicFilter : TopicFilter[0..*]
-searcher : Searcher
-index : Index
-hitSets : Map<String, HitSet>
#filterTopic(ein topic : Topic) : Topic
+getGeometricMeanAveragePrecision() : double
+getMeanAveragePrecision() : double
+getMeanPrecisionAtN(ein n : int) : double
+getMeanRPrecision() : double
+getHitSet(ein qid : String) : HitSet
+searchTopic(ein topic : Topic) : HitSet
+getIndex() : Index
+getHitSets() : HitSet[0..*]
+load(ein path : String) : Run
+Run(ein runId : String)
+Run(ein runId : String, ein index : Index, ein searcher : Searcher, ein topics : Topic[0..*], ein topicFilter : TopicFilter[0..*])
+getSearcher() : Searcher
+getRunId() : String
+getTopic(ein qid : String) : Topic
+getTopics() : Topic[0..*]
+getTopicFilter() : TopicFilter[0..*]
+rename(ein newRunId : String) : void
+save(ein path : String) : void
+setIndex(ein index : Index) : void
+setRunId(ein runId : String) : void
+setSearcher(ein searcher : Searcher) : void
+setTopicFilter(ein topicFilter : TopicFilter[0..*]) : void
+setTopics(ein topics : Topic[0..*]) : void
Index
Searcher
HitSet
TopicFilter
Topic
BoxPlot
-extreme : double[0..*]
-iqr : double
-max : double
-mild : double[0..*]
-min : double
-q1 : double
-q2 : double
-q3 : double
-plots : Map<Run, BoxPlot>
-run : Run
+get() : BoxPlot
+BoxPlot(ein run : Run)
+getIQR() : double
+getMax() : double
+getMin() : double
+getQ1() : double
+getQ2() : double
+getQ3() : double
+getExtreme() : double[0..*]
+getMild() : double[0..*]
+refresh() : void
+refreshAll() : void
RecallPrecisionGraph
-dirty : boolean = false
#recallPoints : float[0..*]
#standardRecallPoints : int = 41
-graphs : Map<Object, RecallPrecisionGraph>
#refresh() : void
+getPointArray(ein x : int, ein y : int, ein w : int, ein h : int) : int[0..*]
+get(ein hitSet : HitSet) : RecallPrecisionGraph
+get(ein run : Run) : RecallPrecisionGraph
+markDirty() : void
+refreshAll() : void
HitSetRecallPrecisionGraph
-hitSet : HitSet
#refresh() : void
+HitSetRecallPrecisionGraph(ein hitSet : HitSet)
+getQid() : String
+getRunId() : String
RunRecallPrecisionGraph
-run : Run
#refresh() : void
+getRunId() : String
+RunRecallPrecisionGraph(ein run : Run)