Zusammenfassung IRG

René Bernhardsgrütter, 13.05.2014/16.06.2014
IB=Informationsbedürfnis.

Grundlagen

Information Retireval
Viele un-/strukturierte Informationen organisieren und auffindbar machen.

Pyramide
#3: Daten: Fakten in codierten Form. Tragen per-se keine Bedeutung, können aber korrekt od. falsch sein.
#2: Information: Daten+Bedeutung. Information ist i-/relevant. Benötigt, um Aufgaben zu erledigen, für Aufgabe mehr/weniger relevant.
#1: Wissen: Verarbeitete, vernetzte Information (Erkenntnisse). Z.B. Bestellung abwickeln. Häufig interne und externe Informationen vernetzen.

Retrievalproblem
„Auffinden von möglichst viel relevanter Information bei gleichzeitigem Minimieren von ebenfalls gelieferten irrelevanten Information."

Nicht nur Informationen wieder finden, sondern vor allem neue Information finden. Information wird indirekt geliefert, in Form von „relevanten" Dokumenten.

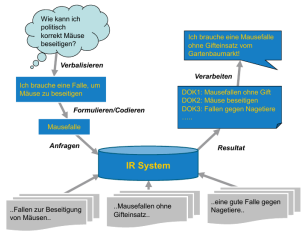
Folge: Ein perfektes Retrievalresultat losgelöst von Benutzer und Kontext gibt es nicht.

Sprachproblem
- Sprache ist nicht „eindeutig": Synonyme (eine Bedeutung – mehrere Wörter), Homonyme (mehrere Bedeutungen – ein Wort) Umschreibungen, Metaphern, Wortformen (Singular, Plural, Fälle, etc.).
- IB ungenügend verbalisiert und formuliert.
- Dok./Infos im System unstrukturiert/ inhomogen.
- Irreführender Inhalt.
- Autorität, Quelle, Aktualität, Urheberrecht, Einsammeln der Dokumente.

Relevanz
Immer subjektiv: Vor-/Hintergrundwissen, Reihenfolge des Auffindens, wandelnde IB, persönliche Präferenzen, Vollständigkeit der Antwort.

Konsequenzen
Unscharfer Relevanzbegriff führt zu wskbasierter Lösung:
- Für User vermutlich relevante Dokumente top gelistet.
- Resultat fast nie vollständig "korrekt": relevante Dokumente fehlen, od. zusätzlich irrelevante gefunden.
- **Scharfe Kriterien (ja/nein) sind ungeeignet**, da User Anz. und Form der gesuchten Dokumente der Anfrage kennen müsste, welche das gewünschte Resultat liefert -> Paradox.
- Gute Retrieval-Systeme erlauben alles zu formulieren was bekannt, ohne zu viel od. zu wenig zu finden.

IR-Prozess
Verbalisierung: „Verstehen des Problems": richtige Begriffe, Vollständigkeit. => Problem muss verstanden werden, um es richtig zu verbalisieren.
Codierung: „Verstehen des Systems": richtige Operatoren, etc.. => Die Resultate müssen bekannt sein, um Anfrage richtig zu codieren.



Das Resultat ist immer nur so gut wie die Anfrage!

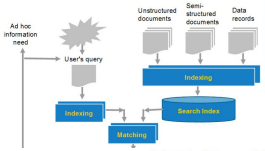
Suchparadoxon
Google kann sich "Vereinfachungen" erlauben dank dem Suchparadox: **Es ist einfacher, in mehreren Milliarden Dokumenten zu suchen als in mehreren Tausend.**

- Grosse Datenmengen => mehr Redundanz.
- Information mit "beliebigen" Verbalisierungen findbar.
- Benutzer von Google sind häufig präzisionsorientiert, wenige gute Treffer reichen.

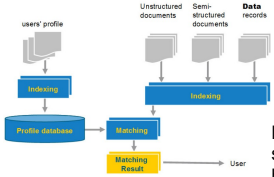
Auf "kleinen" Datenmengen ist Übereinstimmung zwischen Informationsbedürfnis und Dokumenten schwerer nachzuweisen.

Iterative Suche
Verständnis des Benutzers für Informationsbedürfnis ändert sich mit gesammelter Information => iterativ:
- Unstersrtützung bei Umfromulierung: automatische Erweiterung der Anfrage, Suche nach ähnlichen Dokumenten.
- Werden relevante Dokumente gefunden, ändert sich das Verständnis des IBs.
- Benutzer kann beim Verständnis unterstützt werden, z. B. durch automatische Kontextanalyse.

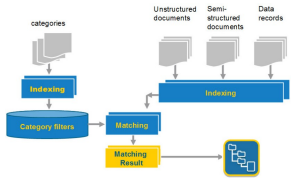
Information Retrieval Paradigmen
Pull
Aufgrund eines Ad hoc-IBs Dokumente mit relevanten Informationen suchen.
Bsp: Ad-hoc-Suchen wie von Bundesgericht.



Push
Dokumente mit relevanten Informationen aus einem Dokumentenstrom herausfiltern und weiterleiten.
Bsp: RSS Reader (*Bringdienste*)

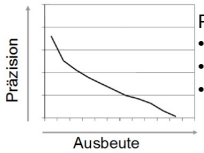


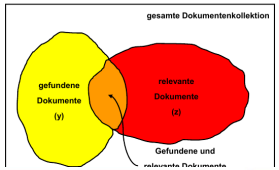
Browse
Neue Dokumente kategorisieren und einordnen.
Bsp: dir.yahoo.com



IR vs. DB-Suche
- Datenbanken liefern für strukturierte Information mit kontrolliertem Vokabular ***perfekte*** Resultate.
- Daten in Datenbanken sollten unabhängig sein von Applikation. Redundanz wird vermieden.
- Elemente sind entweder Teil der Resultatmenge od. nicht (binäre Unterscheidung).
- Boole'sche Kriterien zur Selektion Geeignet für die Suche in (hoch-)struktierter Information mit kontrolliertem Vokabular.

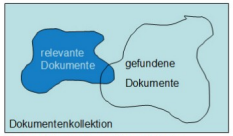
Qualität von Retrievaleffekt
Ausbeute und *Präzision* modellieren Annahme, dass möglichst viel relevante, und möglichst wenig irrelevante Information gefunden werden soll.



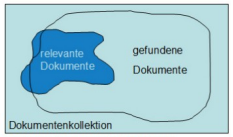


Präzision = x / y

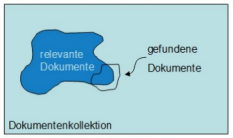
Ausbeute = x / z



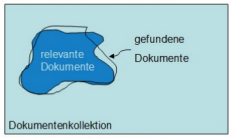
Geringe Ausbeute, geringe Präzision



Hohe Ausbeute, geringe Präzision



Geringe Ausbeute, hohe Präzision



Hohe Ausbeute, hohe Präzision

Ausbeute od. Präzision sollen optimiert werden.

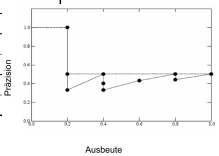
$$\text{Präzision} := \frac{\#\text{relevante Dokumente im Resultat}}{\#\text{Dokumente im Resultat}}$$

$$\text{Ausbeute} := \frac{\#\text{ relevante Dokumente im Resultat}}{\#\text{relevante Dokumente in der Kollektion}}$$

Da beide Masse mengenbasiert sind, widersprechen sie sich oft: hohe Ausbeute -> geringe Präzision, hohe Präzision -> geringe Ausbeute

Beispielauswertung
- Berechnung von Ausbeute und Präzision auf Ranglisten

Rang	Relevant?	Ausbeute	Präzision	Prz. Interpoliert
1	+	0.20	1.00	1.00
2	-	0.20	0.50	0.50
3	-	0.20	0.33	0.50
4	+	0.40	0.50	0.50
5	-	0.40	0.40	0.50
6	-	0.40	0.33	0.50
7	+	0.60	0.43	0.50
8	+	0.80	0.50	0.50
9	-	0.80	0.44	0.50
10	+	1.00	0.50	0.50

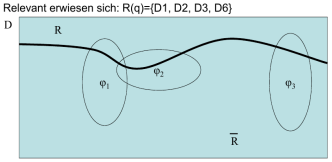


Probability Ranking Principle (PRP)
Resultat sortiert nach Relevanz-WSK. *Optimal* unter Berücksichtigung sämtlicher zur Verfügung stehender Informationen und geeigneter Annahmen. Ist eher Hypothese als Prinzip.

PRP halt folgende math. Eigenschaften:
- Präzision an beliebigem "cut-off point" wird optimiert.
- Ausbeute an beliebigem "cut-off point" wird optimiert.
- Kosten der Auswertung (relevant=positiv, irrelevant=negativ) werden optimiert.

=> WSK-basierte Ranglisten theoretisch fundiert

Problem
q ="Terrorismus, bekämpfen" = ϕ1 ϕ2
D1 = "Gegenmassnahmen gegen Terrorismus" = ϕ3 ϕ1
D2 = "Kampf gegen den Terror" = ϕ2 ϕ1
D3 ="Sicherheit bei asymmetrischer Bedrohung und asymmetrische Sicherheit" = ϕ5 ϕ6 ϕ7 ϕ6 ϕ5
D4 ="Terror bekämpfen" = ϕ1 ϕ2
D5 ="Extremismus und Gewalt" = ϕ8 ϕ9
D6 ="Terrorismus und innere Sicherheit" = ϕ1ϕ10 ϕ5
Relevant erwiesen sich: R(q)={D1, D2, D3, D6}



Indexierung und Vergleichen
Term/Merkmal: eindeutiges Wort
Token: Auftreten eines Terms
Wort: Einheit zwischen Trennzeichen bei Tokenisierung.
Merkmalshäufigkeit: Anz. Vorkommen von Merkmals in Dokument.
Dokumentenhäufigkeit: Anz. Dokumente, die Begriff enthalten.

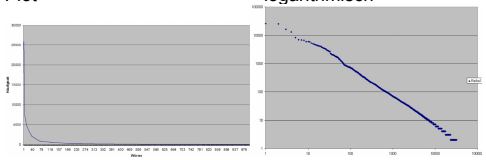
Inhaltstragende Wörter
Manche Wörter tragen wenig *lexikalischen* Inhalt. Z.B. Artikel, Partikel, Pronomen, Konjunktionen, ..

Aber: Bedeutung von "nicht" ist wichitg! Für Retrieval interessant sind Wörter, welche inhaltstragend sind und Text "auszeichnen".

Zipfsches Gesetz

Rang der Häufigkeit * Häufigkeit = *constant*

Plot logarithmisch



Häufigste Wörter: der, die, und, ...

Indexierung

Automatische Verschlagwortung ist nicht trivial.

Vollindexierung

Alle Wörter analysiert, nicht alle gespeichert. Reihenf.:

1. Buchstabenumwandlung
2. Wortextraktion
3. Stoppwortelimination

Tokenisierung

Extrahiert einzelne Wörter aus Text:

1. Dokumentenformate konvertieren.
2. Zeichencodierung anpassen (ISO-8859-1, UTF-8, ..).
3. Gross-/Kleinschreibung normalisieren.
4. Text entlang Trennzeichen zu Tokens separiert.

Stoppwortelemination

Nicht inhaltstragende Wörter ignorieren. Spart 40% Platz.

Stemming

Entfernt Dinge wie Kasus, Numerus, Genus etc.. Dokumente auffinden unabhängig von einzelnen Wortformen. Sollten aber relevant sein, d.h., gesuchte Sachverhalt sollte "abgebildet" werden. Wörter können linguistisch verwandt sein, aber für Suche wäre Treffer unnütz: Bildung <-> Bild

Englisch: Porter-Stemmer, Wirkung umstritten.
Deutsch: verschiedene Ansätze, bis zu +30% Effektivität.

Komposita

Sollten zerlegt werden (+30 % Effektivität), aber nicht alle (Wettbewerb, Frühstück). Wie trennen? Fussballweltmeisterschaft zu Fussball|Weltmeisterschaft od. Fuss|Ball|Welt|Meisterschaft?

Wenige Systeme bieten eine solche Zerlegung!

Bag of words

Dokument wird als Menge von Wörtern betrachtet.

Indexierung unterschiedlicher Granularität/n-Grams

Im Falle, dass einzelne Terme als Indexierungselemente ungeeignet sind (Texte mit grammatikalischen od. typografischen Fehlern, unbekannte Sprache des Textes, etc.) müssen kleinere Elemente für die Indexierung herangezogen werden.

Information
_In
Inf
nfo
for
orm
rma
mat
ati
tio
ion
on_

Rangierungsprinzipien

1. Je mehr Suchbegriffe in Dokument desto, wsk ist Dokument relevant.
2. Je häufiger ein Suchbegriff in Dokument, desto wsk ist das Dokument relevant.
3. Dokumente, die seltene Suchbegriffe enthalten, sind mit einer höheren Wsk relevant als Dokumente, die häufige Suchbegriffe enthalten.
4. Je mehr Hyperlinks auf Dokument zeigen desto wsk ist es relevant.
5. Je näher die Suchbegriffe beieinander liegen desto relevanter ist Dokument.
6. Je früher Suchbegriffe in Dokument vorkommen desto höher dessen Relevanz.

Wortstatistiken

feature frequency (ff) / term frequency (tf): Merkmalshäufigkeit bzw. Termhäufigkeit, also wie oft ein Merkmal/Term in Dokument auftritt.
df (document frequency): Dokumentenhäufigkeit; in wie vielen Dokumenten Merkmal/Term auftritt.
Dokumentlänge: z.B. Anz. Tokens, Anz. Merkmale/Terme, Bytelänge.

Inverse document frequency (idf)

Ermittelt wie viele Dokumente einer Kollektion einen Term beinhalten, relativ zur totalen Anz. von Dokumenten der Kollektion: *Inverse Doc. Frequency* $idf(\varphi_k)$ von Term k.

N: Anz. Dokumente in der Kollektion.
$df(\varphi_k)$: Anz. Dokumente, die Term k enthalten.

$$idf(\varphi_k) = \log \frac{1+N}{1+df(\varphi_k)}$$

Mit $idf(\varphi_k)$ kann Ausprägung (wie "charakteristisch") eines Terms für ein Dokument besser bestimmt werden als durch simples Zählen der Häufigkeiten.

Termgewichtung mit idf

Wichtigkeit/Gewicht eines Terms *k* in Dokument *i* nimmt zu, wenn Häufigkeit *tf(dj)* steigt, nimmt aber ab, wenn Dokumentenhäufigkeit *df* zunimmt.

Dadurch wird die *Gewichtungsfunktion* wie folgt definiert:

$$w(\varphi_k, d_i) = tf(\varphi_k, d_i) * idf(\varphi_k)$$

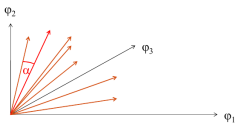
$tf(\varphi_k,d_j)$: Anz. der Vorkommen des Terms *k* in Dokument *i*.

Hohes Gewicht haben dabei Terme, die in wenigen Dokumenten innerhalb der Kollektion häufig auftreten.

Vektorraummodell

Gewichtung einzelner Terme reicht nicht. Anfrage als Ganzes mit Dokument als Ganzes vergleichen.

Dokumente als Vektoren (n-dimensionaler Raum!)
Anfrage als Vektor
Sim(Dokument, Anfrage) = Kosinus(Winkel)



Längennormalisierung

Um Dokumente und Anfrage als Vektoren in n-dim. Vektorraum darzustellen, ist folgende Gleichung gut.

RSV: retrieval status value
α: Winkel zw. den Vektoren q und d.
(q,d): skalare Produkt, kennzeichnet Länge des Vektors.

$$RSV = \frac{(q, d)}{\|q\|\|d\|} = \cos(\alpha)$$

Gewichtungsformel

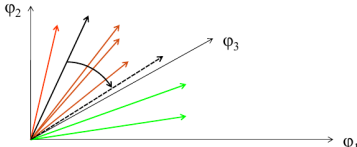
$$a_{i,j} := ff(\varphi_i, d_j) * idf(\varphi_i)$$

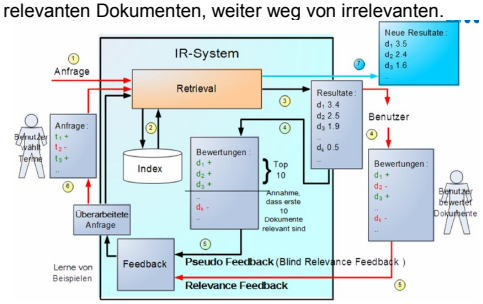
$$b_i := ff(\varphi_i, q) * idf(\varphi_i)$$

$$RSV(q, d_j) := \frac{\sum_{\varphi_i \in \Phi(q) \cap \Phi(d_j)} a_{i,j} * b_i}{\sqrt{\sum_{\varphi_i \in \Phi(d_j)} a_{i,j}^2} * \sqrt{\sum_{\varphi_i \in \Phi(q)} b_i^2}}$$

Relevanzrückkoppelung (Relevance Feedback)

Benutzer identifiziert im Resultat i-/relevante Dokumente. Neuer Vektor wird konstruiert, näher bei relevanten Dokumenten, weiter weg von irrelevanten.





Binary Independence Retrieval (BIR)

Relevanzwsk durch Auftretenswsk von Termen darstellen.

Annahmen/Einschränkungen:

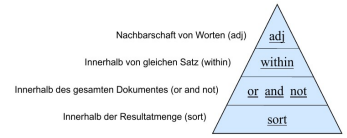
- Dokumente sind "Bag of Words", nur so dargestellt.
- Dok. und Anfrage durch binäre Vektoren repräsentiert.
- Einzelne Terme sind voneinander unabhängig, d.h. P(Vektor x|R,q) = Produkt der einzelnen P(Term xi|R,q).
- Terme, die nicht in Anfrage vorkommen, sind in i-/relevanten Dokumenten gleich häufig.
- Alle nicht vom Dokument abhängigen Faktoren werden ignoriert, Logarithmus vereinfacht Berechnung.
- Sehr wenig relevante, sehr viele irrelevante Dok.

Resultat **RSJ (Robertson-Sparck Jones)** Gewichtung. log(pi(1-qi)/(qi(1-pi))) ergibt ohne Relevanzinformation eine idf-Gewichtung.

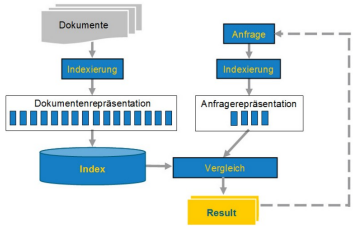
Boolesches Retrieval

Älteste und populärste Retrievaltechnik basierend auf Nicht-/Existenz von Merkmalen, z.B ungewichtete Terme. Boolesche Ausdrücke werden als Anfrage verwendet, Evaluation gm. bool. Logik.

Ergebnis sind zwei disjunkte Mengen, Resultat und Rest, wobei Elemente nicht nach Relevanz geordnet.



Systeme, Architektur



Invertierte Liste

Besteht aus geordneten Liste von Merkmalen mit min. Information über Häufigkeit jedes Merkmals in Kollektion. Enthält Dokumenthäufigkeit, z.B. in wie vielen Dokumenten Merkmal auftritt, in welchem Dokument es auftritt, an welche Stellen im Dokument Merkmal auftritt.

Invertierte Liste

Merkmal #	Merkmal	Merkmal und Lokalisierungsinformation
1	cold	2; (1, 6), (4, 8)
2	days	2; (3, 2), (6, 2)
3	hot	2; (1, 3), (4, 4)
4	in	2; (2, 3), (5, 4)
5	it	2; (4, 3,7), (5, 3)
6	like	2; (4, 2,6), (5, 2)
7	nine	2; (3, 1), (6, 1)
8	old	2; (3, 3), (6, 3)
9	pease	2; (1, 1,4), (2, 1)
10	porridge	2; (1, 2,5), (2, 2)
11	pot	2; (2, 5), (5, 6)
12	some	2; (4, 1,5), (5, 1)
13	the	2; (2, 4), (5, 5)

Lokalisierung innerhalb Dokument
Dokumentnummer
Dokumentenhäufigkeit

Granularität

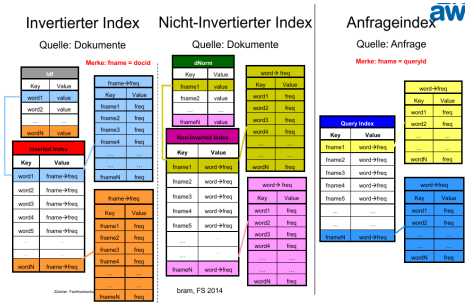
Genauigkeit, zu welcher eine invertierte Liste die Lokalisierung der Merkmale festlegt.

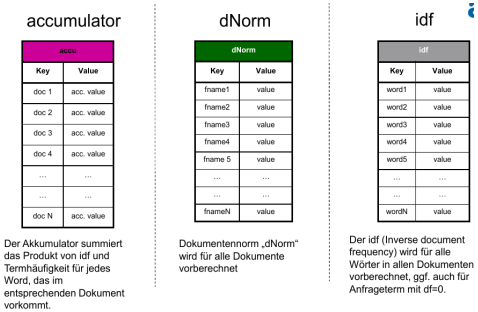
Grober Index: z.B. nur Blöcke von Texten in welchen mehrere Dokumente gespeichert sein können.
Mittlerer Index: z.B. Lokalisierung als Dok-ID gespeich.
Feiner Index: Index liefert ein Satz, Wort od. Bytes.

Lokalisierungsmerkmale

Extrem grosse Liste mit allen Merkmalen aus allen Dokumenten. Mit Hash-Tabellen Einträge lokalisieren.

MiniRetrieve





Evaluation

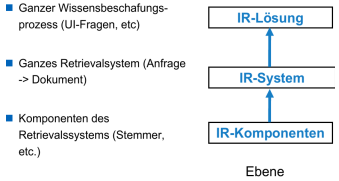
Nötig, um „Leistung" des Systems zu bewerten.

Zutaten einer Evaluation

Aufgabenstellung: Motivation, Ziele? Zielgruppe? Interne od. externe Evaluation? Black/Whitebox? Was ist Benchmarkgrösse?
Ausgestaltung: Was für Leistungsfaktoren (Umgebungsvariablen, Systemparameter)? Was für Leistungsmassstäbe? Was für Leistungsmasse (Effektivität, Effizienz, Akzeptanz)?
Testdaten: Daten über Users, IB, Dokumente mit Relevanz bzgl. IB.

IR-Evaluation nach Cranfield/SMART

Heute meistverbreitet in akademischen Kreisen.
3 Stufen der Evaluation der Retrievaleffektivität:



Aufgabenstellung

- Retrievaleffektivität des IR-Systems soll eval. werden.
- „Interne"/„direkte" Evaluation angestrebt, d. h., Effektivität direkt messen, nicht über Anteil an grösserem Resultat.
- Evaluation soll experimentell erfolgen.
- System während Evaluation als „black box" behandelt.
- Evaluation erfolgt im Vergleich zu optimalem Resultat.
- Soll quantitative Evaluation durchgeführen (Effektivität).

Ausgestaltung: Labortest

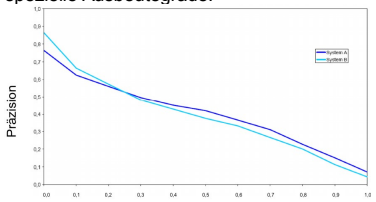
- Setup von operationellen Umgebungen abgekoppelt.
- wird von spezifischen Benutzern und Interpretationen ihrer IB abstrahiert (-> Anz. Umgebungsvariablen minimiert).
- Leistung als durchschnittliche Leistung über Anz. von Retrievalvorgängen gemessen.
- Leistungsmasse sind Präzision und Ausbeute.
- Dokumentdaten geeignet bestimmt und „eingefroren".
- „Batch-Setup" und Experimente beliebig **wiederholbar**.

Testdaten: Testkollektion

- Reihe IBn von fiktiven Benutzern.
- „eingefrorene" Menge Dokumenten als Suchdaten.
- Relevanzbeurteilungen von Dokumenten relativ zu IB.

Average Precision

non-interpolated: Durchschnitt der Präzisionswerte an Rängen aller relevanten Dokumente.
interpolated: Durchschnitt der Präzisionswerte für spezielle Ausbeutegrade.



Beliebteste „Ein-Zahl-Mass". Konvergiert gegen Fläche unter dem *Ausbeute/Präzisions-Graph*.
Aber: Widersprüchlichkeit von Ausbeute und Präzision bedingt, dass „Average Precision" nicht immer ein praxisrelevantes Mass darstellt.

Mean Average Precision

Durchschnittliche Average Precision über alle Anfragen.

Ausbeute

Um Ausbeute zu bestimmen, sämtliche Dokumente nach Relevanz bewerten. Heute aber unrealistisch.

=> Vorhandene Testollektionen wenn möglich nutzen.
=> Ausbeute auf alternative Art bestimmen.

Evaluations-Kampagnen

Idee: wenn genügend viele verschiedene Teilnehmer mit unterschiedlichen Systemen Evaluation durchführen, werden (fast) alle relevanten Dokumente von mindestens 1 System gefunden – müssen nicht alle Dokumente gelesen werden, um Ausbeute zu bestimmen.

Konkret z. B. als TREC-Kampagnen.

Gossip

Bei 1 Feedback eines unzufrieden Kunden haben 20 andere bereits selbes Problem gehabt, geben aber kein Feedback (20 mal Faust-im-Sack)

- System darf sich keine groben Ausrutscher erlauben.
- Avg. Performance kompensiert nicht für Ausrutscher.
- Man sollte sich nicht allzu viele Gedanken über **positive** Meldungen machen.

Known Item Retrieval

Idee: mit Suchmaschine nach „bekannten" Dokumenten suchen, simuliert „Da war doch was".

Mean Reciprocal Rank (MRR) = 1/R: Effektivität.

Ermittelt Durchschnitt über Anz. von Anfragen.

Kategorisierung und Klassierung

Kategorie: Dokument kann in keiner, einer od. mehreren Kategorien sein.
Klasse: Jedes Dokument in genau einer Klasse (es gibt möglicherweise auch Klasse *unklassiert* od. *andere*).

Dokumentkategorisierung

Kategorien bieten Mehrwert bei (manueller) Indexierung, Suche nach Information in Dokumentenarchiven od. News Feeds.

Indexierung: Verschlagwortung
Routing/Filtering: Themenprofil erstellen
Clustering: Gruppieren von Kollek. (Memos, E-Mails,..)
Annotation: Gruppieren von Dokumenten

Informationsstrukturen

Bei *Informationsstrukturen* geht es immer darum, Dokumente/Informationsobjekte in Verbindung setzen.

Typische Informationsstrukturen: Thesauri, Klassifikationen, Prozess-Modelle (Geschäftsprozesse), UML, Inferenzmodelle aus der künstliche Intelligenz, Navigationsstrukturen (Intranet), ...

Rocchio Model

Modelliert Kategorie C mittels Repräsentanten c. Dieser ist wie Dokument ein Vektor, aber hypothetisch. Initial aus positiven Beispielen generiert, regelmässig aktualisiert.

Algorithmus: Neue Dokumente d als Anfrage zum Vergleich mit den Repräsentanten . Falls s(d, c) > Δ Dokument d der Kategorie c zugeteilt. Δ ist Grenzwert, geeignet zu bestimmen.
Vorteile: einfach implementierbar, extrem effizient.
Nachteile: Nicht robust wenn Anz. von negativen Instanzen gross. Festlegung von Parametern knifflig.

k-Nearest Neighbor (kNN)

Verwendet Ähnlichkeitsmass (Eukl. Dist., cos) und Regel, wie Dokumente in D Kategorien zuzuordnen.

Algorithmus: Bestimme k ähnliche Dokumente zu D , d.h. k nächsten "Nachbarn". Ordne D min. 1 Kategorien von Nachbar zu.
Nachteil: Testkollektion nötig um Vorgang zu starten.

Erweiterte kNN-Klassifikation

Je weiter Dokument D vom Nachbar D_j entfernt (φ), desto weniger trägt es zum Entscheid bei, Dokument D in Kategorie C_j zuzuordnen.
Wobei kNN(D) Menge von k nächsten Nachbarn von D.
$a_{i,j}$=1 falls Dokument D_i zu C_j gehört, $a_{i,j}$=0 sonst.

Probleme: Richtige Wahl von k, Funktion sc, und max. Anz. zugeordneter Kategorien. Auch Schwellwert ab wann ein Dok. in mehrere Kategorien.
Vorteile: Effektiv und einfach, stabile Schwellw. zu finden
Nachteile: Langsam, Wahl einzelnen Wertes "k" schwer.

Bayes Klassifizierung

Gegeben: Kategorie C_i mit angemessenen Anz. von bereits zugeordneten Objekten (Trainingsdaten).
Methode: Bilde statistische Modelle aus Kategorien, um bestimmen zu welcher Klasse neues Objekt D gehört. $P(t|C_i)$ ∀ t bekannt, C_i, aber interessiert an $P(C_i|t)$ od. $P(C_i|D)$. D ist Menge von Merkmalen in Objekt/Dok. D.

Wsk, dass D zu C_i gehört:

$$P(C_i \mid D) = \frac{P(D \mid C_i)P(C_i)}{P(D)}$$

Mit D=(t1,M.,tn) und in Beziehung zu Klasse Ci:

$$P(D \mid C_i) = \prod_{j=1}^{n} P(t_j \mid C_i)$$

Gibt verschiedenen Wege um $P(t|C_i)$ zu berechnen: zähle Anz. Merkmale, binär (Nicht-/Vorkommen), gewichtet...

Vorteile: Sauberes Modell, einfach zu implementieren.
Nachteile: Performt sehr schlecht, typischerweise schlechter als andere einfache Verfahren. Unanbhängigkeitsannahme wohl zu simpel.

Regelbasierte Methode

Gewünschte Kategorie mittels Regeln beschreiben.
Bsp: Selektion geeigneter Beispieldokumenten, was für Suchanfrage ergibt diese Dokumente als Resultat?

Problem: Extrem schwierig, beständige (lange gültige) Anfragen zu Aufwand Konsistenz (manuell konstruierte Thesauri sind selten zyklenfrei) Unschärfe? (Transitivität?) Vollständigkeit? Bedeutungen abhängig von Zeit, Betrachter, thematischen Kontext. Wie global kann Thesaurus sein? Thesaurus kann helfen, Begriffe konsistent zu verwenden.
Use-Case: Regelbasierte Kategorisierung gut für "scharfe" Konzepte => Exprtensysteme.
Bsp: Regel für Kategorisierung von AUSTRALIAN DOLLAR sieht z.B. so aus: [australian-dollar-concept] or ([dollar-concept] and [australia-concept] and not [us-dollar-concept] and not [singapore-dollar-concept])

Verbesserungen dann möglich, wenn zu suchenden Konzepte in Feldern vom Text auftreten (z.B. Titel).

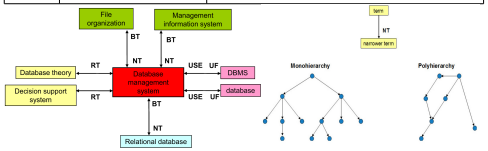
Vordefinierte Informationsstrukturen

Traditionelle Verfahren um Beziehungen zwischen Konzepten/Kategorien, Worten und Phrasen zu definieren. Unterscheidet zw. **Deskriptoren** und **Nicht-Deskriptoren**, so genannte „lead-in terms“. Klassifikationen bestehen nur aus NT-Relationen. Sind eher statisch und hauptsächlich im Bibliotheksumfeld zu finden.

Use-Case: Indexierungszwecke, Anfrageformulierung, verhindert „mismatch“ zw. Anfrage- und Dokumentmerkmalen. Abstimmung eines Retrieval-Alg.

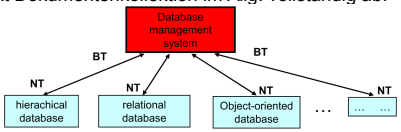
Komponenten: kontrolliertem Vokabular (z.B Kategorien!), Beziehung zw. einzelnen Merkmalen des Vokabulars, und Info zu Merkmalen und Beziehungen.

USE	synonym	use another term as descriptor
UF	used for	use this term instead
NT	narrower term	restricted concept
BT	broader term	umbrella concept
TT	top term	concept at head of hierarchy
RT	related term	similar concept

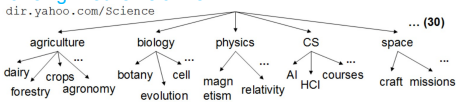


Herausforderungen: Aufwand, Konsistenz, Unschärfe, Vollständigkeit, Bedeutungen abhängig von Zeit, Betrachter, thematischen Kontext.

Klassifikation: Wenn Thesaurus nur NT (+BT) Beziehungen hat. Handelt dann um Monohierarchie. Deckt Dokumentenkollektion im Allg. vollständig ab.



Browsing-Webverzeichnis



Herausforderungen: Verwaltungsaufwand, Vollständigkeit, Bedeutungen ändern, Klassifikation muss ändern können, Menschen klassifizieren anders, etc..

Clustering

Menge von Objekten in "Cluster" von ähnlichen Objekten. Cluster können beliebige Untercluster besitzen.

Bsp: Resultat einer Suchanfrage durch Clustering geordnet. Hift User, effizienter weiterzusuchen od. Überblick über Kontext zu erhalten. Navigationshilfe.

Social Tagging

Social Tagging beschreibt Prozess bei welchem Benutzer Metadaten in Form von Schlüsselwörtern hinzufügen und diese Daten mit anderen Nutzern teilen.

Tag: "Aufkleber", beschreibt Informationsobjekt näher.

Tagging: Tags zu Informationsobjekt adden u. ggf. teilen.

Eigenschaften: Spontan durch Community, keine fixen Regeln, kein kontrolliertes Vokabular, keine Hierarchie, keine Beziehung zwischen Tags, Vergleiche streben nach Konsistenz bei Kategorisierung, analog zum Verhältnis Verschlagwortung vs. Volltextindexierung.

Ressourcen: Dokumente, Artikel, Einträge, etc.. Linking zw. Ressourcen gut untersucht (PageRank/HITS).

Vorteile für Betreiber: Nutzer kategorisieren, Tags reflektieren Benutzervokabular. Users erstellen Metadaten. Flexibel auf Veränderungen, bessere Abdeckung als mit kontrolliertem Vokabular, kostengünstig, stärkere Bindung an Angebot.

Vorteile für Benutzer: Auffinden eigener Informationsobj, leichteres Auffinden von gem. Informationsobjekten, mehrere Benutzer teilen, Anreiz schaffen für andere, Selbstdarstellung, Ausdruck eigener Meinung.

Nachteile: Zersplitterung der Kat. da unkontrolliertes Vokabular, Singular/Plural, mangelnde Struktur der Metadaten, mangelnde Präzision, ungenaue Deskriptoren

Wortwolke (Tag Cloud)

Methode zur Informationsvisualisierung, Wortliste alphabetisch aufgeführt und Grösse eines Worts seiner Wichtigkeit entspricht.

Web Search

Daten im Web sehr unbeständig (40% monatliche Änderung), unstrukturiert, teilweise schlechte Qualität.

Ziel: Finde qualitativ hoch stehende Resultate (nicht unbedingt Dokumente!), die für den Benutzer relevant.

Resultat: Statische Seiten (Dokumente, z.B. Texte, mp3, Photos, Videos), Resultat nicht ausbeuteorientiert, ggf. Probleme mit dynamischen Seiten.

Bedürfnisse

informationell: etwas lernen (~40%) z.B. Was ist ISIS?

navigational: will zu Webseite (~25%) z.B. SBB

transaktional: man will etwas tun (~35%), z. B. Zugriff auf Service “Wetter in Zürich”, Downloads, Shop, iTunes.

Graubereiche: Finde guten Hub, z.B. “Automiete in Seattle”, Erkundungssuche “see what’s there”.

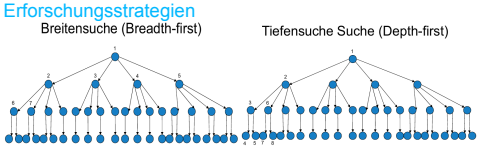
Spider

in-degree: Anz. Links, die auf Page zeigen.

out-degree: Anz. Links, die von Page weg zeigen.

Starte mit umfassenden Menge von URLs, von welchen Suche gestartet wird. Speichere Dokumente in D und Hyperlinks in E. Während Crawling wird Liste Q von URLs intern unterhalten. Wir extrahieren eventuell URL von Q od. fügen URL Q hinzu (Funktionen Dequeue() und Enqueue()).

Erforschungsstrategien



Tiefensuche erfordert Speicher nur für die Tiefe (d) mal den Verzweigungsgrad (b) (O(bd)) => Algorithmus benötigt aber zu viel Zeit, um nur 1 Zweig nachzugehen.

Breitensuche erforscht von Rootwebseite gleichmässig nach aussen. Erfordert Speicher für alle Konten des Graphen von früheren Ebenen (O(b^d)). Kompromiss nötig. Wie neue Links zu Q hinzugefügt (Enqueue()) und extrahiert (Dequeue()) hängt von Suchstrategie ab.

FIFO, first in first out: Ende von Q angeh., Breitensuche.

LIFO, last in first out: Anfang von Q angeh., Tiefensuche.

Heuristisches Anordnen von Q ergibt fokusorientierten Crawler, der Suche auf interessante Seiten ausrichtet. Z. B. auf Sites, Ordner, Sprachen, etc. begrenzbar.

Spreading Activation (SA)

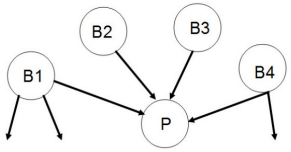
Grad der Ähnlichkeit zw. Di und Q, (bzw mit Sim(Di, Q)), wird durch bestimmte Anz. von Kreisläufen (normalerweise 1) zu verlinkten Dokumenten propagiert. Dazu Faktor λ verwenden. Nur Top r Dok. angepassen.

PageRank

Zu Beginn ist User auf zufälliger Page. Jeder Schritt bewegt User mit Wsk d zu zufällig gewählten Page (z. B. d = 0.15) od. zu zufällig gewähltem Nachfolger der aktuellen Page mit der Wsk 1-d (z.B. 1 - 0.15 = 0.85).

PageRank einer Webseite: Wsk, dass User zu beliebigem Zeitpunkt auf Page ist.

- Qualität von P: Q(P) = Q(B1)/3 + Q(B2) + Q(B3) + Q(B4)/2



$$PR^{c+1}(D_i) = (1-d)\frac{1}{n} + d\left[\frac{PR^c(D_1)}{C(D_1)} + \ldots + \frac{PR^c(D_m)}{C(D_m)}\right]$$

C(D1): Anzahl Outlinks von D1. Faktor 1/n ist umstritten.

Eigenschaften: Berechnung rekursiv, offline, anfangs ist Page-Rank aller Knoten 1.0, konvergiert gegen "Random Surfer"-WSK, anfällig auf Spam.

Nachteil: Muss theoretisch nach jeder Änderung neu berechnet werden.

Kleinberg Model/HITS

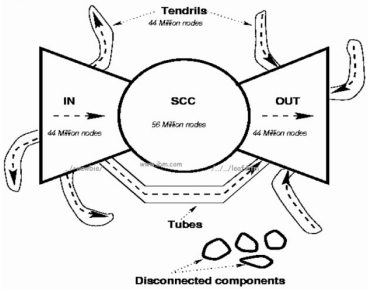
HITS: Hypertext Induced Topic Selection. Site ist Hub (verweist auf Quellen) od. Autorität (besitzt Inhalt). Werte iterativ berechnen (nach 5 Its schon relativ stabil). Nach jeder It normalisiert => Summe der Quadrate = 1.

Link-Analyse/Bowtie Model

Betrachtet Anz. Pages einer Site und Anz. Links einer Site oder Page.

$$P(n) = b \cdot n^{-\beta}$$
with b > 0 and $\beta > 1$

Anz. von Pages einer Site wird als n angenommen. P(n) ist die Wsk, dass die Site n Pages hat.



Enterprise IR

Suchsysteme für (un-)strukturierte Daten in Unternehmen; unterstützt wissensintensive Geschäftsprozesse.

Use-Case: Interne, vertikale Kollektionen, firmenübergreifende Suchsysteme (integrative Funktion!), Intranets, Desktop Search, Extranets, spezielle Suchapplikationen.

Impact: Grossen Markt für Enterprise Search. 70% aller Anfragen direkt bei Sites.

Aufsetzen

- Problem verstanden? (Informationen in 50 Dokumenten suchen != Informationen 50 Mio. Dokumenten suchen).
- Wer sind die Benutzer, was für IB?
- Wer "besitzt" die Daten? Welche Dokumente werden erschlossen?
- Risiko: Es besteht die Gefahr, dass Bedürfnisse mit einem bestimmten Produkt od. einen bestimmten Technologie assoziiert werden.

Anforderungen

- geschäftsrelevanten Entitäten (Produkte, Kundenbeziehungen, Lieferanten, etc.) „kennen“
- wichtigsten Benutzer- und Bedürfniskategorien „kennen“
- relevante Informationsquellen für welchen Prozess
- Transformation von unstrukturierter Information in strukturiertes Wissen unterstützen.

Technologien

- Topic Detection & Tracking
- Kategorisierung
- Clustering
- Information Extraktion: Message Understanding, Named Entity Recognition (erkennen von Personennamen, Ortsnamen, Produktenamen etc.)

Merkmal	Web IR	Enterprise IR
Wissensrepräsentation und Metainformation	Ist auf reine Suche ausgelegt, Stark heterogen, Qualität schwankt, Spam, Verlinkung von Dokumenten	Auf Unternehmensprozesse ausgelegt, heterogen strukturiert, dynamisch, wenig Metainformation
Umfang der Systeme	unendlich	Eher klein (Unternehmen)
Anfragesprache	Einfach, natürliche Sprache	Advanced Search, Browsing, Kategorien
Such-Kompetenz der Anwender	Beliebig: gering → trainiert	Gering, untrainierte Enduser
Fachkompetenz der Anwender	Beliebig: gering → trainiert	Hoch, professionelle Anwender mit meist sehr spezifischem Fachwissen