

Relevance Feedback

Seminararbeit

vorgelegt am

Lehrstuhl für Praktische Informatik IV
Prof. Dr.-Ing. W. Effelsberg
Universität Mannheim

im
Juni 2006

im Rahmen des Seminars

Bridging the Semantic Gap
im SS 2006

Betreuer
Stephan Kopf

von

Stephan Filipczyk

Inhalt

1. Motivation	3
Bedeutung des Relevance Feedback für moderne Systeme	4
Ziele des Relevance Feedback	4
2. Voraussetzungen	4
3. Herausforderungen	5
Informationsgrundlagen	5
Userinteraktion und Userverhalten.....	7
Entscheidungsprobleme.....	7
Lösungsansätze	8
4. Ausprägungen	9
„Klassisches“ vs. „Pseudo“ Relevance Feedback	9
Positives vs. Negatives Relevance Feedback.....	10
„Ostensive“ Relevance Feedback	11
5. Aufbau eines Relevance-Feedback-Systems.....	12
Die Wissensbasis.....	12
Der Learner	12
Der Selektierer	13
Der User.....	14
6. Relevance Feedback bei Videodaten.....	15
7. Zusammenfassung.....	17
8. Literatur	18

1. Motivation

Suchen im Web und in Datenbanken laufen nach einem immer wiederkehrenden Schema ab:

- Zunächst stellt der User eine Anfrage, die üblicherweise so knapp wie möglich und so umfangreich wie nötig ausfällt.
- Darauf liefert das System eine Auswahl von „Treffern“ zurück, die auf Grundlage der vorliegenden Daten der Anfrage so genau wie möglich entsprechen.
- Hieraus wählt der User die für ihn tatsächlich relevanten Treffer aus.

Dabei ist zu beobachten, dass der User dazu neigt, mit möglichst wenigen Abfragedaten eine sehr „gute“ Auswahl zu erwarten. Dabei stellen sich natürlich einige Probleme:

Zunächst ist nicht a priori davon auszugehen, dass der User ein Fachmann auf dem Gebiet ist, aus dem die zu findenden Daten stammen. Daher ist es durchaus möglich, dass der User zu Beginn gar nicht so genau weiß, was er eigentlich sucht. Dementsprechend kann es auch möglich sein, dass seine Anfrage nicht genau zu der erwarteten Auswahl führt.

Damit zusammenhängend stellt sich die Frage, ob die Kriterien der Anfrage sich nicht paarweise widersprechen und so zu unerwarteten Ergebnissen führen. Zum Beispiel kann man sich leicht vorstellen, dass eine textuelle Suche nach Bildern mit den Begriffen „Tag“ und „Nacht“ ein System leicht verwirren kann, falls dem nicht von vorne herein Abhilfe geschaffen wurde.

Ein weiteres großes Problem besteht darin, ob das System mit den Abfragedaten überhaupt „etwas anfangen“ kann. Z.b. wird ein System, das englischsprachige Begriffe erwartet, mit deutschen Suchwörtern wenig anfangen können.

Auf der anderen Seite gilt zu bedenken, auf welcher Grundlage eine Auswahl getroffen werden kann, damit die Ergebnisse einer Anfrage, die das System für am relevantesten hält, auch diejenigen sind, die für den User am relevantesten sind.

Diese Liste lässt sich mit Sicherheit fortsetzen, doch hier soll gleich ein möglicher Ansatz genannt werden, der diesen Problemen begegnet.

Relevance Feedback erweitert das oben geschilderte Verfahren um ein Zusatzsystem. Den Anfang macht wieder der User mit einer Anfrage, das System mit einer Suche, gefolgt von der Präsentation der Ergebnisse. Diese Ergebnisse werden nun vom User bewertet nach „relevant“ und „nicht relevant“. Diese Bewertungen werden nun vom System in eine neue Suche miteinbezogen. Der genaue Ablauf wird in den Kapiteln 4 und 5 erläutert.

Bedeutung des Relevance Feedback für moderne Systeme

Verfahren, die RF implementieren sind bereits aus den späten 60er Jahren für textuelle Recherchen bekannt [10]. Man könnte also erwarten, dass es mittlerweile bessere Methoden gibt.

Tatsächlich hat sich herausgestellt, dass man durch RF mit relativ wenig Rechenaufwand (je nach System und Verfahren) die Ergebnisse bereits signifikant verbessern kann. Außerdem lassen sich die bereits bewährten und gut verstandenen Methoden auf moderne Systeme übertragen, so dass über textuelle Suche hinaus auch Bilder und sogar Filmdatenbankanfragen merklich verbessert werden können.

Ziele des Relevance Feedback

Relevance Feedback soll dem User ermöglichen, auf Grundlage von sehr wenigen Eingabedaten zu einem „vernünftigen“ Suchergebnis zu gelangen. D.h. der User soll nicht unnötig viel Energie in die Formulierung einer Suchanfrage stecken müssen, um zu einem annehmbaren Ergebnis zu kommen.

Unbefriedigende Ergebnisse sollen mit möglichst geringem Aufwand verbessert werden. In der Praxis empfindet es der User als weit weniger aufwendig, durch Mausklick eine Auswahl zu bewerten, als eine Suchanfrage durch weitere Begriffe oder allgemeiner gesagt zusätzliche Informationen zu erweitern.

Darüber hinaus sollen „Missverständnisse“ zwischen User und System („Semantic Gap“) minimiert werden. Relevance Feedback soll es also ermöglichen, auch verwirrende oder widersprüchliche Anfragen, die auf sprachlicher Ungenauigkeit gründen, für das System besser „verständlich“ zu machen.

2. Voraussetzungen

Speziell beim Einsatz des RF im Zusammenhang mit Multimediataten¹ gibt es einige Annahmen, die zwingend gegeben sein müssen [1].

- Die Unterscheidung in „relevant“ und „irrelevant“ muss an Hand der zur Verfügung stehenden Deskriptoren möglich sein.
- Es gibt einen *relativ einfachen* Zusammenhang zwischen dem Beschreibungsraum und den (gemeinsamen) Charakteristiken der Bilder, nach denen der User sucht.
- „Relevante“ Bilder sind ein *kleiner Teil* der gesamten Datenbank.

Im Gegensatz zu frühen Arbeiten zu diesem Thema ist mittlerweile bekannt, dass der User nur ein sehr begrenztes Maß an Feedback erzeugt. Ausführliches Feedback kann also **nicht** angenommen werden.

¹ Im Folgenden wird von Bildern gesprochen. Die Voraussetzungen und folgende Aussagen lassen sich jedoch auf andere multimediale Inhalte ausdehnen.

3. Herausforderungen

Informationsgrundlagen

Bei textueller Suche ist noch relativ klar, wonach gesucht werden kann: Worte. Im Titel, im Inhalt, in den Metadaten. Aber wie sieht es bei multimedialen Inhalten aus?

Dateinamen

können kryptisch sein (z.B. xz-9879-abc.mp3) oder sogar verwirrend.

Das Beispielbild (Abb.1) trägt den Namen „bank.jpg“. Soweit eindeutig – allerdings nicht, wenn man nach Logos von Kreditinstituten mittels eben dieses Begriffes sucht.



Abbildung 1 [Quelle: <http://www.photohomepage.de>, 12]

Textuelle Annotationen

sind gut verwendbar, erzeugen aber hohe Kosten, wenn sie von Hand erstellt werden müssen. Die Automation gestaltet sich leider schwierig, denn wie sollen komplizierte semantische Zusammenhänge maschinell erfasst werden? Erkannt werden können z.B. in Filmen die mitgelieferten Untertitel, oder in Audiostücken (Musik) der bekannte Text [8].

„Low-Level“ – Features

bezeichnen Eigenschaften, wie Histogramme, Color Structure Descriptors, Kanteninformationen oder Bewegungsinformation in Filmen. Diese Merkmale tragen allerdings wenig semantische Information in sich.

„High-Level“ – Features

semantische Informationen mit hohem Beschreibungswert. Ein Beispiel wäre die Aussage, dass ein Gesicht eine spezielle Person zeigt. Dies zeigt auch gleich noch ein weiteres Problem auf:

Granularität

Soll nur nach ganzen Bildern gesucht werden können, oder auch nach identifizierten Bildinhalten und lohnt es sich, diese im Voraus zu berechnen? Kompliziertere Deskriptoren, die im Rahmen von Metadaten vorberechnet vorgehalten werden können lohnen durchaus diesen Aufwand. Mit Hilfe von Ontologien können so z.B. bereits durch ein Erkennungsverfahren identifizierte Objekte effizient gespeichert werden.

Ein Beispiel für ein System zur Bildersuche

Zur Vorauswahl wird ein System [5] auf der Basis von textueller Suche verwendet (Yahoo Bildersuche). Anschließend werden die Bilder in Form von „Thumbnails“ präsentiert und der Bewertung durch den User preisgegeben. Die Granularität ist also „ganzes Bild“ und es werden zunächst Dateiname und textuelle Annotationen zur Suche herangezogen.

Als Maß für die Ähnlichkeit dient in diesem System eine Metrik über die Eigenschaften (Deskriptoren) „Color Histogram“ und „Color Structure Descriptor“ (CSD) [6,7]. Ersterer teilt die Farben eines Bildes in „Bins“ ein und zählt dann für jedes Bin die Anzahl der Pixel, deren Farbe in das Bin fallen.

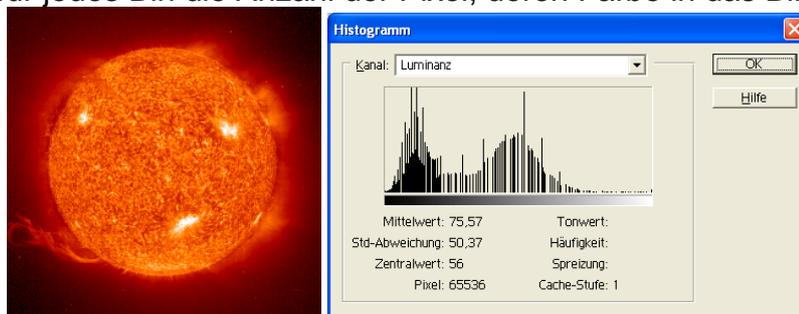


Abbildung 2: Bild und zugehöriges Histogramm [13]

Der CSD ist ein ähnliches Maß. Hier werden die Farben wieder in Bins quantisiert, nur dass diesmal anders gezählt wird. Das Bild wird in 8x8 Pixel-Bereiche unterteilt und getrennt betrachtet.

Jedes Bin, aus dem eine Farbe im aktuell betrachteten Bereich vorkommt wird um 1 inkrementiert. (siehe Abbildung 3)

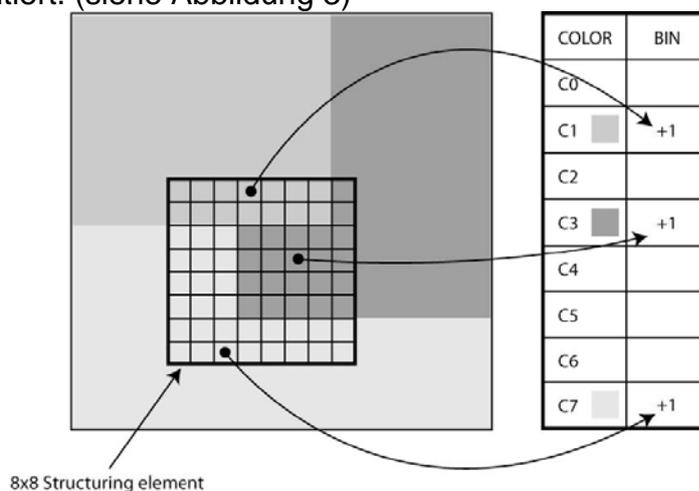


Abbildung 3: Color Structure Descriptor [6]

Dieses System baut also auf zwei recht simple Low-Level-Features und erreicht erstaunliche Verbesserungen im Vergleich zur Suche ohne Histogramm oder CSD, wie die Tabelle aus dem Artikel [5] zeigt:

Query	Distance from top for most relevant
Yahoo only	24,6875
Yahoo+histogram	12,6875
Yahoo+CSD	17,3750

Userinteraktion und Userverhalten

Der User hat das Ziel, mit möglichst wenig Energieaufwand (für ihn selbst) eine möglichst gute Ausbeute an relevanten Informationen vom System zu erhalten. Dabei stellen sich folgende Probleme:

- Gibt der User dem System genügend Feedback?
Er möchte ja schließlich die Arbeit abgenommen bekommen, auf der anderen Seite braucht das System auch ein gewisses Minimum an Informationen, um damit arbeiten zu können.
- Weiß der User überhaupt, was er sucht?
Was sich auf den ersten Blick abwegig anhört ist nicht so weit hergeholt. Kennt sich ein User auf einem bestimmten Gebiet noch nicht so gut aus, so hat er evtl. zu Beginn der Suche noch keine genaue Vorstellung, von dem, was er als akzeptables Ergebnis betrachtet. Genauso kann sich während der Suche herausstellen, dass Begriffe bzw. Informationen der ursprünglichen Anfrage doch nicht relevant waren.

Zudem erwartet der User vom System nahezu Echtzeitverhalten [1]. Das bedeutet, er fordert vom System, fast direkt nach „Abschicken“ der Anfrage erste Ergebnisse betrachten zu können.

Entscheidungsprobleme

Ein Punkt, der für den Wert eines Suchergebnisses ausschlaggebend ist, ist die Frage, wonach gesucht wird. Soll es *ein bestimmtes* Bild oder Musikstück sein („Target-Search“) oder möchte der User eine kleine Auswahl an *ähnlichen* Bildern haben („Category-Search“)?

Letzteres führt gleich zum nächsten Entscheidungsproblem: Wann kann man bei einem Bild von „ähnlich“ sprechen und auf Grund welcher Informationen?

Ein Beispiel:



Abbildung 4 [14]

Alle drei Bilder in Abbildung 2 zeigen die Sonne. Doch sind sie sich „ähnlich“? Und falls ja in welchen Eigenschaften ?

Und schließlich die Entscheidung, die dem Verfahren erst Sinn gibt: Wo ist die Grenze zwischen „relevant“ und „irrelevant“, wenn man keine genauen Aussagen treffen, sondern nur ein gewisses Maß an „Ähnlichkeit“ angeben kann?

Lösungsansätze

Dieser Fülle von Herausforderungen kann man nicht mit der einen „eierlegenden Wollmilchsau“ begegnen. Viel mehr ist es ein ganzes Bündel von Methoden, die sich gegenseitig beeinflussen und meist gleich gegen mehrere Probleme wirken.

Angelehnt an den Denkansatz der Expertensysteme besteht ein Mittel darin, sich auf eine klar begrenzte Umgebung oder auch „Domain“ zu beschränken. Das vermindert zum einen die Größe der Datenbank und zugleich die Menge der sinnvollen Anfragen, die wiederum bedeutend schneller zu akzeptablen Ergebnissen führen.

Ein weiterer Ansatz besteht darin, sich auf sog. „*Low-Cost*“-Deskriptoren zu beschränken. Dies ist nicht zu verwechseln mit „*Low-Level*“-Deskriptoren! Ausschlaggebend ist hier nicht die Komplexität per se, sondern der Aufwand, mit dem diese Deskriptoren erlangt, berechnet und verglichen werden können. Dies ermöglicht es, auch Daten mit höherem Informationsgehalt bereit zu stellen, indem man sie zum Beispiel im Voraus berechnet. Die Möglichkeit eines schnellen Vergleichs vorausgesetzt, sind diese Daten von hohem Wert für die Verfahren des Relevance Feedback.

4. Ausprägungen

„Klassisches“ vs. „Pseudo“ Relevance Feedback

In der Literatur ist neben dem Relevance Feedback auch immer wieder von „Pseudo Relevance Feedback“ die Rede. Die beiden Begriffe werden in der Literatur oft synonym, teilweise auch widersprüchlich verwendet, so dass eine eindeutige Trennung zwischen beiden kaum möglich ist. Daher unterscheidet auch der vorliegende Text – mit Ausnahme dieses Abschnittes – nicht zwischen beiden. Dennoch sollte erwähnt werden, dass es diese Unterscheidung gibt und dass man sich durch die Literatur nicht verwirren lassen sollte.

Aus Sicht des Autors trifft am ehesten folgende Unterscheidung zu:

Von rein textbasierten Verfahren her kommend, bezeichnet das „klassische“ Relevance Feedback die Methode durch Feedback die ursprüngliche Suchanfrage selbst zu verfeinern und neu zu suchen.

Pseudo Relevance Feedback dagegen geht davon aus, dass die Auswahl der initialen Suchanfrage schon relevante Elemente liefert, nur in der „falschen Reihenfolge“. Feedback führt hier zu einer Neubewertung der Relevanz und Reihenfolgeänderung der Auswahl bzw. „Auswahl der Auswahl“, je nach Verfahren.

Beispiel

Beispielhaft sei hier die Suche nach der Webseite einer Fachbereichsbibliothek der Universität Mannheim.

Ausgehend von den Suchbegriffen „Bibliothek“ und „Mannheim“ erhalten wir vom System eine Auswahl zurück, die wie folgt aussehen könnte:

1. Mannheim Stadtbibliothek
2. Bibliotheken in Mannheim
3. Bibliothek der Berufsakademie
4. FH Mannheim: Bibliothek
5. Leitseite der Bibliothek (Institut für Deutsche Sprache IDS)
6. UB Mannheim: Bibliothek digital
7. ...

In Systemen mit Relevance Feedback würde nun der User bestimmte Links als „relevant“ bzw. „irrelevant“ markieren. Bis hier besteht zwischen beiden Ansätzen noch kein Unterschied.

Angenommen der User hätte nun den Eintrag 6 als „Relevant“ markiert.

Bei Verwendung des *Klassischen RF* würde das System nun z.B. den mit dem Eintrag verknüpften Begriff „Universität“ **der Anfrage hinzufügen** und eine erneute Suche auf dem Web starten. Ergebnis (mit Google, Anfrage manuell geändert):

1. UB Mannheim: Bibliothek digital
2. Universitätsbibliothek Mannheim
3. UB Mannheim: Katalog der Universitätsbibliothek
4. Medizinisch-Wissenschaftliche Bibliothek der Fakultät für Klinische Medizin Mannheim
5. ...

Pseudo Relevance Feedback würde hier lediglich die bisherige Auswahl anhand der markierten Einträge neu bewerten, d.h. umsortieren. Eine erneute Suche auf dem kompletten WWW würde hier nicht mehr durchgeführt. Folglich würden auch hier die bereits gefundenen (aber evtl. nicht angezeigten) Einträge, die die Eigenschaft „Universität“ zusätzlich zu denen aus der Ursprungsanfrage erfüllen auf die obersten Plätze rücken.

Auf den ersten Blick mag dies wenig sinnvoll erscheinen, gehen hierdurch doch vermeintlich Treffer verloren, die eine erneute Suche mit den erweiterten Kriterien über den kompletten Suchraum finden würde. Führt man sich allerdings vor Augen, dass bereits eine Suche nach Bildern ungemein komplizierter ist, als rein textuelle Suche und Suche nach Filmen noch mal um Größenordnungen komplexer ist, so kann die letztere Methode deutliche Performancevorteile für sich verbuchen.

Positives vs. Negatives Relevance Feedback

Wie schon gesehen, wird beim RF eine Bewertung nach „relevant“ und „irrelevant“ durch den User vorgenommen.

Eine Feinheit des Verfahrens besteht nun darin, ob man den User nur relevante Elemente markieren lässt und somit quasi den Rest als irrelevant betrachtet [4]. Dieses Verfahren entspräche dem *Positive RF*. Umgekehrt arbeiten manche Verfahren damit, die Elemente der Ausgangsanfrage als „relevanteste“ zu betrachten, den User die „irrelevanten“ Elemente der Auswahl markieren zu lassen und die (restliche) Auswahl dann mit den neu gewonnenen Informationen neu zu bewerten.

Bei Verwendung des positiven RF hat sich gezeigt, dass in den ersten Iterationen der „Verbesserungseffekt“ sehr stark ist, dann aber schnell an seine Grenzen stößt.

Negatives RF hat den gleichen Vorteil, jedoch hält der Effekt länger an. Diese Methode hat jedoch den entscheidenden Nachteil, dass man es auch übertreiben kann: Gibt man zu viel negatives Feedback, können Eigenschaften, die für die Suche tatsächlich wichtig sind als „irrelevant“ eingestuft werden. In der Folge verschlechtern sich die Ergebnisse also ab einem gewissen Punkt. Man spricht dann von einer „Zerstörung der Anfrage“.

Ein weiterer Punkt, der gegen den Einsatz von negativem Feedback alleine spricht, ist das Qualitätsmerkmal der Rate der „falschen Richtigen“. D.h. Objekte, die als relevant klassifiziert werden, es aber tatsächlich nicht sind. Wie kann das passieren? Berücksichtigt man ausschließlich negatives Feedback, so gewinnt man sehr viel Information über Objekte, die möglichst weit von der Anfrage „weg“ sind. Das heißt, dass es für eine hohe Wertung zu „relevant“ ausreicht, in den einschlägigen Eigenschaften den Irrelevanten *nicht* zu ähneln. Dies hat nicht zwangsläufig zur Folge, dass das besagte Objekt, dem verfolgten Ziel nahe kommt.

Studien [4] haben gezeigt, dass eine Mischung aus beiden mit schwacher Gewichtung des negativen RF die besten und stabilsten Ergebnisse liefert.

„Ostensive“ Relevance Feedback

Diese spezielle Form macht sich die Grundlegenden Annahmen zu nutze, dass der Nutzer zu Beginn einer Suche nur eine vage Vorstellung davon hat, was er sucht und sich dem Ziel im Laufe der Feedback-Iterationen immer klarer wird [3]. Wie ist dies zu bewerkstelligen? In jeder Feedbackiteration werden die zuletzt gewonnenen Informationen am stärksten gewichtet und die Ausgangsanfrage tritt mehr und mehr in den Hintergrund.

Beispiel

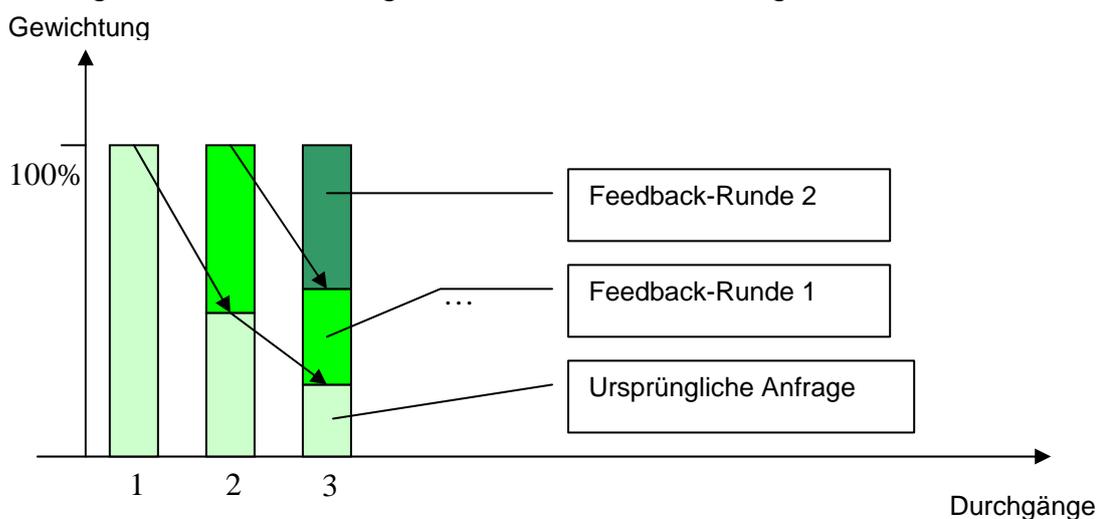
Beispielhaft sei hier die Suche nach wissenschaftlichen Papern genannt. Besonders für die Teilnehmer des Seminars sollte dies nun mittlerweile leicht nachzuvollziehen sein.

Zu Beginn haben wir einen User, der sich momentan mit einem bestimmten Thema beschäftigt und oberflächliche Kenntnisse oder vielmehr Ahnungen oder Arbeitsanweisungen besitzt. Um einen fundierten Vortrag halten zu können, begibt er sich nun an unser Beispielsystem mit ostensive Relevance Feedback und beginnt seine Suche damit, dass er eine Anfrage erstellt, die die Oberbegriffe enthält, aus denen sein Vortragsthema besteht.

Das System liefert ihm eine breite Auswahl an Papern zurück, von denen er einige auswählt, deren Zusammenfassungen er auf Relevanz hin prüft und sie entsprechend markiert. Anschließend überlässt er dem System eine Neubewertung, die seine Markierungen berücksichtigt.

Jedoch gewinnt er mit jedem neuen Paper, das er an- oder bei besonderem Interesse durchliest an Wissen über die Thematik, die in seine Bewertungen mit eingehen.

Diesen Umstand macht sich das System zu nutze: Es gewichtet jeden Durchgang von Bewertungen verschieden stark. Genauer gesagt wendet es eine „Decay-Funktion“ an, die die Ursprungsanfrage und die folgenden Bewertungen mit jedem neuen Durchgang an Gewicht verlieren lässt und die zuletzt getroffene Bewertung am stärksten berücksichtigt.



5. Aufbau eines Relevance-Feedback-Systems

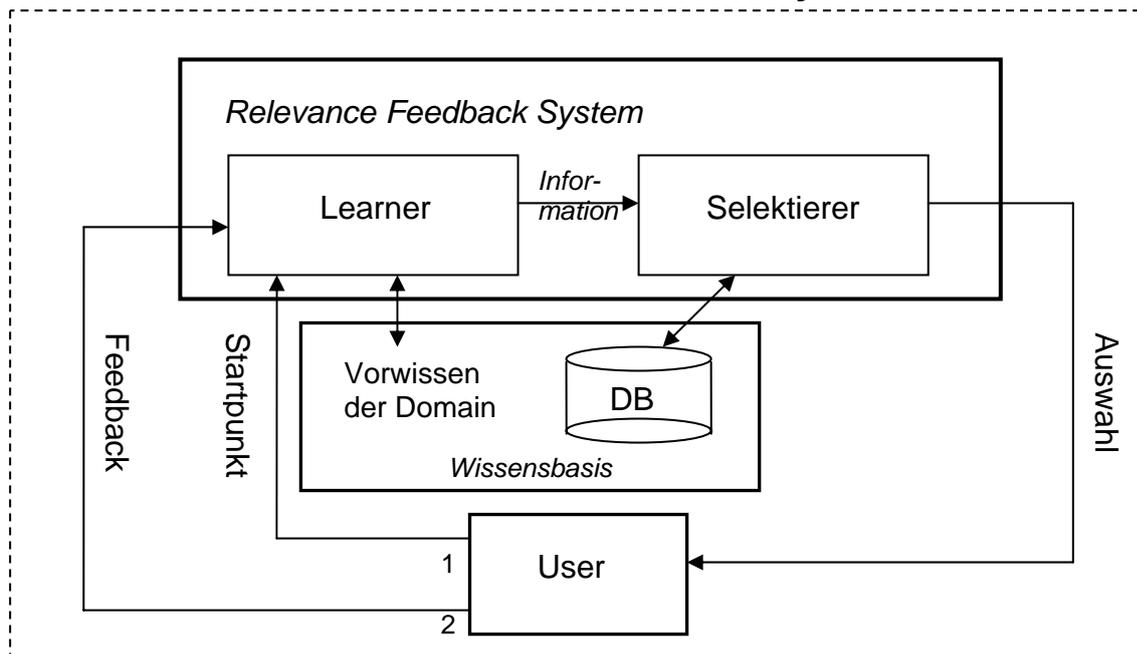


Abbildung 5: Schematische Darstellung eines RF-Systems.

Die obige Abbildung zeigt die grundlegenden Elemente eines RF-Systems[1]. Welche Ausprägung auch immer verwendet wird wirkt sich nur auf die Implementierung der einzelnen Komponenten aus.

Diese sind im Einzelnen:

Die Wissensbasis

Sie umfasst sowohl die zu verarbeitenden Elemente, also z.B. Texte, Bilder, Filme, Audiodateien, als auch zusätzliches „Wissen“, z.B. in Form von Metadaten [8], vorberechneten Deskriptoren oder grundsätzlichem Wissen über die Domain der Daten. Letzteres kann sich z.B. in einer vorgefertigten Gruppierung der Daten niederschlagen – man erinnere sich an den Vortrag über Ontologien [11].

Der Learner

Die Aufgabe der „Lerneinheit“ ist es, aus den Eingaben des Users ein Auswahl- bzw. Bewertungskriterium zu formulieren. Dazu muss natürlich Wissen aus mindestens einer Feedback-Iteration berücksichtigt werden.

Eines der Probleme, die sich dabei stellen besteht darin, dass die Lerneinheit mit sehr wenig Trainingsdaten auskommen muss, wobei gleichzeitig die Zielklasse sehr kompliziert sein kann. Auch kann der Lerner mit Negativbeispielen schlecht auf das Ziel trainiert werden. Dies stellt beim positiven RF ein echtes Problem dar: Alle nicht markierten Beispiele sind hier Negativbeispiele und damit deutlich in der Überzahl.

Auch komplett voneinander unabhängige Eigenschaften stellen hier ein Problem dar, denn der Lerner versucht, Zusammenhänge zu erkennen.

Ein weiteres Problem ist schließlich die Zeit. Die Berechnungen des Learners nehmen einen Großteil der Bearbeitungszeit in Anspruch. Daher gilt es ihn so schnell wie möglich zu machen, will man doch die Performanceanforderungen des Users befriedigen.

Eine oft verwendete Lösungsmöglichkeit sind „**Support Vector Machines**“ [9]. Support Vector Machines sind ein Bündel von Lernmethoden zur Klassifizierung. Dabei werden Punkte (nicht notwendigerweise im R^2) durch eine „Hyperplane“ in zwei Klassen getrennt, sofern diese existiert. Diese hat den größtmöglichen Abstand von den nächsten Punkten. Diesen Abstand nennt man den „Margin“, die Punkte, die der Hyperplane am nächsten sind „Support Vektoren“.

Ist eine SVM hinreichend mit Trainingsdaten versorgt, kann sie eine Klassifizierung von neuen Punkten vornehmen.

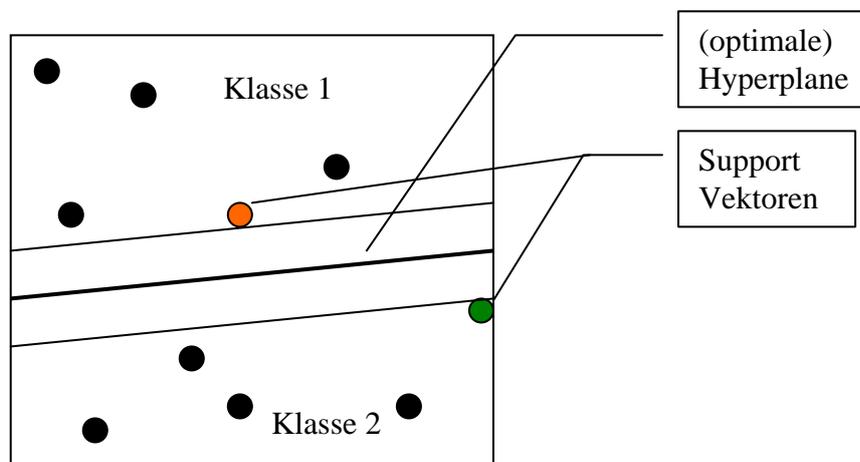


Abbildung 6: Support Vector Machine.

Der Selektierer

Dem Selektierer kommt die Aufgabe zu, eine Auswahl aus der Datenbank zu ziehen und dem User zu präsentieren. Er verfolgt dabei zwei wesentliche, jedoch widersprüchliche Ziele:

1. Dem User sollen so viele **relevante** Objekte wie möglich geliefert werden.
2. Anhand der Auswahl sollen dem User möglichst viele Informationen darüber „entlockt“ werden, was „relevant“ ist und was nicht.

Der Widerspruch dieser beiden Ziele liegt auf der Hand:

Ziel Nummer 1 hat zur Folge, dass alle Bilder (oder Objekte) in der Auswahl der Anfrage möglichst nahe kommen, sich also in möglichst vielen Eigenschaften sehr ähnlich sind. Das macht es ungeheuer schwierig, *nicht relevante* Eigenschaften eindeutig zu bestimmen und kann zu Fehleinschätzungen führen.

Ein naiver Ansatz zur Lösung dieses Dilemmas wäre, in die Auswahl absichtlich Objekte einzufügen, die möglichst weit von der Anfrage weg sind, um irrelevante Eigenschaften besser identifizieren zu können und damit eine schärfere „Grenze“ zwischen „relevant“ und „irrelevant“ zu erlangen. Für den User ist das allerdings sehr störend und entgegen der Erwartung wird er nicht

zwingend mehr negatives Feedback liefern, sondern nur mit dem System unzufrieden sein. Tatsächlich werden Methoden dieser Art, die Hybride aus den folgenden Ansätzen sind, trotzdem verwendet, um die Vorteile beider auszunutzen, jedoch will die Gewichtung dabei mit viel Fingerspitzengefühl abgewogen sein.

- **„Most Positive Images“**
die in der Literatur am häufigsten vertretene Methode konzentriert sich auf das erste Ziel. Hintergedanke dabei ist, dass dem User möglichst früh mögliche Ziele seiner Anfrage präsentiert werden sollen. Ist die Zielklasse jedoch kompliziert, kann es etwas länger dauern, bis sie gut genug identifiziert ist.
- **„Most Informative Images“**
Diese Variante konzentriert sich auf das zweite Ziel und eignet sich besonders für Target-Suchen, also Suchen nach einem bestimmten Objekt.
Dabei werden dem User die Objekte präsentiert, deren Bewertung durch ihn die größtmögliche Fülle an Zusatzinformation über die Zielklasse für die nächste Iteration liefern.

Der User

Kann hier auf Grund seiner Aufgabe getrost als „Teil des Systems“ genannt werden. Auch soll sein Einfluss auf den Zielfindungsprozess nicht unterschätzt werden.

Schon ganz zu Beginn einer Recherche können Fehler gemacht werden. So hat der „Startpunkt“, also die ursprüngliche Anfrage, auf die die folgenden Feedback-Iterationen aufbauen, entscheidenden Einfluss darauf, ob und wenn, wie schnell das System ein akzeptables Ergebnis liefert.

Der zweite Interaktionspunkt ist die Bewertung der Auswahl durch den User. Trifft er hier oft widersprüchliche Entscheidungen, kann man sich leicht vorstellen, dass der Learner schnell überfordert sein kann. Aber nicht nur „falsches“ Feedback führt zu Problemen. Zu wenig Feedback wird dem System kaum eine Chance lassen, die Auswahl innerhalb der Iteration signifikant zu verbessern. Auch der Verzicht auf negatives Feedback bedeutet zwangsläufig auch den Verzicht auf die Vorteile, die dem System und damit auch dem User dadurch entstanden wären.

6. Relevance Feedback bei Videodaten

Videodaten stellen eine besondere Herausforderung für Feedback-Systeme dar. Nochmal zur Erinnerung:

- Der User wünscht *fast Echtzeitverhalten*.
- Der User wünscht schnell zu einem Ergebnis zu kommen.
- Der User wünscht so wenig wie möglich eingreifen zu müssen.

Alleine diese Anforderungen stellen die Systeme für Videodaten vor gravierendere Probleme als Text- oder Bildsysteme. Warum?

„Videos sind mehr als die Summe ihrer Bilder“: Zum einen enthalten Videos neben Bildinformationen – und zwar alle Informationen ihrer Einzelbilder – auch meist Audio. Diese Ton-Spuren können selbst wieder die vielfältigsten Informationen mit semantischem Gehalt beinhalten.

Zum anderen enthalten Videos zusätzliche Informationen in Form von Bewegung. Das hört sich zunächst nicht sehr kompliziert an, doch: Objekte können sich bewegen oder die Kamera selbst. Bewegungen zeichnen sich durch Richtung und Geschwindigkeit aus. Doch was ist z.B. ein Zoom? Er wird evtl. auch als Bewegung detektiert.

Auch das Problem der Granularität erhöht sich um einen Faktor: Nicht nur, dass man anhand der Bewegung und Kanten Objekte identifizieren kann, man kann sich auch fragen, ob man nach ganzen Filmen, Szenen oder Einstellungen suchen können soll.

Und schließlich gilt noch zu fragen, welche Metriken man verwendet, um auf ein Maß für „Ähnlichkeit“ zu kommen und wenn dies geschafft ist, wie präsentiert man die Auswahl?

Letztlich ist also zu sagen, dass die ungeheure Menge an Informationen einem Suchsystem auch ungeheure Rechenleistung abverlangt. Deshalb sind zum einen ausgeklügelte Algorithmen von Nöten, zum anderen muss man zwangsläufig Kompromisse eingehen und Abstriche hinnehmen.

Ein Beispiel,

das einige Verfahren und Kompromissvorschläge aufzeigt wird in dem Artikel [3] „Video information retrieval using objects and ostensive relevance feedback“ dargestellt.

Dieses System geht von vornherein schon einen mächtigen Kompromiss ein: Die Datenbasis beschränkt sich auf die Folgen der US-Zeichentrick-Serie „Die Simpsons“.

Diese Entscheidung hat gleich mehrere Vorteile: Der Suchraum wird sehr stark eingeschränkt, so dass es sogar möglich ist, nach Objekten, nämlich den Hauptcharakteren, zu suchen. Vereinfacht wird dies noch durch die Tatsache, dass die Zeichentrickfiguren durch klare Striche, also scharfe Kanten begrenzt werden.

So bereitet die Forschungsgruppe die einzelnen Folgen auf, indem der DVD-Inhalt nach MPEG-1 transcodiert wird und die Informationen der „closed Captions“ (versteckte Bildtitel im Videosignal) festgehalten werden. Anschließend wird eine Shot-Boundary Detection und eine Objekterkennung durchgeführt. Diese Daten befinden sich in der Datenbank.

Die Anfrageoberfläche besteht aus einem Textfeld und einem Zeichenfeld. Hier können textuelle Daten, sowie eine „Beispielzeichnung“ sowie Farbinformationen angegeben werden.

Als Antwort wird dem User eine Auswahl an Einstellungen präsentiert und zwar in Form von Standbildern. Anhand dieser kann er Einteilungen in relevant oder irrelevant vornehmen. Als weitere Quelle für Informationen hat der User die Möglichkeit, einen Shot „genauer“ zu betrachten. Hierzu erhält er in einer weiteren Maske eine Auswahl von Einzelbildern eines bestimmten Shots nebst einem vergrößerten Einzelbild, auf dem die detektierten Objekte markiert sind. Durch Mausklick kann er hier sogar Objekte als relevant (oder irrelevant) markieren.

Um den Rechenaufwand beherrschbar zu halten, wurden mehrere Verfahren getestet, wie man zwei „Shots“ miteinander vergleicht. Neben Verfahren, die mit Durchschnittswerten arbeiten und einem Verfahren, das jedes Bild des einen mit jedem Bild des anderen Shots vergleicht, wurde auch getestet, das Mittlere Bild eines Shots als „repräsentativ“ zu nehmen und allein auf seiner Grundlage zu arbeiten.

Erstaunlicherweise verschlechtern sich dadurch die Ergebnisse bei weitem nicht so signifikant, dass die Performanceeinbußen durch eines der anderen Verfahren gerechtfertigt wäre.

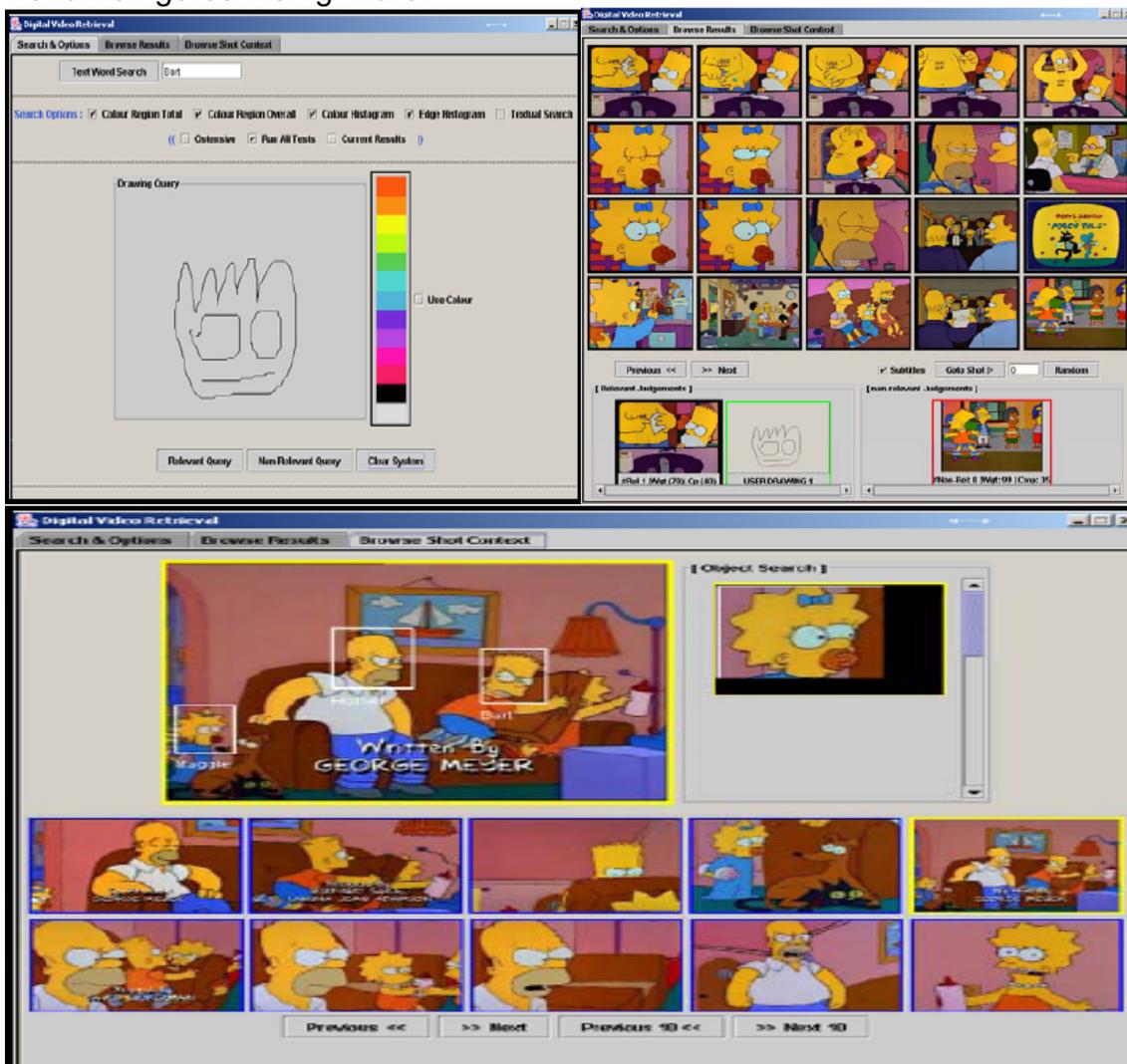


Abbildung 7: Masken des Simpsons-Systems. [3]

7. Zusammenfassung

Relevance Feedback ist ein Verfahren, das die Ergebnisse von Suchvorgängen in Datenbanken auf eine Weise verbessert, die den Bedürfnissen des Users entgegen kommt. Es arbeitet auf Daten, die zum Zwecke der Suche ohnehin schon existieren und ist an die verschiedensten Anwendungsgebiete anpassbar. Speziell im Bereich von Multimediadaten werden z.Z. Forschungsleistungen erbracht, die einen Ausblick auf sinnvolle und ökonomisch lohnende Anwendungen geben.

Selbst mit simplen Implementierungen lassen sich performante Systeme verwirklichen, die einen signifikanten Beitrag zur Userzufriedenheit leisten. Mit den genannten Möglichkeiten der Anpassung jedes seiner Komponenten ist auch der Einsatz des Relevance Feedback als Webanwendung denkbar.

8. Literatur

- [1] CRUCIANU, M.; FERECATU, M.; BOUJEMAA, N. (2004): [Relevance feedback for image retrieval: a short survey](#), 20 p., in State of the Art in Audiovisual Content-Based Retrieval, Information Universal Access and Interaction, Including Datamodels and Languages, report of the DELOS2 European Network of Excellence (FP6).
- [2] YAN, Rong; HAUPTMANN, Alexander G.; JIN, Rong: [Negative pseudo-relevance feedback in content-based video retrieval](#). ACM Multimedia 2003: 343-346.
- [3] BROWNE, Paul; SMEATON, Alan F.: [Video information retrieval using objects and ostensive relevance feedback](#). Symposium on Applied Computing (SAC) 2004: 1084-1090.
- [4] MÜLLER, Henning; MÜLLER, Wolfgang; MARCHAND-MAILLET, Stephane; PUN, Thierry; SQUIRE, David: [Strategies for Positive and Negative Relevance Feedback in Image Retrieval](#). ICPR 2000: 5043-5042.
- [5] BEZERRA, F. N.; WERBET, E.; SILVA, BEZERRA, Wendel: [Client-side content-based refinement for image search in the web](#). In: XI Simposio Brasileiro de Sistemas Multimidia e Web - WebMedia, 2005, Pocos de Caldas, MG, Brasil.. Proceedings of the WebMedia, 2005.
- [6] BUTUROVIC, Adis: MPEG 7 Color Structure Descriptor for visual information retrieval project VizIR1. Institute for Software Technology and Interactive Systems, Technical University Vienna, 2005.
- [7] DE NATALE, F.G.B. and GRANELLI, F: Structure-Based Image Retrieval Using a Structured Color Descriptor. DICA – University of Trento, 1999.
- [8] HOFFMANN, Florian: Metadaten für Multimedia im Überblick. Seminararbeit im Rahmen des Seminars „Bridging the Semantic Gap“. SS2006.
- [9] Wikipedia: „Support Vector Machine“; http://en.wikipedia.org/wiki/Support_vector_machine ; Zuletzt besucht: 12.06.2006.
- [10] ROCCHIO, J.J.: Relevance Feedback in information retrieval. In *The SMART Retrieval System, Experiments in Automatic Document Processing*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1971.
- [11] SCHUMACHER, Marcus: Ontologien. Seminararbeit im Rahmen des Seminars „Bridging the Semantic Gap“. SS2006.
- [12] Photohomepage: „Sitzbank“; http://www.photohomepage.de/galerien_infrarotfoto_sitzbank.htm

Zuletzt besucht: 12.06.2006.

- [13] Institut für Gebäude – und Solartechnik: ohne Titel;

<http://www.igs.bau.tu-bs.de/html/main.htm>

Zuletzt besucht: 12.06.2006.

- [14] Astrogilde: „Sonne“;

<http://www.astrogilde.de/astrophotos/sun/sun.htm>

Zuletzt besucht: 12.06.2006.

Max-Planck-Institut für Kernphysik: „Die Sonne – Ein Stern“;

<http://www.mpi-hd.mpg.de/nuastro/Educ/sonne/slide01.html>

Zuletzt besucht: 12.06.2006.

Institut für Gebäude – und Solartechnik: ohne Titel;

<http://www.igs.bau.tu-bs.de/html/main.htm>

Zuletzt besucht: 12.06.2006.