Information Retrieval

von Markus Schütze

Matr. 338084

a. Inhaltsverzeichnis

Kapitel

1. Ein kurzer Überblick über Information Retrieval und Definition der Begriffe 'Information' und 'Wissen'.

In den letzten Jahren hat, bedingt durch die rasante technische Entwicklung, die Menge der für den Menschen verfügbaren Informationen zugenommen.
Uns stehen heutzutage quasi unerschöpfliche Informationsbibliotheken zur Verfügung.
Durch die zunehmende Vernetzung (z.B. durch das Internet) ist die Beschaffung von Informationen, deren Katalogisation, Verwaltung und Speicherung ein Problem geworden, das in immer größerem Ausmaß das normale Alltagsleben des Menschen bestimmt.
Informationen sind die Grundlage, um bei der täglichen Problemlösungsfindung, eine optimale Entscheidung zu treffen.
Darum ist es wichtig für die Individuen der heutigen Informationsgesellschaft, diese Informationsflut beherrschbar zu machen.
Ein großes Problem hierbei besteht darin für einen Einzelnen zu entscheiden, welche Arten von Informationen wichtig sind und welche nicht. Da die Menge der täglich auf uns einstürzenden Informationen so groß geworden ist, ist es hilfreich, Hilfsmittel zu erschaffen, die uns bei der Informationsbeschaffung und deren Verwaltung unterstützen, ohne uns einzuschränken.
Dieses Hilfsmittel sind die heutigen Information Retrieval Systeme.
Information Retrieval beschäftigt sich mit der Repräsentation, der Speicherung, der Organisation und dem Zugriff auf Informationen.
Was ist aber genau Information, und wie grenzt sich diese von den Begriffen 'Wissen' und Daten ab ?
Unter Information versteht man 'die Teilmenge von Wissen die von einer bestimmten Person oder Gruppe in einer konkreten Situation zur Lösung von Problemen benötigt wird..' [Kulen 1989]
Da Informationen häufig nicht vorhanden ist , sucht der Mensch danach in externen Quellen, um sein Verlangen zu befriedigen. Das Information-Retrieval-System hilft ihm, aus dem gespeicherten Wissen die benötigte Information zu extrahieren.
Ein wesentlicher Unterschied zu Wissen und Informationen besteht darin, das Informationen flüchtig sind, Wissen dagegen permanent zur Verfügung steht. Schlagwortartig lässt sich die Beziehung zwischen den Begriffen 'Wissen' und 'Information' durch die Formulierung ausdrücken : 'Information ist Wissen in Aktion'.

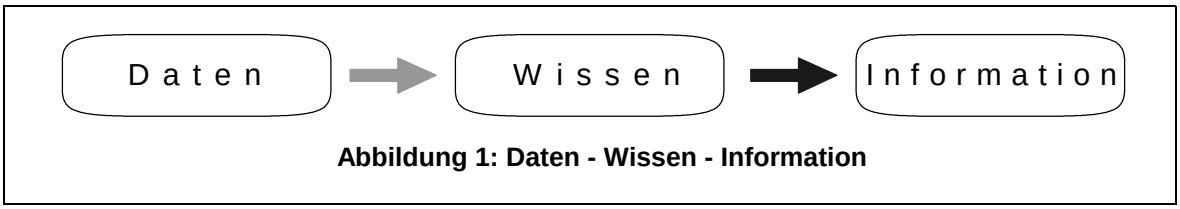


Abbildung 1: Daten - Wissen - Information

Die Hauptaufgabe eines Information Retrieval System besteht also darin, Informationen aus der Gesamtheit des z.B. im Internet angebotenen Wissens zu gewinnen, die für den Benutzer nützlich oder relevant sind. Diese sollen ihm bei seiner Problemlösungfindung optimal unterstützen, und somit sein Informationsbedarf stillen.
Hierbei ist es Aufgabe des Information Retrieval Systems wichtige Informationen bzw. Dokumente von Unwichtigen abzugrenzen und diese in eine Rangfolge zu bringen.
Hierbei sei noch mal darauf hingewiesen, dass Informationen 'gewonnen' werden sollen und nicht nur einfache Daten.

1.1 Wodurch unterscheidet sich Information Retrieval gegenüber Data Retrieval

Data Retrival unterscheidet sich vom Information Retrieval in folgenden Punkten: Zum einen arbeitet das Data Retrieval auf einer abgeschlossenen Menge von Einträgen z.B. einer Datenbank. Das Information Retrieval arbeitet auf einer sehr viel größeren eventuell wachsenden und untergeordneten Menge von Eingabedokumenten, wie z. B. dem World-Wide-Web (WWW). Weiter versucht das Data Retrieval als Ergebnis einer (Datenbank-) Suchanfrage einen exakt passenden Eintrag (engl. „exact match") zu finden. Das Information Retrieval beschränkt sich dagegen bei der Fülle der zu untersuchenden Dokumenten auf eine Menge „partiell zutreffender" Suchantworten, von denen wiederum die geeignetsten Antworten zur weiteren Untersuchung ausgewählt werden. Die Informationssuche ist beim Information Retrieval probabilistisch, beim Data Retrieval deterministisch. Die deterministische Suchtechnik beruht auf einer deduktiven Logik d.h. auf Regeln der Form aRb und bRc, aus denen sich die Regel aRc ableiten lässt, wobei R eine beliebige Relation definiert. Wird die deterministische Suche weiter strukturiert, so lässt sich ein sogenannter Entscheidungsbaum

generieren. Beim probabilistischen Information Retrieval sind alle oben genannten Relationen mit einer gewissen „Unsicherheit" behaftet, wodurch die Zuverlässigkeit der ermittelten Endregel variabel bleibt. Das Ergebnis ist eine Suchergebnismenge die nur eine gewisse Ähnlichkeit mit der Suchanfrage hat und nicht wie beim Data Retrieval einem exaktem Eintrag entspricht. Beide Retrievalarten unterscheiden sich in der Art , wie sie die zugrundeliegenden Objekte klassifizieren. Beim Data Retrieval werden attributbehaftete Objekte eindeutig einer bestimmten Klasse zugeordnet (monothetische Klassifikation). Das Information Retrieval hingegen besitzt Objekte mit Attributen, welche das Objekt verschiedenen Klassen zuordnen können. Überwiegen dabei Attribute, die das Objekt einer bestimmten Klasse zuordnen, so wird das Objekt der gesamten Klasse zugeordnet. Auch unterscheiden sich beide Retrievalarten in der Art ihrer Suchanfragesprachen. Die Suchanfragesprache des Data Retrieval ist künstlich, die des Information Retrievals hingegen natürlich. Eine künstliche Suchanfragesprache versucht das für den Benutzer relevante Objekt vollständig über die Suchanfrage zu spezifizieren. Dazu dienen auch boolsche Operationen, die Wörter der Suchanfragesprache miteinander verknüpfen, wie z.B. die Operatoren 'AND', 'OR' und 'NOT' (oder deren äquivalente Darstellungen). Der Benutzer verknüpft also Wörter, die die zu findenden Objekte charakterisieren, und schließt gleichzeitig solche aus, die auf keinen Fall in der Ergebnismenge vorkommen sollen. Solche Informationssysteme bezeichnet man auch als boolsches Retrieval, welches eher dem Data Retrieval zuzuordnen ist, aber gleichzeitig ein Modell des klassischen Information Retrieval bildet.
Natürlichsprachliche Suchanfragesprachen sind dagegen unvollständig, vage und spiegeln die Tatsache wider, dass sich das Information Retrieval darauf beschränkt, relevante, jedoch nicht exakt zutreffende Suchergebnisobjekte zu finden. Die Suchantwort des Information Retrieval gibt also die Wahrscheinlichkeit der Relevanz jedes der gefundenen Objekte wider. Ein weiteres Merkmal künstlicher Suchanfragesprachen ist, dass sie empfindlicher auf Eingabefehler reagieren.
Abschließend ist nochmals darauf hinzuweisen, dass das hauptsächliche Ziel des Information Retrieval die Ermittlung all derjenigen Objekte ist, die relevant zu der vom Benutzer gegebenen natürlichen Suchanfrage sind.
Dabei ist es aber möglich, dass weniger nichtrelevante Objekte zu ermittelt werden können, was den probabilistischen Charakter des Information Retrievals widerspiegelt.

	Data Retrieval	Information Retrieval
Matching	exact	partiell, best match
Inferenz	Deduktion	Induktion
Modell	deterministisch	probabilistisch
Klassifikation	monothetisch	polithetisch
Anfragesprache	formal	natürlich
Fragesezifikation	vollständig	unvollständig
Gesuchte Objekte	Die Fragespezifikation erfüllende	relevante
Reaktion auf Datenfehler	sensitiv	insensitiv

Tabelle 1.1.1

1.2 Der Retrieval Prozess am abstraktem Beispiel

Der effektive Retrieval Prozess , d.h. das Finden von relevanter Information, wird von zwei Sichten beeinflusst. Der eine maßgebliche Faktor ist der sogenannte 'User Task**'.** Er deckt im maßgeblichen die Kommunikation zwischen Benutzer und Information Retrieval System ab.
Er wird unterteilt in den 'Retrival Task' und in den 'Browsing Task'. Im 'Retrieval Task' spezifiziert der Benutzer seine Suchanfrage mit Hilfe einer Suchanfragesprache (z.B einem regulären Ausdruck) Diese Suchanfrage ist der maßgeblich bestimmende Faktor (engl. Constraint) bei der Ermittlung eines Suchanfragergebnisses.
Browsing stellt mittels Navigationshilfen (z.B. Knowledge-Maps oder Verzeichnissen) ein Hilfsmittel dar, mit dem sich der Benutzer bei einer eher unscharfen Vorstellung seiner Wissensziele einen Überblick über das vorhandene Wissen der Wissensbasis verschaffen kann. Der Suchprozess ist interaktiv, da hier durch das vorliegende Wissen navigiert keine Suche initiiert wird. In heutigen kommerziellen und nichtkommerziellen Retrievalsystemen kann der Benutzer meist zwischen 'Retrieval Task' und 'Browsing Task' umschalten.(z.B. Suchmaschienen des WWW, Google etc.)

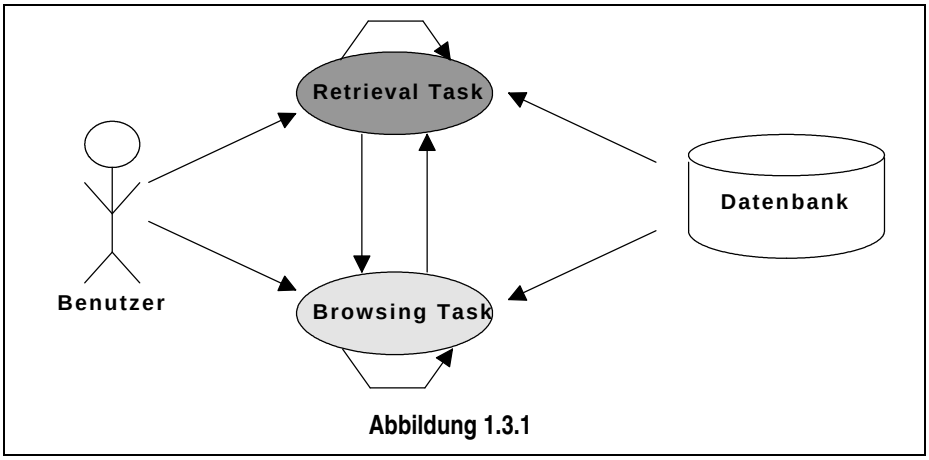
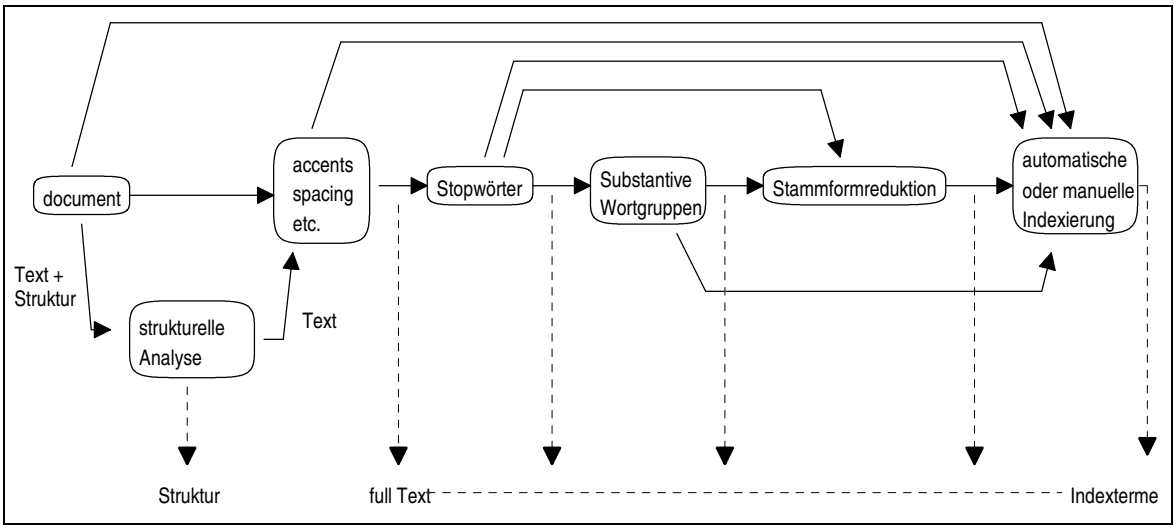


Abbildung 1.3.1

Die andere Sicht des eigentlichen Retrieval-Prozesses ist die logische Sicht auf die Dokumente bzw. Objekte, auf denen die Suche des Benutzers abgebildet wird.
Die logische Sicht ist wichtig, da auf ihr die Retrievaloprationen und Algorithmen arbeiten, die dafür zuständig sind, relevante von nicht relevanter Information zu trennen und Objekte bzw. Dokumente der Ergebnismenge in eine Rangfolge ihrer Relevanz anzuordnen (engl. ranking).
Die Dokumente werden häufig durch Schlüsselwörter oder Indexterme repräsentiert. Solche Schlüsselwörter oder Indizes werden entweder direkt aus dem Dokument entnommen oder durch einen Menschen spezifiziert.
Wie werden aber nun Schlüsselwörter bzw. Indexterme gefunden, die die Dokumente bzw. die Objekte möglichst genau beschreiben?
Ein einfacher Ansatz besteht darin Dokumente oder Objekte in ihrer Gesamtheit zur repräsentieren. Bei textuellen Dokumenten spricht man dann auch von einer 'Full-Text-Representation'. Überschreiten die Suchobjekte eine gewisse Größe, so wird es schwierig diese für das Information Retrieval akzeptabel handhabbar zu halten.
Die Lösung dieses Problems besteht darin, die Menge der Schlüsselwörter zu verringern.
Dieses geschieht z.B. bei einem Textdokument dadurch, indem man eine Strukturanalyse durchführt, danach sogenannte Stopwörter (Artikel oder Bindewörter) eliminiert , Stammformreduktion durchführt, und/oder Substantive identifiziert.
Ein solches Vorgehen lässt sich auch auf andere Suchobjekte übertragen.
Eine so geartete Informationsreduktion hat zur Folge, dass unterschiedliche logische Sichten (engl. logical views) von ein und demselben Dokument entstehen können, die ein Information Retrieval System bearbeiten muss bzw. für den Retrievalprozess benutzt werden können. Abbildung 1.3.2 zeigt nochmals anschaulich die Informationsreduktion eines Objektes.
Auf diese Art entstehen sogenannte Wörterbücher (Thesauri), die die zu durchsuchende Objektmenge genau charakterisieren.



Es ist offentsichtlich, dass Indexstructuren (in irgendeiner Form) die Basis eines modernen Information Retrieval System sind. Auch ist es wichtig zwischen den so definierten zwei unterschiedlichen Sichten des Information Retrieval Problems zu unterscheiden. In der Computer zentrierten Sicht besteht die Information Retrieval Aufgabe hauptsächlich aus der Bildung und Ermittlung effektiver Indizes, Abarbeitung der Suchanfragen des Benutzers mit hoher Performanz und Entwicklung von 'Ranking'-Algorithmen, die die 'Qualität' des Suchanfrageergebnis verbessern.
In der auf den Menschen zentrierte Sicht besteht die Aufgabe des Information Retrieval hauptsächlich auf der Analyse des Verhalten des Benutzers, des Verstehens seiner Hauptbedürfnisse und der Bestimmung, wie sich solche Effekte auf die Organisation und die Algorithmen eines Retrieval-Systems auswirken.
Wie verläuft der Information Retrieval Prozess aber nun genau ?
Bevor der eigentliche Prozess beginnen kann, ist es unumgänglich, die (Text-) Datenbank zu definieren. Dieses wird gewöhnlich vom Datenbankadministrator übernommen, der dann folgende Schritte durchführt :

- Spezifizierung der benutzten Dokumente/Objekte
- Spezifikation von erlaubten Operationen, die auf den Dokumenten/Objekten ausgeführt werden und
- Definition des zugrunde liegenden Dokument/Objektmodell (Welche Elemente können im IR-Prozeß benutzt werden ?)

Ist diese logische Sicht der Dokumente/Objekte definiert, wird die Indexerstellung durchgeführt. Ein Index ist eine kritische Datenstruktur, da sie dem Information Retrieval System ermöglicht, effizient auf größere Datenvolumen zuzugreifen und somit ein schnelles Suchen überhaupt ermöglicht. Nach der Indexierung der Dokumentdatenbasis kann nun der eigentliche Retrieval Prozess beginnen. Zuerst spezifiziert der Benutzer eine Suchanfrage. Bei diesem Vorgang kann ihn ein Benutzerinterface hilfreich unterstützen. Diese wird dann vom IR-System unter Zuhilfenahme der gleichen Operationen, die auf den Dokumenten angewendet werden, analysiert. Danach wird die Datenbasis durchsucht und Dokumente ermittelt, die auf die Suchanfrage zutreffen. Bevor die ermittelten Dokumente/Objekte dem Benutzer zur Einsicht angezeigt werden, werden sie bewertet und hinsichtlich ihrer Relevanz sortiert.Die Algorithmen die dieses 'Ranking' durchführen, werden in dieser Abhandlung im Kapitel 2 näher betrachtet.
Nach dem 'Ranking' werden die Dokumente dem Benutzer zur Einsicht angezeigt. Zu diesem Zeitpunkt kann er die Ergebnismenge einsehen und eine Untergruppe spezifizieren, die seine Interessen besser erfüllt. Durch mehrmaliges Selektieren, initiiert der Benutzer einen Feedbackzyklus. In diesem Zyklus benutzt das IR-System die Informationen, die es durch die Selektierung der für den Benutzer relevanten Informationen gewinnt, zur Abänderung der ursprünglichen Suchanfrage. Diese modifizierte Frage ist meistens eine bessere Repräsentation der 'realen' Wünsche bzw. Interessen des Benutzers. Eine grundlegende Übersicht liefert Abbildung 1.3.3

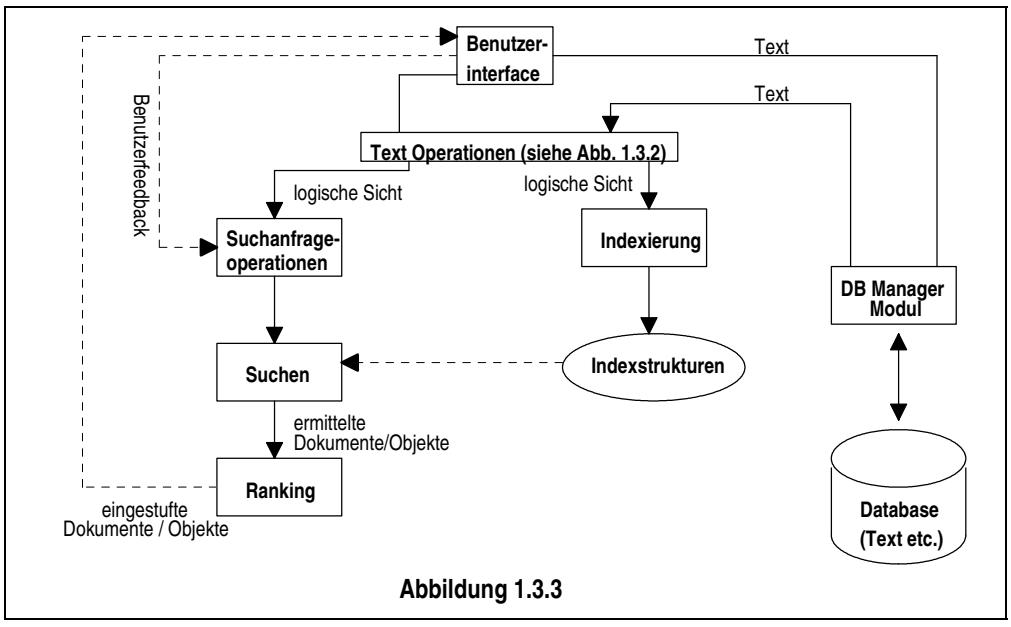


Abbildung 1.3.3

2. Modellierung von Retrieval , Grundlegende Begriffe und Probleme

Wie also im letzten Kapitel erwähnt wurde, ist das Sortieren der gefundenen Dokumente/Objekte nach ihrer Relevanz einer der wesentlichen Bestandteile des Information Retrieval Prozesses. Viele Information Retrieval Systeme erstellen einen Thesaurus (durch Indexierung) und ermitteln Dokumente/Objekte anhand von diesen erstellten Indexstrukturen. Ein Indexterm ist ein Schlüsselwort oder eine Gruppe verwandter Wörter, die eine einzigartige Bedeutung innehaben. Vereinfacht kann man z.B. für Text Retrieval auch sagen, dass ein Indexterm ein Wort ist , das im Inhalt eines Dokumentes in der Suchmenge auftaucht, und dieses zusammen mit anderen Termen eindeutig charakterisiert. Information Retrieval, welches auf Indextermen basiert, ist relativ einfach durchzuführen. Es wirft aber auch Fragen auf, die den zuvor erwähnten 'Retrieval Task' betreffen. Auf der einen Seite basiert Information Retrieval auf Basis von Schlüsselwörtern auf der Annahme, dass man mit einem Satz von Schlüsselwörtern die Semantik hinreichend natürlich abbilden kann. Gleiche Annahmen werden auch über die Darstellbarkeit des Informationsbedarfs des Benutzers gemacht. Auf der anderen Seite gilt dieses aber nur begrenzt, da bei der Konvertierung von Dokumenten und Suchanfrage in Schlüsselwörter natürliche Semantik der ursprünglichen Dokumente verloren geht. Daher entsteht beim Vergleich von einzelnen Dokumenten mit einer vom Benutzer gegebenen Suchanfrage immer eine gewisse Unsicherheit bzw. Ungenauigkeit, die sich im Endergebnis des Retrievalprozeß äußert und dessen probabilistischen Charakter unterstreicht . Es ist folglich keine Überraschung, das ermittelte Dokumente als Ergebnis einer Anfrage, ausgedrückt durch Schlüsselwörter, oft irrelevant sind. Diese Fakten verkomplizieren das Hauptproblem des Information Retrieval, das darin besteht, zu erkennen, welche Dokumente hinsichtlich der Suchanfrage relevant sind und welche irrelevant. Um nun gutes Information Retrieval durchzuführen, ist es notwendig gute 'Ranking'-Algorithmen zu entwerfen, deren Aufgabe es ist, die Menge der ermittelten Dokumente/Objekte zu ordnen, um so diese probabilistischen Erscheinungen des Prozesses zu minimieren. Dokumente die am Anfang dieser Ordnung stehen sollten relevanter gegen der Suchanfrage sein als solche die am Ende der Ordnung stehen.

2.1 Einführung und kurzer Überblick über Retrieval Algorithmen

Den Kern eines Information-Retrieval-Systems bilden die Ranking-Algorithmen
Im Laufe der technischen Entwicklung und als Ergebnis der Forschungen im Gebiet des Information Retrievals haben sich unterschiedliche Gruppen bzw. Verfahren des 'Rankings' gebildet. Eine Übersicht liefert uns folgendes Schaubild.

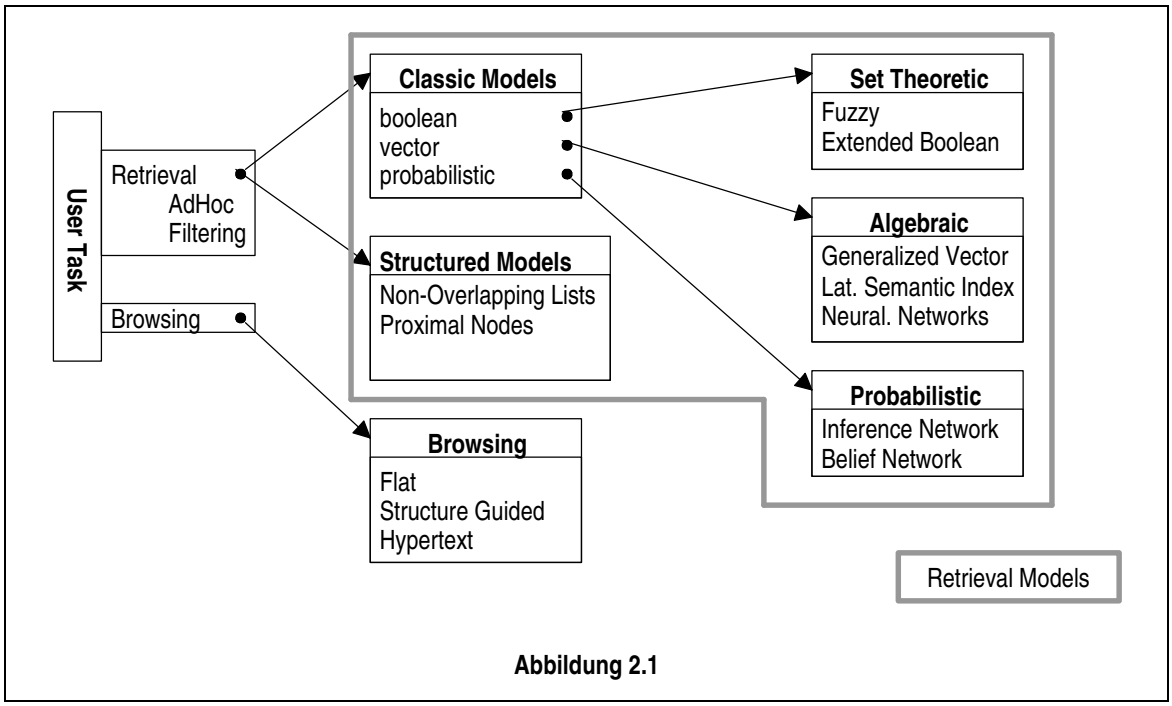


Abbildung 2.1

Wie man sieht gibt es zwei Arten des Retrievals. Beim 'ad hoc'-Retrieval bleibt die Menge der Dokumente in der Suchmenge relativ statisch, während neue Anfragen an das Information-Retrieval-System übermittelt werden. Dieses Modell ist die übliche Form des 'User-Task' des Information Retrievals. Ein ähnlicher, aber

unterschiedlicher Ansatz ist das 'Filtering'. Beim 'Filtering' verbleiben die Suchanfragen relativ statisch. Werden neue Dokumente in das System eingebunden werden, werden sie mit dem Profil verglichen und eingestuft. Je nachdem wie die Einstufung verläuft, werden sie in die Teilmenge der für den Benutzer relevanten oder nichtrelevanten Dokumente/Objekte eingeordnet. Dieser Ansatz wird dann eingesetzt wenn die Menge der zu selektierenden Dokumente im Verhältnis zum Wachstum der Menge der Suchdokumente sehr gering ist. (z.B Newsfiltersysteme....)
Typischerweise ermittelt der Filtertask die Dokumente, die für den Benutzer relevant erscheinen. Die eigentliche Auswahl, welche Dokumente denn nun relevant sind, bleibt aber dem Benutzer überlassen. Auch hier wirken wieder 'Ranking'-Algorithmen auf die Ergebnismenge ein, da dem Benutzer das Ranking der gefilterten Dokumente angezeigt wird.
Beim 'Filtering' ist der wesentliche Schritt nicht das 'Ranking' sondern eher die präzise Erstellung des Profils, das die Interessen des Benutzers vertritt.
Ein simpler Ansatz ist die Definition des Profils durch Schlüsselwörter, die der Benutzer angeben muss. Dieser Ansatz ist zwar einfach, bürdet aber dem Systemnutzer zuviel Arbeit auf. Erstens ist der Benutzer nicht vertraut mit den Vorgängen, die ein solches System vollzieht, wie es beispielsweise Suchdokumente verwaltet. Zweitens liegt das Problem benutzerseitig darin, zu spezifizieren, welche Interessen er eigentlich hat, welche Informationen für ihn relevant sind und wie er gute Schlüsselwörter wählt. Drittens ist es für den Benutzer schwer, die dem System zugrunde liegende Suchanfragesprache zu erlernen. Dieser Prozeß kann sehr zeitraubend sein und sollte daher vermieden werden.
Ein besserer Ansatz definiert sich durch das Sammeln von Informationen über Benutzervorlieben und Interessen. Diese Informationen werden dann eingesetzt um das Benutzerprofil dynamisch zu kreieren. Dieser Vorgang läuft wie folgt ab:
Zuerst stellt der Benutzer einen Satz von Schlüsselwörtern zu Verfügung, mit dessen Hilfe er seine Interessen rudimentär spezifiziert. Bei Eingang neuer Dokumente wird dem Benutzer eine Liste angezeigt, in der er auswählt, welche Dokumente für ihn relevant sind und welche nicht. Im Hintergrund dieses Vorganges passt das Information-Retrieval-System das durch die Schlüsselwörter gegebene Benutzerprofil dynamisch an. Es sollte klar sein, dass bei diesem Ansatz das Benutzerprofil einem ständigen Wandel unterliegt. Dieses sollte sich aber nach einer gewissen Zeit stabilisieren.
'Adhoc-', sowie 'Filtering' sind den den gleichen Retrieval-Ranking-Algorithmen unterworfen.
Die klassischen Modelle, die die Rankingalgorithmen beschreiben, beruhen darauf, dass jedes Dokument durch repräsentative Schlüsselwörter beschrieben wird. Die drei klassischen Modelle sind 'Boolean', 'Vektor' und 'probalistisches Model'. Im 'Boolean-Model' sind Dokumente und Suchanfragen als Sätze von Indextermen repräsentiert. Man bezeichnet dieses Modell auch als 'Theoretic- Set'
Im Vektormodell werden Dokumente und Suchanfragen durch Vektoren in einem n-dimensionalen Raum abgebildet. Dieses Modell wird auch als 'algebraisches Modell' bezeichnet.
Die logische Sicht der Dokumenten und Suchanfragen im probabilistischen Modell basiert auf der probalistischen Theorie. Im Laufe der Forschungen sind die klassischen Modelle verfeinert worden. (siehe Abbildung 2.1). Diese Modelle werden im nachfolgenden Kapiteln dieser Ausarbeitung nur teilweise kurz beschreiben, da Sie den Rahmen dieser Abhandlung sprengen würden.

2.2 Eine Formelle Definition eines IR-Modells

Bevor wir die Basiskonzepte der klassischen 'Ranking'-Algorithmen verstehen können, müssen wir uns erst verdeutlichen, welches Grundmodell ihnen zugrunde liegt.
Die nötige Definition lautet wie folgt:

Definition 1 : *Ein Information Retrieval Modell ist ein 4er-Tupel* $[D,Q,F, R(q_i, d_j)]$ *mit*

(1) $\boldsymbol{D}$ *ist ein Set aus zusammengesetzten logischen Sichten der Dokumente der Suchmenge*
(2) $\boldsymbol{Q}$ *ist die Suchanfragemenge, die aus einem Satz zusammengesetzter logischer Sichten/Representationen besteht.*
(3) F *ist ein Schema, das beschreibt wie Dokumente und wie Anfragen auf diese modelliert sind*
(4) $\boldsymbol{R(q_i,d_j)}$ *ist eine Rankingfunktion, die eine reele zahl zwischen einer Suchanfrage* $q_i \in Q$ *und einem Dokument* $d_j \in D$ *abbildet.*
Do eine Rankingfunktion definiert die Reihenfolge, die Relevanz zwischen Dokumenten, die in Beziehung stehen mit einer Frage q_i

2.3 Die Basismodelle

Nun fehlt uns nur noch der Begriff der Gewichtung eines Indexterms, um alle Voraussetzungen zu haben, die nötig sind, um sich mit der Materie von Gewichtungsalgorithmen erfolgreich auseinandersetzen zu können.
Wie bereits zuvor beschrieben wurde, sind Indexterme hauptsächlich Substantive (im textbasierten Retrieval), die die Hauptaussagen eines Dokumentes eindeutig beschreiben.
Da nicht alle Schlüsselwörter das indexierte Dokument gleich gut beschreiben, ist es notwendig eine Gewichtung der Indexterme zu erwägen. So hat ein Schlüsselwort, das ein Dokument nützlich beschreibt eine höhere Gewichtung als ein Schlüsselwort, das das Dokument weniger nützlich beschreibt.
Eine effiziente Beschreibung von Schlüsselwortgewichtung liefert uns folgende Definition :

Sei k_i ein Schlüsselwort (index term), d_j ein Dokument und $w_{i,j} \geq 0$
Dann isteine Gewichtung asoziiert mit dem Paar $(d_j, w_{i,j})$:

Sei t die Anzahl von Schlüsselwörtern innerhalb des IR_Systems und k_i ein generisches Schlüsselwort. $K = \{k_1,..., k_t\}$ der gesamte Satz der vergebenen Schlüsselwörter. Eine Gewichtung $w_{i,j} > 0$ wird für jedes Schlüsselwort k_i, das im Document d_j erscheint vergeben. Für jedes Schlüsselwort k_i , das nicht im Dokument d_j auftaucht ist $w_{i,j} = 0$. Folglich ist jedes Dokument d_j mit einem Gewichtungsvektor $\vec{d}_{i,j} = (w_{1,j}, w_{2,j},...,w_{t,j})$ assoziiert.
Dann liefert uns die Funktion g_i das Gewicht zurück, das mit dem Schlüsselwort k_i im t-dimensionalen Vektor $\vec{d}_{i,j}$ vorkommt. Beispiel $g_i(\vec{d}_{i,j}) = w_{i,j}$

Zusätzlich gilt, dass Gewichtungen von Schlüsselwörtern oder Indextermen unabhängig sind.
D.h. , falls wir die Gewichtung $w_{i,j}$ vom Paar (k_i, d_i) kennen, können wir keine Rückschlüsse auf die Gewichtung $w_{i+1,j}$ vom Paar (k_{i+1}, d_i) ziehen.
Mit diesen Definitionen haben wir nun das nötige Rüstzeug, um uns die Ranking-Algorithmen näher zu betrachten.

2.3.1 Boolsches Modell

Das boolsche Modell ist das einfachste Modell des Information Retrieval. Es basiert auf der boolschen Algebra. Die Suchanfragen, die mit Hilfe von boolschen Ausdrücken spezifiziert werden haben eine präzise Semantik. Das boolsche Modell erfreut sich einer großen Beliebtheit, da es häufig in den ersten bibliographischen Systemen verwendet wurde und auch heute noch verwendet wird. Diese Einfachheit und der Formalismus ist zugleich aber ein großer Nachteil. Ersten basiert dieses Retrievalstrategie auf einem binären Entscheidungskriterium, das keinerlei Möglichkeit für feinere 'Bewertungsgranulate' zulässt.
Zweitens haben boolsche Suchanfragen (engl. Queries) eine allzu präzise Semantik, die es dem Benutzer nicht einfach machen, eine natürlichsprachliche Suchanfrage in einen boolschen Ausdruck zu überzuführen. Dies äußert sich darin, dass viele Benutzer ihre Suchanfragen sehr einfach gestellten und somit das Retrieval irrelevante Dokumente hervorbringt.
Nachdem wir uns Vorteile und Nachteile verdeutlicht haben, werden wir nun den Retrievalalgorithmus genauer betrachten.
Kurz gesagt, ermittelt des boolsche Retrieval Einschätzungen von Information durch Analyse, ob Schlüsselwörter der Suchanfrage (spezifiziert durch den Benutzer) in der Liste der Schlüsselwörter eines Dokumentes vorkommen oder nicht. Für einzelne Werte des Gewichtungsvektors $\vec{d}_{i,j}$ eines Dokumentes sind folglich nur binäre Zustände zugelassen. Also ist die Gewichtung eines einzelnen Schlüsselworts ki wi,j = {0,1}
Eine Suchanfrage q besteht aus Schlüsselworten, die beliebig mit den Bindewörtern : not, and, or zusammengesetzt werden können. Da q somit ein regulärer Ausdruck ist, können wir ihn auch repräsentieren als Disjunktion von Konjunktionen auch bekannt als Disjunktive Normalform (DNF).
Folgendes Beispiel veranschaulicht uns diesen Sachverhalt.

Gesucht sind alle Dokumente die folgendes Kriterium erfüllen :

$$k_a \wedge (k_b \vee \neg k_c)$$

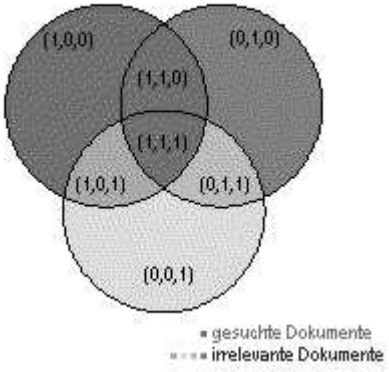
Die DNF zu dieser Anfrage lautet:

$$(1,1,1) \vee (1,1,0) \vee (1,0,0)$$

Wobei ein Element der DNF die Schlüsselwörter wie folgt repräsentiert:

(k_a,k_b,k_c)

Graphisch veranschaulicht



Man sieht hier sehr schön, dass alle die Dokumente , die für den Benutzer als relevant eingeschätzt werden, die gleiche Gewichtung wie ein Element der DNF der Suchanfrage.
Die formale Definition des Ranking Algorithmus des boolschen Modells lautet folglich :

Für das boolsche Model, sind alle Schlüsselwort bzw. Indextermgewichtungen binär, $w_{i,j} \in \{0,1\}$. *Eine Suchanfrage ist ein konventioneller boolscher Ausdruck. Sei* q_{DNF} *die Disjunktive Normal Form der Suchanfrage q, Dann sei* q_{cc} *eine der konjunktitiven Componenten von* q_{DNF}. *Die* ***Ähnlichkeit sim*** *eines Dokumentes* d_j *zu der Suchanfrage q ist definiert als :*

$$sim(d_j, q) = \left\{ \begin{array}{l} 1 \text{ falls } \exists q_{cc} | (q_{cc} \in q_{DNF}) \wedge (\forall k_i, g_i(d_j) = g_i(q_{cc}) \\ 0 \text{ andernfalls} \end{array} \right\}$$

Falls $sim(d_j,q)=1$ *dann nimmt das boolsche Modell an, dass das Dokument* d_j *relevant zur Suchanfrage q ist (Vorsicht es könnte in Realität aber nicht so sein). Andernfalls nimmt es an, dass das Dokument nicht relevant ist,*

Hier kann man auch den Nachteil das boolschen Modell erkennen. Es macht nur Unterscheidungen zwischen relevanter Information und irrelevanter. Es gibt keine Konzeption eines partikulären Suchtreffers betreffend einer gegebenen Suchanfrage q.

Beispielsweise sei d_j ein Dokument mit der Gewichtung $\vec{d_j}$ = (0,1,0). Man sieht, dass d_j das Schlüsselwort k_b enthält. Das Dokument könnte folglich relevant für den Benutzer sein, aber nach obiger Definition ist es nicht relevant zu unserer Beispielsuchanfrage.
Also nochmal zusammengefasst:
Die Vorteile des boolschen Modells sind der klare Formalismus, der hinter dem Model steht, und die Einfachheit des Modells. Der Nachteil dieses Modells besteht darin, dass zu einer Suchanfrage entweder zu wenige oder zu viel Treffer ermittelt werden.

2.3.2 Vektormodell

Das Vektormodell beachtet, dass die Benutzung von binären Gewichtungen der Schlüsselwörter zu begrenzend ist. Es berücksichtigt ein 'Framework' in welchem partielles Matching möglich ist. Dieses ist möglich durch Zuweisung von nicht binären Gewichtungen von Schlüsselwörtern der Suchanfrage , sowie der von Dokumenten. Der prinzipielle Ansatz des Vektormodells ist die Berechnung des Grades der Ähnlichkeit, der zwischen jedem im System gespeichertem Dokument und der Suchanfrage des Benutzers herrscht.
Die Dokumente werden in aufsteigender Reihenfolge sortiert und dem Benutzer präsentiert. Dokumente mit einer geringeren berechneten gradmäßigen Abweichung sind relevanter als solche mit einer höheren Abweichung. Damit berücksichtigt das Vektormodell Dokumente, die partiell auf eine Suchanfrage passen. Das Resultat ist, dass das Ergebnis des Retrievalprozesses präziser ist, als jenes, dass das boolsche Model dem Benutzer präsentiert.
Die Gewichtung der Suchanfragen und Dokumente definiert sich wie folgt :

Für das Vektormodell ist die Gewichtung $w_{i,j}$ assoziiert mit dem Paar (k_i,d_j) eine positive, nicht binäre Zahl. Die Gewichtung der Schlüsselwörter (engl. index terms) in der Suchanfrage erfolgt auf die gleiche Weise. Sei nun $w_{i,q}$ die Gewichtung assoziiert mit dem Paar (k_i,q), mit $w_{i,q} \geq 0$, dann ist der Suchanfragevektor $\vec{q}$ definiert als

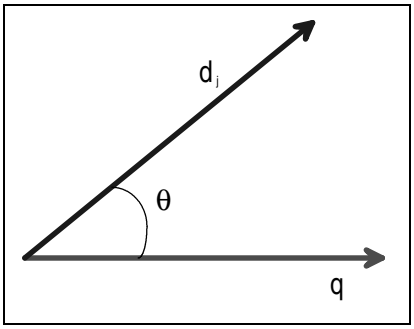
$$\vec{q} = (w_{1,q}, w_{2,q}, ..., w_{t,q})$$

mit t als die Anzahl aller Schlüsselwörter, die im IR-System vergeben wurden.

Der Gewichtungsvektor $\vec{d_j}$ ist repräsentiert als

$$\vec{d_j} = (w_{1,j}, w_{2,j}, ..., w_{t,j})$$

Also sind die Dokumente und die Suchanfrage repräsentiert als t-dimensionale Vektoren.



Wie man sehen kann, ist es sinnvoll, denn Grad θ zwischen Suchanfrage und jedem einzelnen Dokument als Kosinus zwischen dem Dokument- und dem Suchanfragevektor zu verwirklichen.

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

Um mit oben genannter Formel eine aussagekräftige Einschätzung zu bekommen, muss man einen Weg finden die Dokument- und die Suchanfragevektoren zu normieren.
Hierzu gibt es verschiedene Ansätze und Vorgehensweisen.
Eine Methode die hier näher ausführt wird, ist das sogenannte 'Clustering', das auf das Information Retrieval 'aufgesetzt' wird.

Clustering löst folgendes Problem.
Sei C eine Kollektion von Objekten und eine vage (nicht näher spezifizierte) Beschreibung eines Sets A. Das Ziel simpler Clustering-Algorithmen beruht darauf, die Menge C in zwei Teilmengen aufzuspalten. Die erste Teilmenge beinhaltet Objekte, die dem Set A ähnlich sind, die andere Menge Objekte, die nicht A ähnlich sind. Vage ist hier ein Ausdruck dafür, dass man nicht genau entscheiden kann, welche Objekte aus C A ähnlich sind und welche Objekte nicht. Als Beispiel sei eine Menge A von Autos genannt, die preislich vergleichbar sind mit dem Auto des Typs LX400 der Marke Lexus.. Was aber ist mit 'preislich vergleichbar' gemeint. 'Vergleichbar' bedeutet hier, das die Menge A nicht präzise bzw. Einzigartig definiert werden kann. Clustering-Algorithmen beruhen nicht nur darauf, eine Menge in zwei Menge aufzuteilen. Sie können auch mehrere Submengen bilden. Dies ist aber für das Information-Retrieval-Problem unwichtig.
Für uns bedeutet das, dass wir nur den einfacheren 'Clustering'-Algorithmus brauchen.
Die Suchanfrage sei hier die vage Spezifikation des Satzes A von Dokumenten, die durchsucht werden sollen. Im Clustering werden nun zwei Fragestellungen gelöst.
Einmal die Frage, welche 'Features' Objekte des Satzes A beschreiben und zweitens, welche 'Features' die Objekte der Kollektion C die Objekte des Satzes A unterscheiden.
Die erste Lösung beschreibt die Einschätzung 'intra-clusterischer' Ähnlichkeit ('df' = engl. document freqency); die zweite Lösung beschreibt die 'inter-clustersche' Nichtähnlichkeit ('idf').
Im Vektormodell ist die 'inter-clustersche' Ähnlichkeit quantifiziert als die Maßeinheit, die ermittelt, wie oft das Schlüsselwort k_i innerhalb eines Dokumentes d_j vorkommt. Diese nennt man df-Faktor.
Die Maßeinheit, die die die inter-clustersche Unähnlichkeit spezifiziert, wird spezifiziert, als die Häufigkeit des Auftretens von k_i in allen Dokumenten der Kollektion. Diese wird ab sofort idf-Faktor genannt (inverse document frequency).
Die Motivation bei dem idf-Faktor liegt darin, dass Schlüsselworte, die in vielen Dokumenten auftauchen, schlechtere Kandidaten sind, Dokumente eindeutig zu spezifizieren.
Effektive Information-Retrieval-Algorithmen berücksichtigen beide Faktoren.
Diese Algorithmen sind spezifiziert durch :

Sei N die totale Anzahl von Dokumenten im System und n_i sei die Anzahl der Dokumente in welchen das Schlüsselwort k_i erscheint. Sei $freq_{i,j}$ die Häufigkeit des Schlüsselwortes k_i im Dokument d_j. Dann ist die normalisierte Häufigkeit $f_{i,j}$ des Schlüsselwortes k_i im Dokument d_j gegeben durch :

$$f_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}}$$

Das Maximum sei hier über alle Schlüsselwörter berechnet, die im Text des Dokumentes d_j vorkommen. Falls k_j nicht im Dokument d_j vorkommt ist $f_{i,j} = 0$.
Die idf_i (inverse dokument frequency für k_i) sei gegeben durch:

$$idf_i = \log \frac{N}{n_i}$$

Folglich sind die besten Schlüsselwortgewichtungsschemata gegeben durch die Verbindung von df und idf von Dokumenten:

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}$$

Oder durch Varianten dieser Formel.

Für den interessierteren Leser, sei das Buch 'Information Retrieval' von Salton und McGill erwähnt, die grundlegendste Forschungsarbeit auf dem Thema des Information Retrievals geleistet haben.
Für Suchanfragen empfehlen Salton und McGill eine andere Gewichtung :

$$w_{i,q}0(0.5 + \frac{0.5 \cdot freq_{i,q}}{\max_l freq_{l,q}} \times \frac{N}{n_i}$$

Mit $freq_{i,q}$ der Häufigkeit des Schlüsselwortes k_i der Suchanfrage q.

Zusammenfassend:

i. Das Gewichtungschema fördert Performanz des Retrieval Prozesses
ii. Die partielle Trefferstrategie erlaubt partikuläres Information Retrieval
iii. Die kosinusartige Rankingformel sortiert Dokumente entsprechend des Abweichungsgerades zur Suchanfrage

2.3.3 Probabilistisches Modell

Das probabilistische Retrieval Modell nimmt an, dass eine Menge von Dokumenten existiert, die exakt die Benutzerwünsche erfüllt. Diese ideale Ergebnismenge (= engl. ideal answer set) enthält also nur die relevanten Dokumente und keine anderen.
Das Modell fasst den Suchanfrageprozeß als die Spezifikation der Eigenschaften auf, die diese ideale Menge beschreiben. Das Problem hierbei ist, dass man nicht genau entscheiden kann, welche diese Eigenschaften genau sind. Was man bestenfalls machen kann, ist die Ermittlung der Schlüsselworte , die diese Eigenschaften des 'ideal answer sets' beschreiben. Da diese Eigenschaften zum Suchanfragezeitpunkt nicht bekannt sind, muss der Versuch unternommen werden, diese abzuschätzen und zunächst eine erste unpräzise Suchanfrage zu spezifizieren.
Diese Suchanfrage erlaubt es , eine erste probabilistisch ermittelte Beschreibung des 'ideal answer sets' zu generieren, die dazu benutzt wird, eine erste ungenaue Ergebnismenge des Prozesses zu ermitteln.
In Zusammenarbeit mit dem Benutzer wird nun sukzessiv diese anfängliche Beschreibung verbessert, indem ihm das Ergebnis seiner initiierten Suche zur Durchsicht angezeigt wird und er angibt, welche Dokumente für ihn relevant sind und welche Dokumente nicht. Das IR-System benutzt nun diese Information, um die anfängliche generierte Suchanfragebeschreibung zu verbessern. Dieser Prozess wird mehrmals wiederholt, damit die Suchanfragebeschreibung immer mehr an das 'ideale answer set' approximiert wird.
Das probabilistische Model bezieht sich auf folgende fundamentale Annahme :

Sei q eine Suchanfrage und d_j ein Dokument einer Kollektion, dann versucht das probabilistische Modell die Möglichkeit abzuschätzen, ob der Benutzer das Dokument d_j relevant einschätzt oder nicht.
Es nimmt dabei an, dass die Relevanz nur von der Suchanfrage und der Dokumentrepräsentation abhängt.
Es existiert eine Untermenge aller Dokumente, die der Benutzer als Ergebnismenge seiner Suche bevorzugt. Diese Ergebnismenge sei R.

Diese Annahme erklärt noch nicht eindeutig, wie das probabilistische Modell die Relevanz eines Dokuments zu der Suchanfrage ermittelt.

Die folgende Definition liefert dieses nach.

Im probabilistischen Modell seien alle Schlüsselwortgewichtungen binär. D.h. $w_{i,j} = \{0,1\}$, $w_{i,q}$ 0 {0, 1}. Die Suchanfrage q ist eine Teilmenge von Schlüsselworten. Sei R die Teilmenge der Dokumente, die als relevant angenommen werden. Sei $\overline{R}$ das Komplement dieser Menge .
Sei $P(R|\vec{d})$ die Wahrscheinlichkeit, dass das Dokument wirklich zur Suchanfrage relevant ist und $P(\overline{R}|\vec{d})$ die Wahrscheinlichkeit, dass d_j nicht relevant ist zu q.
Die Ähnlichkeit sim(d_j{q) eines Dokumentes d_j zu der Frage q, berechnet sich aus

$$sim(d_j|q) = \frac{P(R|\vec{d_j})}{P(\overline{R}|\vec{d_j})}$$

Durch Anwendung der Formel von Bayes ergibt sich:

$$sim(d_j|q) = \frac{P(\vec{d_j}|R)\times P(R)}{P(\vec{d_j}|\overline{R})\times P(\overline{R})}$$

Zur Erläuterung:

$P(d_j|R)$ steht für die Wahrscheinlichkeit, dass ein zufällig gewähltes Dokument d_j aus der Menge der relevanten Objekte stammt.
$P(R)$ steht für die Wahrscheinlichkeit, dass ein Dokument relevant ist.
Analoges gilt für $P(d_j|\overline{R})$ und $P(\overline{R})$.

Wenn man nun annimmt, dass $P(R)$ und $P(\overline{R})$ gleich sind, kann man obige Formel vereinfachen zu:

$$sim(d_j|q) = \frac{P(d_j|R)}{P(d_j|\overline{R})}$$

Durch Vereinfachung einiger konstanter Werte ist die endgültige Berechnungsformel :

$$sim(d_j|q) \sim \sum_{i=1}^{t} w_{i,q} \times w_{i,q} + (\log \frac{P(k_i|R)}{1-P(k_i|R)} + \log \frac{1-P(k_i|\overline{R})}{P(k_i|\overline{R})})$$

Da wir R am Anfang des Retrieval Prozesses nicht wissen, müssen wir eine Methode finden , die $P(k_i|R)$ und $P(k_i|\overline{R})$ für die oben gebannte Berechnung liefert.
Durch folgende Annahmen :

- Dass $P(k_i|R)$ konstant ist für alle Schlüsselwörter (typischerweise 0,5) und
- Die Annahme, dass die Verteilung der nicht relevanten Schlüsselwörter abgeschätzt werden kann, durch die Division durch die Verteilung aller vergebenen Schlüsselworte in den Dokument

Ergibt sich :

$$P(k_i|R) = 0.5$$
$$P(k_i|\overline{R}) = \frac{n_i}{N}$$

mit n_i = Anzahl der Dokument, die das Schlüsselwort k_i enthalten
Und N die Anzahl aller Dokumente

Diese beiden Formeln geben uns die Möglichkeit ein erstes vages Ergebnis zu einer Suchanfrage zu definieren. Wie beschrieben wird dieses Ergebnis verbessert, indem Korrekturen der Wahrscheinlichkeiten $P(k_i|R)$ und $P(k_i|\overline{R})$ vorgenommen werden.
Dieses funktioniert wie folgt:

Sei V die Teilmenge der zu ermittelnden Dokumente, deren Einschätzung unter einem gewissen Grenzwert r fallen.
Sei V_i die Teilmenge von V, die alle Schlüsselwörter (vom Benutzer spezifiziert) k_i enthalten, dann werden $P(k_i|R)$ und $P(k_i|\overline{R})$ beschrieben durch :

$$P(k_i|R) = \frac{V_i}{V}$$
$$P(k_i|\overline{R}) = \frac{n_i - V_i}{N - V}$$

Mit Hilfe dieser Formeln kann das Neuberechnen des Rankings des IR vorgenommen werden, so dass wir fähig sind die Wahrscheinlichkeiten für die Berechnung der Ähnlichkeit ‘sim’ ohne nochmalige Benutzerinteraktion durchzuführen.
Hingewiesen sei nochmals darauf, dass obige Formeln nur Grundlagen und nicht optimal sind.Sie müssen für spezielle Retrieval Systeme angepasst werden müssen, z.B. durch Korrektur mit konstanten Werten.
Zusammenfassend sind die Vorteile des probabilistischen Information Retrievals

- dass die Dokumente automatisch nach ihrer Relevanz geordnet werden,
- dass das Modell berücksichtigt, dass der Benutzer oft nicht weiß, welche Informationen genau sucht, indem es eine Benutzerinteraktion im Information Retrieval Prozess vorsieht.

Nachteile sind :

- Die Voraussetzung, dass man die Dokumentmenge in relevante und nichtrelevante Teilmengen teilen muss., obwohl noch nicht klar ist, was sie wirklich sind.
- Die Tatsache, dass nicht berücksichtigt wird, wie oft ein Schlüsselwort in den zu durchsuchenden Dokumenten vorkommt und
- Die Annahme, dass Schlüsselwörter gegenseitig unabhängig sind, was sich auf die Anfangseinschätzung der Wahrscheinlichkeit hat und sich auf die repräsentierte Ergebnismenge auswirkt.

3. Retrival Evaluation

Retrieval Evaluation beschäftigt sich mit dem Nachweis der Retrievaleffizienz und der Retrievaleffektivität.
Retrievaleffektivität bezieht sich auf die Fähigkeit des Systems, dem Nutzer die Informationen nachzuweisen, die er sucht.
Retrievaleffizienz ist die Kenngröße, in der Kosten und Zeit gemessen werden, die zur Ausführung bestimmter Systemoperationen erforderlich sind. Letztlich hängt der Wert eines Systems sowohl von der Suchqualität als auch von den Suchkosten ab.
Die vollständige Bewertung eines Retrievalsystems berücksichtigt dementsprechend Effektivität und Effizienz.
Bewertungen von Retrievalsystemen werden aus vielen Gründen durchgeführt. In einem Fall soll ein bereits vorhandenes System mit einem neuem System verglichen werden. Im anderem Fall, möchte man überprüfen, wie sich die Leistung des gesamten Systems ändert, wenn man bestimmte Komponenten modifiziert. Wird beispielsweise das Retrievalverfahren modifiziert oder der Datenbankinhalt verändert, empfiehlt sich eine neue Leistungsbewertung. Ein weiterer Grund für die Systembewertung ist die Implementierung völlig neuer Systemkomponenten in ein bestehendes Retrievalsystem.
Um eine Systembewertung durchzuführen müssen zunächst folgende Vorraussetzungen erfüllt sein :

i. Es muss eine vollständige Beschreibung oder ein Modell des Systems oder der Systemkomponenten, die analysiert werden sollen vorliegen
ii. Es müssen Testhypothesen formuliert werden
iii. Testkriterien und die entsprechenden Messvorschriften müssen festgelegt werden
iv. Die Methoden der Datengewinnung und der Datenbewertung müssen eindeutig spezifiziert werden.

3.1 Bewertung der Retrievaleffektivität - wichtige Kenngrößen

Die Leitung von Retrievalsystemen wird oft mit Recall- und Precisionwerten gemessen, wobei die Recallwerte die Fähigkeit eines Systems zum Ausdruck bringen, verwertbare Dokumente nachzuweisen, während umgekehrt die Precision die Fähigkeit misst , nichtrelevante Dokumente zurückzuweisen. Es gibt noch andere wichtige Kenngrössen u.a. der Aufwand, die Zeit, die Form der Ergebnispräsentation und die Abdeckung der Datenbank, die aus Sicht des Benutzers die Effektivität des Retrievalsystems kennzeichnen. Daraus ergeben sich folgende 6 Kriterien:

i. **Recall**: Die Fähigkeit eines Systems, alle relevanten Dokumente nachzuweisen.
ii. **Precision**: Die Fähigkeit eines Systems, nur relevante Dokumente nachzuweisen
iii. Der intellektuelle oder physische **Aufwand**, der getrieben werden muss, um Suchanfragen zu formulieren, die Informationssuche durchzuführen und das Suchergebnis durchzusehen.
iv. Die **Zeit**, die zwischen der Eingabe einer Suchanfrage und der Präsentation des Suchergebnis vergeht.
v. **Form der Ergebnispräsentation** : Sie beeinflusst die weitere Verwertung der Suchergebnisse.
vi. **Abdeckung der Datenbank:** Diese Kenngröße erfasst das Ausmaß, in dem alle relevanten Dokumente eines Sachgebiets in der Datenbank vorhanden sind.

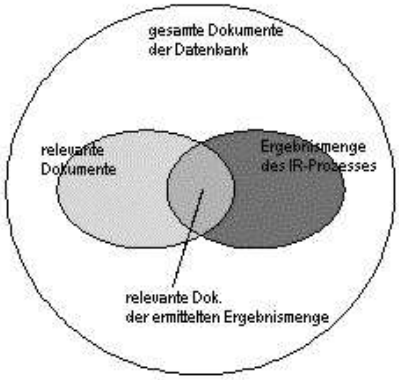
Da Recall und Precision die ermittelte Ergebnismenge aus Sicht des Benutzers bewerten, sind diese beiden Kenngrößen die wichtigsten Maße um Retrievalalgorithmen nach ihrer Effektivität einzuordnen und uns damit eine Bewertungsgrundlage liefern.

Beide Kenngrößen definieren sich wie folgt :

$$Recall = \frac{\text{Zahl der nachgewiesenen relvanten Dokumente}}{\text{Zahl aller relevanten Dokumente in der Datenbank}}$$

$$Precision = \frac{\text{Zahl der nachgewiesenen relevanten Dokumente}}{\text{Zahl aller nachgewiesen Dokumnente}}$$

Oder zu besseren Veranschaulichung sei auf folgende Grafik verwiesen**:**



Für jede Suchanfrage lässt sich demnach ein Recall-Precisionwert berechnen. Vergleicht man für die beiden Suchanfragen i unh j die paarweisen Recall- und Precisionwerte, so wird immer dann, wenn der $RECALL_i \leq RECALL_j$ und die $PRECISION_i \leq PRECISION_j$ ist, die Suchanfrage j besser sein als die Suchanfrage i eingestuft. Problematisch wird es dann, wenn der $RECALL_i < RECALL_j$ und die $PRECISION_i > PRECISION_j$ oder wenn umgekehrt der $RECALL_i > RECALL_j$ und die $PRECISION_i < PRECISION_j$ ist. In diesem Fall muss der Benutzer selbst entscheiden, welche Suchanfrage die bessere ist. In diesem Fall muss letztlich entschieden werden, ob der RECALL oder die PRECISION schwerer zu gewichten ist.
Normalerweise nimmt der RECALL mit der Zahl der nachgewiesenen Dokumente zu, während die PRECISION abnimmt.
Nutzer die an einem hohen Recall interessiert sind, stellen deshalb relativ breit formulierte Suchanfragen, um möglichst viele Dokumente nachgewiesen zu bekommen. Nutzer, die auf eine hohen Precision Wert legen, ziehen hingegen relativ eng formulierte, spezifische Suchanfragen vor.
Dieses veranschaulicht folgendes Auswertung graphisch:

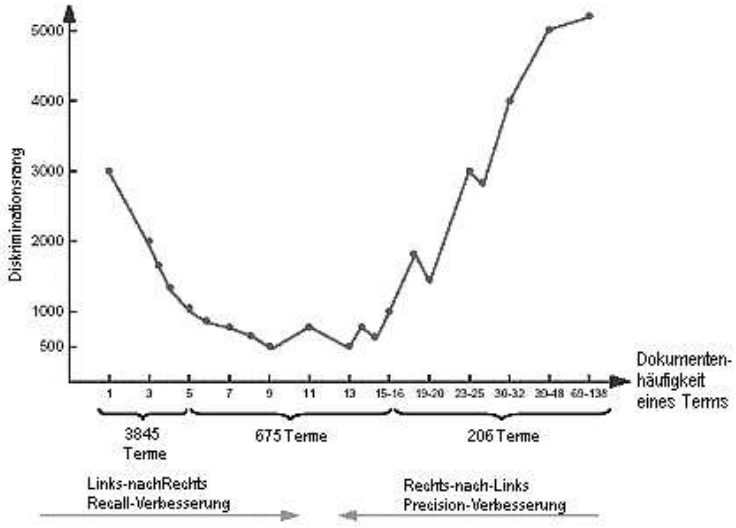


Abbildung 3.1

Zur Berechnung der Recall bzw. der Precision Werte benötigt man also die Gesamtzahl aller relevanten Dokumente bezüglich einer bestimmten Suchanfrage. Ist die Datenbank klein, so stehen meist für alle

Dokumente Relevanzangaben zur Verfügung. Bei großen Datenbanken sind solche ausführlichen Relevanzangaben meist nicht verfügbar. Um dennoch Recallwerte berechnen zu können, muss daher die Zahl der Dokumente geschätzt werden. Diese Schätzung kannn und wird mit Hilfe von Stichprobenverfahren durchgeführt. Das bedeutet allerdings, dass die Relevanzwerte lediglich an einem sehr kleinen Ausschnitt der Datenbank festgemacht werden. Aletrnativ hierzu kann eine bestimmten Suchanfrage mit verschiedenen Suchstrategien oder Nachweisverfahrten durchgeführt werden. Unter der Annahme, dass sich mit einzelnen verschiedenen Strategien alle relevanten Dokumente derartig ermitteln lassen.
Folgendes Beispiel mit 5 relevanten Dokumenten aus einer Suchmenge von 200 Dokumenten, veranschaulicht die Berechnung von Precision und Recall.
(siehe Abbildung 3.2)

n	Dokumentnummer (x = relevant)		Recall	Precision
1	588	x	0.2	1.0
2	589	x	0.4	1.0
3	576		0.4	0.67
4	590	x	0.6	0.75
5	986		0.6	0.60
6	592	x	0.8	0.67
7	984		0.8	0.57
8	988		0.8	0.50
9	578		0.8	0.44
10	985		0.8	0.40
11	103		0.8	0.36
12	591		0.8	0.33
13	772	x	1.0	0.38
14	990		1.0	0.36

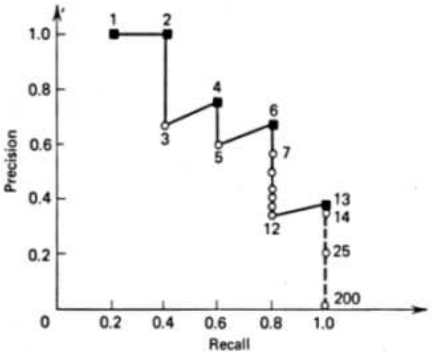


Abbildung 3.2

Wobei die mit X gekenzeichneten Dokumente relevant sind.
Das Problem bei einer solchen Recall-Precisionkurve liegt jedoch darin, dass die Aggregation mehrerer Kurven, um Durchschnittswerte zu ermitteln, nur schwer möglich ist. Andererseits ist die Zahl der nachgewiesen Dokumente und die Größe der Datenmenge nicht erkennbar. Ferner lässt sich keine oder schwer eine funktionale Beziehung ableiten.
Bevor wir nun eine einzige Recall-Precisionkurve konstruieren können, die das durchschnittliche Leistungsvermögen eines IR-Systems auf der Basis einer großen zahl individueller Suchanfragen wiedergibt, versuchen wir nun, obige Kurve zu glätten. Dazu wurden die oben genannten grundlegenden Recall- und Precisionformaln angepasst.
Ein erster Schritt hierzu ist es nur horizontale graphische Verbindungen zuzulassen. Man erhält dann eine Kurve, wie sie in Abbildung 3.3 dargestellt ist. Man erhält sie , indem man mit dem höchsten Recallwert beginnt und eine horizontale Linie zu jedem Schnittpunkt der precision zieht, bis ein höherer Precisionwert erreicht ist.

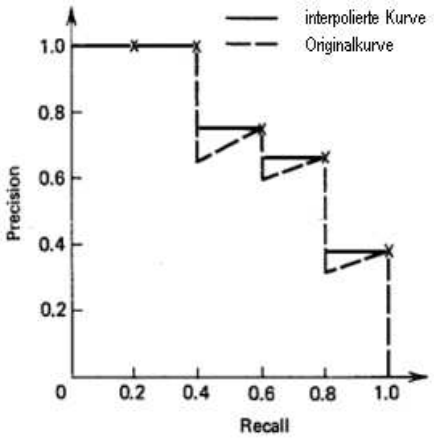


Abbildung 3.3

Es gilt somit

$$Recall_i = \frac{\text{RelDok}_i}{\text{RelDok}_i + \text{NRelDok}_i}$$

$$Precision_i = \frac{\text{RelDok}_i}{\text{RelDok}_i + \text{IRelDok}_i}$$

Mit

RelDok_i = Zahl der nachgewiesenen relevanten Dokumente für eine Query i
NRelDok_i = Zahl der nicht nachgewiesenen relevanten Dokumente für eine Query i
IRelDok_i = Zahl der nachgewiesenen irrelevanten Dokumente für eine Query i

Ein nutzerabhängiger Mittelwert für einzelne Suchanfragen ergibt sich dann durch das arithmetische Mittel über n Suchanfragen:

$$Recall_{RL} = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{\text{RelDok}_i}{\text{RelDok}_i + \text{NRelDok}_i}$$

$$Precision_{RL} = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{\text{RelDok}_i}{\text{RelDok}_i + \text{IRelDok}_i}$$

Dann ergibt sich für unser Beispiel folgende Kurve **:**

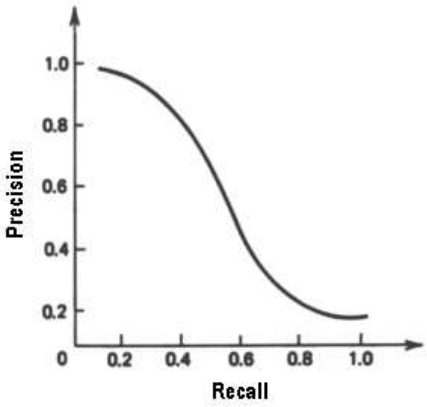


Abbildung 3.4

3.2 Was ist TREC ?

Die Text Retrieval Conference, die abgekürzt TREC genannt wird, ist neben ihre europäischen Pedant CLEF, eine der maßgeblichen Instanzen, die für die Schaffung von großen Retrieval-Test-Kollektionen zuständig ist.
Das Ziel dieser Kollektionen ist eine Basis zu schaffen, auf der unterschiedliche Retrieval Systeme hinsichtlich ihrer Retrievaleffizienz und Retrievaleffektivität verglichen werden können. Lange Zeit existierten nur kleine Retrieval-Testkollektionen, die oft nicht die Hauptmerkmale repräsentierten, wie sie in großen bibliographischen Umgebungen vertreten wurden. Wie wir im Kapitel 3.1 festgestellt hatten, sind Retrievalergebnisse nur dann vergleichbar, wenn eine gewisse Norm der Dokumentenbasis und vergleichbare Suchanfragen existieren, die als Grundlage für den 'Benchmark' von IR-Systemen dienen.
Es war also schwer ermittelte Leistungswerte zu vergleichen bzw. unterschiedliche Aspekte des Retrievalprozesses auf Effektivität und Effizienz zu testen.
In den frühen 90'iger Jahren wurde unter der Führung von Donna Harman am National institute of Standarts and Technology (NIST) in Maryland USA die Text Retrieval Confernce (TREC) ins Leben gerufen. Im Laufe der Jahre vergrößerten sich die Menge der Teilnehmer dieser Konferenz, u. a. bedeutende Größen der Industrie wie z. B. Apple Computer AT&T Labs Research, Daimler Benz Res. Center u. v. A.
Im Laufe der Jahre traf sich dieses Gremium immer wieder, um für die Testkollektion immer neuere Meilensteine zu setzen.
Die TREC-Kollektion ist über die Jahre bis zum heutigen Zeitpunkt stetig gewachsen.
Zum Zeitpunkt der 3. TREC beinhaltete die Kollektion Dokumente und Texte, die zusammen eine Größe von 2 Gigabytes Daten umfassten.
Diese Kollektion wuchs zu einer Größe von bis zu 6 Gigabytes und mehr an.
Sie wird vertrieben auf mehren CD's, die man kostenpflichtig erwerben kann.
Die Inhalte der TREC-Kollektion setzt sich aus folgenden Bereichen unterschiedlicher Kategorien zusammen:

WSJ	Wall Street Journal
AP	Associated Press (news wire)

ZIFF	Coputer Selects (articels) , Ziff-Davis Verlag
FR	Federal Register
DOE	US DOE Publications (abstracts)
SJMN	San Jose Mercury News
PAT	US Patens
FT	Financial Times
CR	Congressional Record
FBIS	Foreign Broadcast Information Service
LAT	LA Times

Die Inhalte diese Sachgebiete sind per SGML als Metadatenstructur gespeichert, so dass sie ein einfaches Parsen erlauben. Hauptstrukturen werden durch die Tags <DOCNO> für die Dokumentnummer und <TEXT> für den Dokumententext gekennzeichnet. Für die jeweils einzelnen Rubriken existieren aber auch noch andere Tags (z.B. <author>, <dateline>, ...), die unterkategoriespezifisch vergeben sind. Diese Tags können unterschiedlich von Subkategorie zu Subkategorie sein.
Ein generelles Beispiel dient folgendes Dokument der Dokumentnummer WSJ880406-0090:

```
<doc>

<docno> WSJ880406-0090 </docno>
<hl> AT&T Unveils Services to Upgrade Phone Networks Under
Global Plan </hl>
<author> Janet Guyon (WSJ Staff) </author>
<dateline> New York </dateline>

<text>
American Telephone & Telegraph Co. introduced the first of a new
generation of phone services with broad ...
</text>

</doc>
```

Abbildung 3.2.1

Die TREC-Collection enthält nicht nur eine Basismenge von zu durchsuchenden Dokumenten, sondern gibt auch eine Menge von Suchanfragen vor.
Diese liegen ebenfalls als SGML-Dokument vor und sollen von dem zu testenden System selbst in eine Suchanfrage (engl. Query) konvertiert werden. In der TREC_Collection werden diese Suchanfragen als Tpics bezeichnet.
Diese Konvertierung ist ein existentieller Bestandteil des TREC-Evaluationprozesses.
Als Beispiel sei folgender Auszug eines Topics angegeben:

```
<top>

<num> Number: 168
<title> Topic: Financing AMTRAK

<desc> Description:
A document will address the role of the Federal Government in
financing the operation of the National Railroad Transportation Cor-
poration (AMTRAK).

<narr> Narrative: A relevant document must provide information on
the government's responsibility to make AMTRAK an economically
viable entity. It could also discuss the privatization of AMTRAK as
an alternative to continuing government subsidies. Documents com-
paring government subsidies given to air and bus transportation with
those provided to AMTRAK would also be relevant.

</top>
```

Abbildung 3.2.1

Ein weiter wichtiger Bestandteil des Evaluationprozesses ist die Angabe und Ausweisung der Menge der relevanten Dokumente zu einer jeden Suchanfrage.
Diese Menge wurde auf den TREC-Konferenzen wie folgt bestimmt.
Sie besteht aus den am besten bewerteten Ergebnisdokumenten der verschiedensten Retrievalprozesse. Diese Submengen sind typischerweise die 100 ersten Dokumente der Ergebnismenge. Diese Konkatenation dieser

Ergebnismengen wird anschließend menschlichen Speziallisten zur Einsicht übergeben, die letztendlich über die Relenvanz jedes der einzelnen Dokumente entscheiden.
Diese Technik wird auch ‘pooling- method’ genannt und oben im Text ansatzweise erwähnt.
Der eigentliche TREC-Benchmark besteht wiederum aus einzelnen Subprozessen.
Diese gliedern sich in zwei hauptsächliche ‘Aufträge’.
Einmal dem ‘adhoc-Task’, indem ein Satz sich ändernder Anfragen gegenüber einer statischen Dokumentmenge gestellt werden, und andererseits dem ‘routing-Task’, indem eine sich ändernde Dokumentmenge gegenüber statischen Anfragen bewertet wird.
Der letztgenannte ‘Task’ ist einem ‘filtering-Task’ gleichzusetzen, bis auf die Tatsache, dass die Ergebnismenge in eine bewertete Reihenfolge (siehe ranking) gebracht wird.
Diese beiden Hauptbewertungen werden durch diverse zusätzlich spezifizierte Benchmarks unterstützt. Sie sind aber einem stetigen Wechsel unterworfen.
Zum Zeitpunkt der 7. TREC-Konferenz wurdem zusätzlich folgende ‘Benchmarktasks’ definiert:

- **Filtering-Task**: Ein routing-Task ohne Bewertung der Dokumente, es wird nur eingeschätzt welche Dokumente relevant sind und welche nicht, partielles Treffer werden nicht ausgewertet
- **Interaktive-Task:** In diesem Vorgang interagiert das zu testende IR-System mit einem Benutzer, um zusammen mit ihm die relevanten Dokumente zu ermitteln.
- **NLP:** Hier wird nachgewiesen , ob Retrievalalgorithmen, die auf natürlichen Sprachprocessing basieren, besser sind als schlüsselwortbasierte Algorithmen.
- **Cross languages Ad hoc-Task :** Wie gut ist Retrieval , dessen Suchanfragen in unterschiedlicher Sprache verfasst sind bei gleich bleibender Dokumentsprache
- **High precision-Task:** Zeitbegrenztes Information Retrieval, bei nicht vorher definierter Suchanfrage
- **Spoken document-Retrival-Task:**
- **Very large corpus-Task**: Ad hoc basiertes Vorgehen, des Information Retrievalprozesses mit Kollektionen von 7.5 Millionen Dokumenten

Diese Benchmarks dienen zur Ermittlung folgender Kenngrößen:

- **Recall-precsion averages:** Wie im Kapitel 3.1 beschrieben
- **Avarage Precison Histogram:** graphische Darstellung der ermittelten größen für jedes einzelne Topic
- **Dokument Level averages:** Bestimmung eine durchschnittlichen Precision auf einer zuvor definierten Stichprobe. Die durchschnittliche Precison wird z.B dann berechnet ,. Wenn nur 5, 10, 15, 20 relevante Dokumente gefunden wurden
- **Summary table statistics:** Statistische Übersicht über Ergebnisse der einzelnen ‘Benchmarktasks’. Diese enthält die Anzahl der Topics, die Anzahl der ermittelten Dokumente über alle Topics, die Anzahl der relevanten Dokumente die effektiv gefunden wurden für alle Topics und die Anzahl der Dokumente, die gefunden hätten müssen über alle Topics.

4. Abschlussbemerkungen

Wie man sieht, ist das Information Retrieval ein sehr großes Gebiet, das uns in immer größeren Maße beeinflussen wird. Es besteht aus vielen Teilbereichen, die für jedes Suchproblem eine optimale Lösung bereitzustellen versuchen.
Information Retrieval versucht sich von Data Retrieval abzugrenzen, indem es Unsicherheiten beachtet, die die Kommunikation zwischen Mensch und Maschiene mit sich bringt. Speziell neben den vektorbasierten Retrievalmodellen versuchen die probabilistischen Modelle diese Phänomen zu erfassen und zu berücksichtigen. Das Gebiet der Retrieval Evolution beschäftigt sich mit der Einschätzung der Effizienz der Information Retrievalsysteme bzw. der in ihnen gekapselten Algorithmen. Ich habe dieses anhand von TREC verdeutlicht. TREC versucht durch eine genormte Datenbasis und genormte Suchanfragen vergleichbare Benchmarks der Retrievalsysteme zu ermöglichen.
Bedingt durch neue technische Entwicklung entwickelt sich auch Information Retrieval immer weiter.
Wir sind also daran gehalten, dieses Gebiet aktiv mit zu gestalten, um künftige Entwicklungen der Zukunft zu beeinflussen, und somit einen positiven Beitrag für die Gesellschaft zu erbringen.

5. Literaturverweise:

- Salton, Gehard; McGill Michael, 1987 , Intruduction to modern information retrieval , MCGraw-Hill New York
- Baeza-Yates, Ricardo, 1999, Modern Information Retrieval, Addison Wesley,
  ISBN: 0-201-39829-X
- LANCASTER, F.W., Information Retrieval Systems: Characteristics, Testing and Evaluation, Wiley, New York (1968)
- Fuhr, Norbert, Information Retrieval, Scriptum zur Vorlesung (englisch)
- RIJSBERGEN van C. J. B.Sc., Ph.D., M.B.C.S., Department of Computing Science University of Glasgow, INFORMATION RETRIEVAL, Scriptum
- Heuser, Udo, Internetsuche und Neuronale Netze: Stand der Technik, 1998
- Willenborg, Josef, Anfragesprachen für Internet-Informationssysteme, 2001