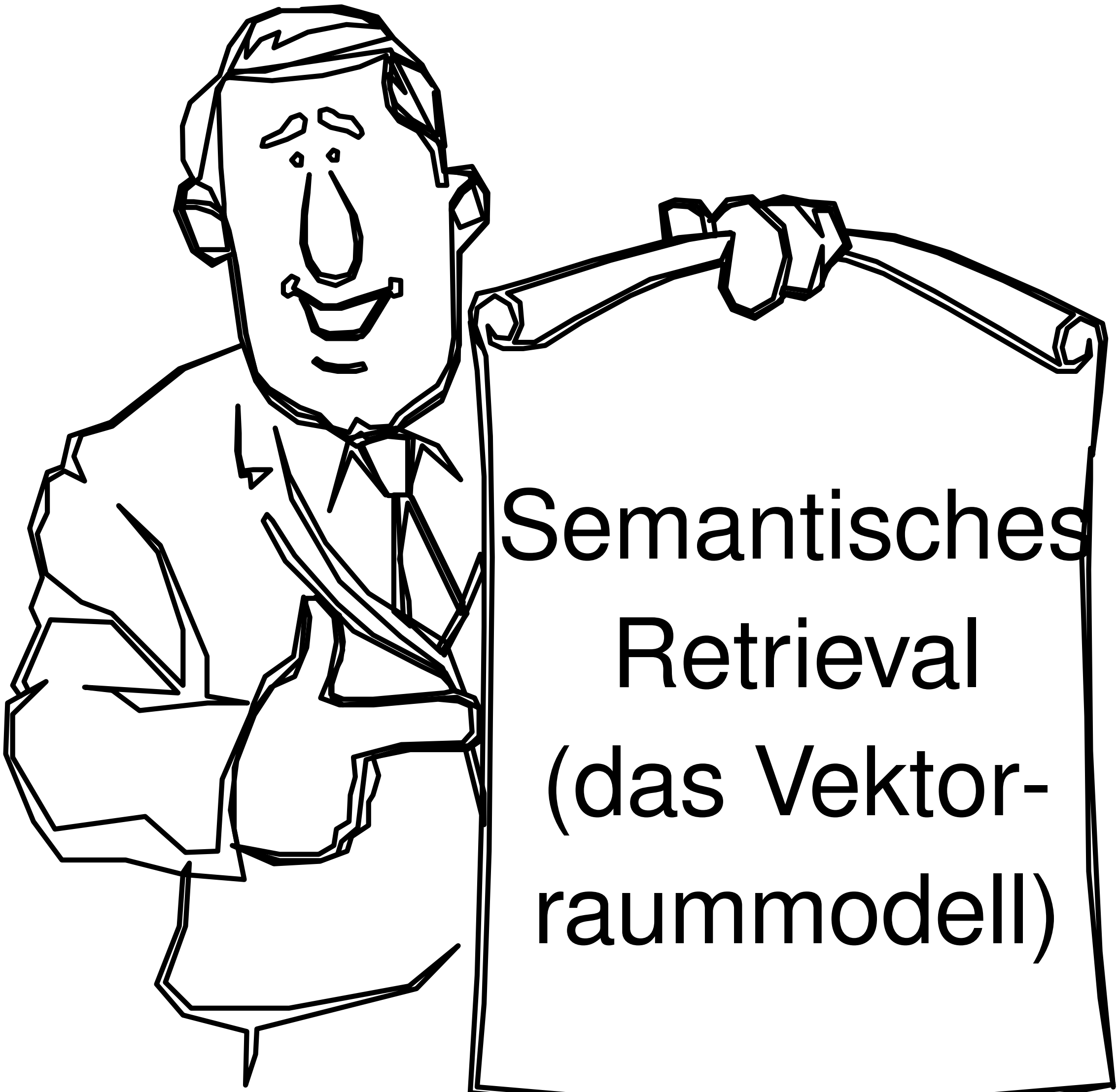

Seminar in der AG Richter
Universität Kaiserslautern

von Stephan Scholz

1. Grundzüge des Vektorraummodells

Das Vektorraummodell (VRM) wurde ursprünglich im Rahmen der Arbeiten am SMART-Projekt entwickelt. SMART ist ein experimentelles Retrieval-System, das von Gerard Salton ab 1961 erstellt wurde.
Im Vektorraummodell werden Anfragen und Dokumente einer Datenbank auf Vektoren abgebildet. Beim Retrieval wird dann nach solchen Dokumenten gesucht, deren Vektoren ähnlich (im Sinne eines vorgegebenen Maßes) zum Anfragevektor sind.

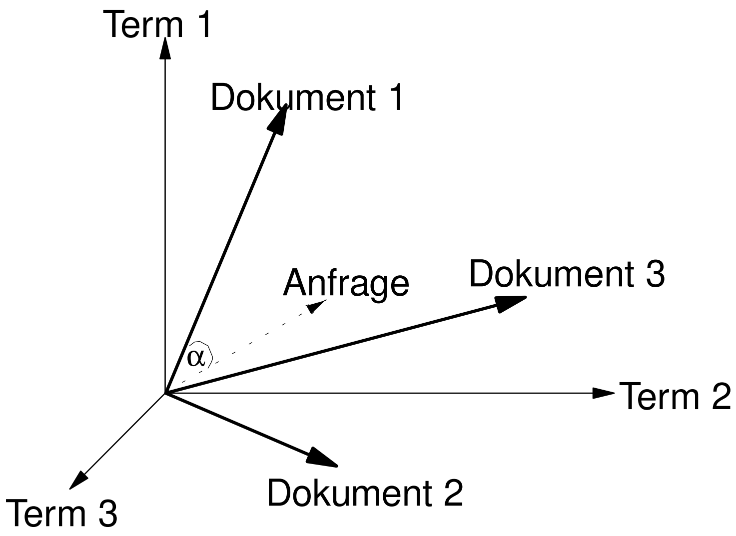

Bild 1 : Beispiel eines Vektorraums

Als Beispiel wurde hier der dreidimensionale Raum gewählt, i.a. hat der Vektorraum natürlich mehr Dimensionen.

"Ähnlichkeit" ist Interpretationssache, d.h. abhängig vom gewählten **Ähnlichkeitsmaß**. Im Laufe der Geschichte wurden unzählige verschiedene Maße verwendet, zum Beispiel :

- Kosinus des eingeschlossenen Winkels

$$\cos(\vec{p}, \vec{q}) = \frac{\sum_{i=1}^{n} p_i \cdot q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}}$$

  Hierbei ist der Wertebereich des Kosinus das Intervall [-1; 1].
  -1 bedeutet : keine Ähnlichkeit (schlecht),
  1 bedeutet : starke Ähnlichkeit (gut).

- Skalarprodukt

$$\vec{p} * \vec{q} = \sum_{i=1}^{n} p_i \cdot q_i$$

  Hier gilt : je größer der Wert, desto mehr Ähnlichkeit.

- Entropie

  Dies ist ein Maß, das in der Statistik verwendet wird.

Der Kosinus ist wohl das gängigste Ähnlichkeitsmaß, und wird auch im SMART-System verwendet. Für weitere Maße siehe [2].

Beispiel :
Eine Datenbank, die lediglich 2 Wörter (bzw. Terme) enthält : "Michael" und "Richter". Die Datenbank enthalte 2 Dokumente :

$$d_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad d_2 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

Die erste Vektorkomponente gebe hier die Anzahl der Wortes "Michael" im Dokument an, die zweite die Anzahl des Wortes "Richter".

Die Anfrage nach "Michael Richter" $q = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ ergibt mit dem Kosinusmaß :

$$\text{Ähnlichkeit}(q, d_1) = \cos(q, d_1) = 0{,}981$$
$$\text{Ähnlichkeit}(q, d_2) = \cos(q, d_2) = 0{,}707$$

$\Rightarrow$ d_1 ist also bzgl. q besser als d_2 , was ja auch intuitiv sinnvoll ist.

Indexierung einer Datenbank

Indexierung ist eine Abbildung der Dokumente einer Datenbank auf Vektoren. Hierbei **kann** ein Dokument eine Datei sein, muß aber nicht. Indexiert man beispielsweise einen EMail-Ordner, so sind viele Mails (also Dokumente) in einer Datei zusammengefaßt.
Die Auswahl der Indexwörter ist vom Zweck der Datenbank abhängig. Eine Fußballdatenbank hat sicherlich andere Schlüsselwörter als ein CD-Verzeichnis.

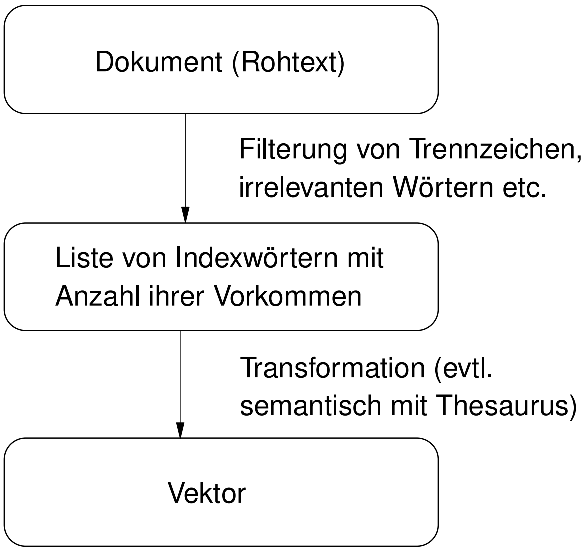

Bild 2 : Indexierungvorgang für ein Dokument

Zu Bild 2 : Irrelevante Wörter sind z.B. "und", "oder" sowie alle im Anwendungsfall unwichtigen Ausdrücke.
Jeder Vektor hat den gleichen Aufbau (Gestalt, Anzahl der Komponenten) :

Beispiel : $V = \begin{pmatrix} \#\text{"Pokal"} \\ \#\text{"Borussia"} \\ \vdots \\ \#\text{"Fußball"} \end{pmatrix}$

Die Transformation in Vektoren kann sehr unterschiedlich ausfallen. Im Beispiel gibt jede Vektorkomponente die Anzahl der Vorkommen eines bestimmten Terms an, in Kapitel 3 wird dies jedoch anders aussehen, da dort die semantischen Beziehungen der Wörter berücksichtigt werden.

2. Probleme beim klassischen Retrieval

Ziel eines Retrieval-Systems ist es, dem Benutzer möglichst einen hohen Recall sowie eine große Precision zu liefern. Hierbei bezeichnet man mit Recall die Anzahl der für den Benutzer relevanten Dokumente, die tatsächlich vom System aus der Datenbank zurückgegeben werden ("Ausbeute"). Mit Precision meint man die "Güte" des Suchergebnisses, d.h. der Anteil der relevanten Dokumente im Suchergebnis im Bezug auf dessen Umfang.
Bei klassischen, booleschen System bekommt der Benutzer auf seine Anfrage entweder zuviel oder zuwenig Information zurückgeliefert, wie die folgenden Beispiele belegen :

Beispiel :

a) zuwenig :

Sucht man mit Archie auf FTP-Servern nach Stereogrammen (Magisches Auge, 3D-Bilder), so führt lediglich die Abkürzung SIRDS (für Single Image Random Dot Stereogram) zum Erfolg. Versuche, das gewünschte Ergebnis mit Wörtern wie "magic", "3D", "stereo" etc. zu bekommen, sind zum Scheitern verurteilt.

b) zuviel :

Viele Wörter haben Doppelbedeutungen, so daß die Suche nach "Maus" nicht nur Informationen über Computermäuse zurückliefert, sondern evtl. auch über biologische Mäuse...
Weitere Doppelbedeutungen : "Bill", "Killer", "Platte".

Ursache 1 : Semantische / Syntaktische Ebene

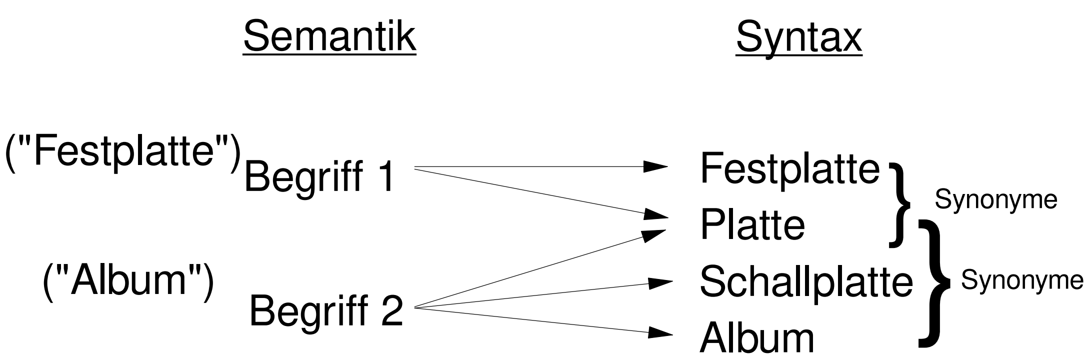

Bild 3 : Beispiel für semantische Beziehungen

Ein **Schlüsselwort** (Syntaktische Ebene) ist eine Buchstabenkette, die einen Begriff repräsentiert.
Ein **Begriff** (Semantische Ebene) ist eine Interpretation eines Schlüsselwortes.
Ein Begriff kann nun mehrere Schlüsselwörter haben ("**Synonyme**"); und ein Schlüsselwort kann mehrere Bedeutungen haben ("**Homonym**", siehe Bild 3 : "Platte").
Nun verwendet der Indexierer einer Datenbank evtl. andere Schlüsselwörter als der Benutzer für den gleichen Begriff. Die Folgen davon sind die oben genannten Probleme.
Die Hauptarbeit in booleschen Systemen besteht demnach darin, die richtige Anfrage zu konstruieren, die von der syntaktischen Umsetzung her den Schlüsselwörtern des Indexierers entspricht.

Hier eine andere Darstellung des Beispiels :

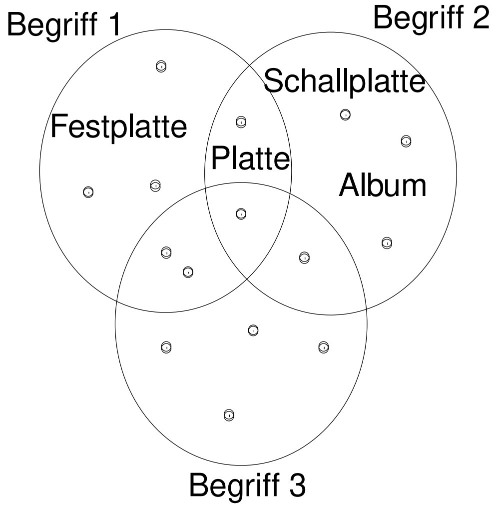

Bild 4 : Beispiel für semantische Beziehungen, graphisch

Dieses Bild ist in der sogenannten **extensionalen Darstellung**. In dieser Form ist ein Begriff durch die Aufzählung seiner Schlüsselwörter gegeben. Man sieht hier sehr gut, daß der Term "Platte" zu zwei verschiedenen Begriffen gehört.
Die **intensionale Darstellung** ist eine verbale Beschreibung des Begriffs, also die Idee die sich hinter ihm verbirgt.
Beim booleschen Retrieval wird die Bedeutung der Schlüsselwörter nicht berücksichtigt. Alle Terme werden als unabhängig angesehen. Dies ist natürlich eine enorme Einschränkung !

Ursache 2 : Fehlendes Ranking

Eine **Retrieval-Status-Wert-Funktion** ist eine Abbildung von Dokumenten auf reelle Zahlenwerte : jedem Dokument wird eine geschätzte Relevanz zugeordnet, woraus sich eine Rangfolge der Dokumente (Ranking) ergibt. Im Idealfall hat das Dokument mit dem höchsten Retrieval-Status-Wert (RSW) auch die größte Relevanz für den Benutzer. Berechnet wird der RSW mit Hilfe des Ähnlichkeitsmaßes des Vektorraummodells. Der Vorteil des Ranking ist der, daß der Benutzer eine vorsortierte Ergebnisliste bekommt. Er kann davon ausgehen, daß die obersten Dokumente diejenigen sind, die seiner Anfrage am nächsten kommen. WAIS (Wide Area Information System) arbeitet mit dem Vektorraummodell, demnach beherrscht WAIS Ranking, wenn auch keine Semantik.
Boolesche IR-Systeme erkennen leider nicht die Relevanz von einzelnen Dokumenten, sondern arbeiten nach dem Prinzip "Dokument erfüllt / erfüllt nicht die Anfrage". Man bekommt also u.U. eine riesige Antwortmenge und muß nun selber einzeln überprüfen, welche Dokumente tatsächlich relevant sind. Oder andersherum : man bekommt zuwenig Ergebnisse.

Beispiel :

Die Anfrage "Information AND Retrieval" liefert in booleschen Systemen alle Dokumente, die beide Wörter enthalten, nicht jedoch solche, die alleine "Retrieval" enthalten.
Beim Ranking hingegen bekommt der Benutzer eine sortierte Liste, bei der Dokumente an oberster Stelle stehen, die beide Wörter enthalten, zusätzlich werden aber auch die Dokumente aufgeführt, die nur eins der beiden Wörter enthalten (mit niedriger Priorität).

3. Informationsstrukturen

Wie lassen sich nun die semantischen Beziehungen von Begriffen formal beschreiben ? Hierzu bedient man sich eines **Thesaurus**.

Beispiel :

Gegeben sei ein Thesaurus mit folgenden drei Relationen :

$(a,b) \in NT$	gdw	b ist Unterbegriff von a
$(a,b) \in RT$	gdw	b ist verwandt mit a
$(a,b) \in SY$	gdw	b ist Synonym von a

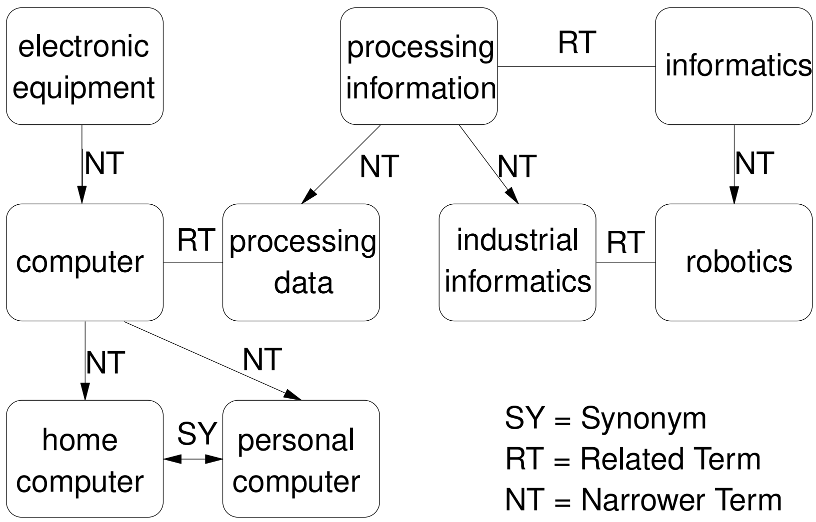

Bild 5 : Beispiel eines Thesaurus

Die Begriffe lassen sich nun aufspalten in elementare, semantische Einheiten, die wir **atomare Begriffe** nennen.

Beispiel :

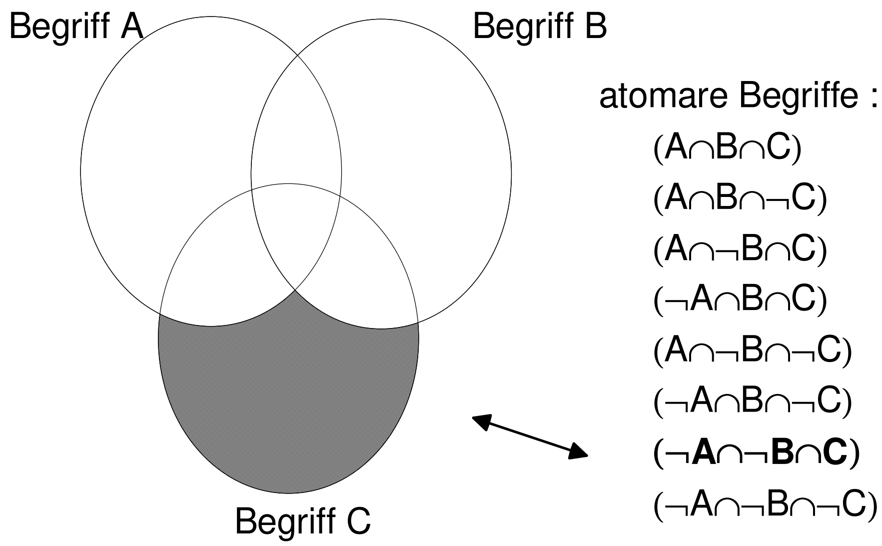

Bild 6 : Aufteilung in atomare Begriffe

Algorithmus zur Herleitung der atomaren Begriffe :

Man bedient sich zweier Tabellen : die ACT ordnet den atomaren Begriffen Terme zu, und die TAC gibt umgekehrt an, welche atomaren Begriffe zu den einzelnen Termen gehören. Diese Tabellen werden nun in mehreren Schritten hergeleitet :

1. Schritt :
   - Jeder Term bekommt einen Eintrag in der TAC und einen eindeutigen Term-Code. Zusätzlich wird in der "Top"-Spalte angegeben, ob es sich der Term auf der höchsten Ebene im Thesaurus befindet.
   - Der Thesaurus wird zeilenweise durchlaufen : für jeden Term wird ein Eintrag in die ACT eingefügt (mit eindeutigem Code), wobei Synonyme nur einmal gezählt werden. Die zugehörigen Terme werden in die TCodes-Spalte aufgenommen. Jeder dieser atomaren Begriffe bekommt den Typ "single".
   - Die Information der ACT wird in die TAC kopiert : Für jeden Term wird in die ACodes-Spalte eingefügt, in welchem atomaren Begriff er enthalten ist.

2. Schritt :
   - Für jede RT-Beziehung wird ein neuer Eintrag in die ACT eingefügt. Diese neuen atomaren Begriffe haben den Typ "double" und bekommen jeweils die beiden zugehörigen Terme der RT-Beziehung in die TCodes-Spalte geschrieben.
   - Wie in Schritt 1 wird die neue Information in die TAC kopiert.

3. Schritt :
   - Für jeden Unterterm werden in der ACT die zugehörigen Oberterme ergänzt (Spalte Tcodes). Hierbei werden Pfade rekursiv durchlaufen, d.h. Unterpfade zusätzlich abgearbeitet.
   - Wiederum wird die neue Information in die TAC kopiert.

4. Schritt :
   - Ein letzter atomarer Begriff wird in die ACT eingefügt, der völlig unabhängig von den gegebenen Termen ist (siehe Bild 7 : $\alpha 12$).

Mit dem Beispiel ergibt sich durch den Algorithmus :

ACT
(Atomarer Begriff → Term)

ACode	Typ	TCodes
$\alpha 1$	single	3
$\alpha 2$	single	6
$\alpha 3$	single	8
$\alpha 4$	single	2,3
$\alpha 5$	single	1,6
$\alpha 6$	single	7,6
$\alpha 7$	single	9,8
$\alpha 8$	single	4,5,2,3
$\alpha 9$	double	6,8
$\alpha 10$	double	1,2,6,3
$\alpha 11$	double	7,9,6,8
$\alpha 12$		

TAC
(Term → Atomarer Begriff)

Term	TCode	Top	ACodes
processing data (D)	1	nein	5,10
computer (C)	2	nein	4,10,8
electronic equipment (E)	3	ja	1,4,8,10
home computer (HC)	4	nein	8
personal computer (PC)	5	nein	8
processing information (P)	6	ja	2,9,5,6,10,11
industrial informatics (II)	7	nein	6,11
informatics (I)	8	ja	3,9,7,11
robotics (R)	9	nein	7,11

1. Schritt : SY - Relationen
2. Schritt : RT - Relationen
3. Schritt : NT - Relationen

Die 9 Begriffe sind also in 12 atomare Begriffe zerlegt worden.

Graphische Darstellung :

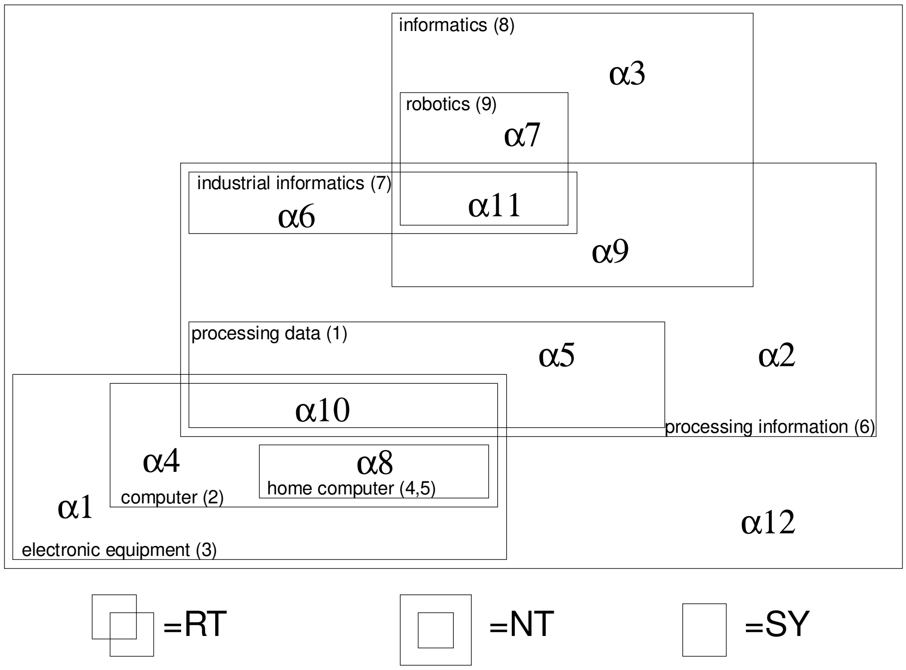

Bild 7 : Graphische Darstellung des Thesaurus

Die atomaren Begriffe lassen sich durch boolesche Formeln ausdrücken :
$\alpha_i = k_1 \mathrm{P} \cap k_2 \mathrm{II} \cap k_3 \mathrm{E} \cap k_4 \mathrm{I} \cap k_5 \mathrm{C} \cap k_6 \mathrm{D} \cap k_7 \mathrm{R} \cap k_8 \mathrm{H}$
mit $k_j \equiv \neg$ für Terme, die nicht in der Zeile von α_i in der ACT enthalten sind (k_j leer sonst).
<u>Beispiel :</u> $\alpha_6 = \mathrm{P} \cap \mathrm{II} \cap \mathrm{E} \cap \mathrm{I} \cap \mathrm{C} \cap \mathrm{D} \cap \mathrm{R} \cap \mathrm{H}$
Bemerkung : "Home computer" (HC) und "Personal computer" (PC) sind mit dem Symbol H zusammengefaßt, da es sich um Synonyme handelt.

Hier eine komplette Auflistung aller zugehörigen booleschen Formeln :

Atomarer Begriff	Boolesche Formel
$\alpha 1$	$E\cap\neg P\cap\neg I\cap\neg C\cap\neg D\cap\neg II\cap\neg R\cap\neg H$
$\alpha 2$	$\neg E\cap P\cap\neg I\cap\neg C\cap\neg D\cap\neg II\cap\neg R\cap\neg H$
$\alpha 3$	$\neg E\cap\neg P\cap I\cap\neg C\cap\neg D\cap\neg II\cap\neg R\cap\neg H$
$\alpha 4$	$E\cap\neg P\cap\neg I\cap C\cap\neg D\cap\neg II\cap\neg R\cap\neg H$
$\alpha 5$	$\neg E\cap P\cap\neg I\cap\neg C\cap D\cap\neg II\cap\neg R\cap\neg H$
$\alpha 6$	$\neg E\cap P\cap\neg I\cap\neg C\cap\neg D\cap II\cap\neg R\cap\neg H$
$\alpha 7$	$\neg E\cap\neg P\cap I\cap\neg C\cap\neg D\cap\neg II\cap R\cap\neg H$
$\alpha 8$	$E\cap\neg P\cap\neg I\cap C\cap\neg D\cap\neg II\cap\neg R\cap H$
$\alpha 9$	$\neg E\cap P\cap I\cap\neg C\cap\neg D\cap\neg II\cap\neg R\cap\neg H$
$\alpha 10$	$E\cap P\cap\neg I\cap C\cap D\cap\neg II\cap\neg R\cap\neg H$
$\alpha 11$	$\neg E\cap P\cap I\cap\neg C\cap\neg D\cap II\cap R\cap\neg H$
$\alpha 12$	$\neg E\cap\neg P\cap\neg I\cap\neg C\cap\neg D\cap\neg II\cap\neg R\cap\neg H$

4. Durchführung einer Anfrage

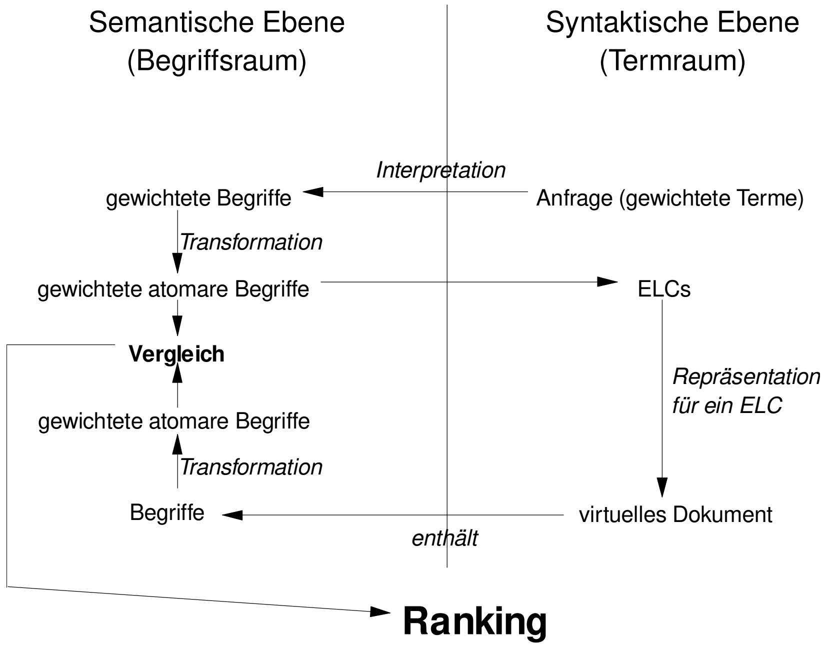

Bild 8 : Abarbeitung einer Anfrage

Es findet ein sogenanntes **Pre-Ranking** statt, d.h. bevor die eigentlichen Dokumente aus der Datenbank verglichenwerden, teilt man sie (durch ELCs[1]) in verschiedene Klassen ein. Dokumente der gleichen ELC-Klasse besitzen die gleiche Relevanz.

Bei der Transformation der Begriffe in atomare Begriffe müssen diese auch eine Gewichtung erhalten. Dazu gibt es mehrere Möglichkeiten :

- Die Methode nach Bärtschi : jeder atomare Begriff bekommt das gleiche Gewicht zugeteilt, nämlich $w(\alpha_i) = 1$ für alle i. Hierbei geht die ursprüngliche Gewichtung der Begriffe verloren.
- H.P. Giger schlägt folgende Gewichtung vor :
  $w(\alpha_i) = \max\{w_j(\alpha_i)\}$ mit
  $$w_j(\alpha_i) = \frac{w(C_j)}{\left|\{\alpha_k \text{für die gilt : } \alpha_k \in C_j\}\right|}$$
  wobei C_j ein Begriff (Term aus der TAC) ist, und $\alpha_k \in C_j$ bedeutet : C_j kommt in der ACT bei α_k vor.

<u>Beispiel :</u> Suche nach

- Robotics 0,5
- Industrial informatics 1,0
- Processing data 0,4

$\}$ Anfrage mit Gewichten

[1] siehe Anhang

Die Gewichtung der Terme der Anfrage ergibt folgende Gewichtung der atomaren Begriffe (nach H.P. Giger) :

atomarer Begriff	Gewicht
$\alpha 1$	0
$\alpha 2$	0
$\alpha 3$	0
$\alpha 4$	0
$\alpha 5$	0,2
$\alpha 6$	0,5
$\alpha 7$	0,25
$\alpha 8$	0
$\alpha 9$	0
$\alpha 10$	0,2
$\alpha 11$	0,5
$\alpha 12$	0

Transformation in Vektoren :

	$\alpha 1$	$\alpha 2$	$\alpha 3$	$\alpha 4$	$\alpha 5$	$\alpha 6$	$\alpha 7$	$\alpha 8$	$\alpha 9$	$\alpha 10$	$\alpha 11$	$\alpha 12$
Anfrage	(0,	0,	0,	0,	1,	1,	1,	0,	0,	1,	$1+\lambda$,	0)
ELC1 {R,II,D}	(0,	0,	0,	0,	1,	1,	1,	0,	0,	1,	$1+\lambda$,	0)
ELC2 {R,II}	(0,	0,	0,	0,	0,	1,	1,	0,	0,	0,	$1+\lambda$,	0)
ELC3 {R,D}	(0,	0,	0,	0,	1,	0,	1,	0,	0,	1,	1,	0)
ELC4 {II,D}	(0,	0,	0,	0,	1,	1,	0,	0,	0,	1,	1,	0)
ELC5 {R}	(0,	0,	0,	0,	0,	0,	1,	0,	0,	0,	1,	0)
ELC6 {II}	(0,	0,	0,	0,	0,	1,	0,	0,	0,	0,	1,	0)
ELC7 {D}	(0,	0,	0,	0,	1,	0,	0,	0,	0,	1,	0,	0)

Es ergeben sich 7 Dokumentklassen (virtuelle Dokumente), wobei ELC1 der Anfrage entspricht. Die Vektorkomponenten beziehen sich nicht mehr auf die Begriffe, sondern auf atomare Begriffe.

Der λ-Faktor bestimmt die Wichtigkeit von stärkerer Präsenz eines atomaren Begriffs im Dokument bzw. der Anfrage. So enthält $\alpha 11$ beispielsweise nach der ACT die Begriffe R und II, somit wird für ELC1 und ELC2 $\alpha 11$ höher bewertet. Die Wahl des λ hängt ab vom Anwendungsfall, also der Datenbank auf der operiert wird. Bezüglich dieses Faktors ist noch viel Forschungsarbeit nötig...hier wird er einfach auf $\lambda=0,5$ gesetzt.

Resultierendes Ranking der virtuellen Dokumente :

<u>Kosinusmaß</u>	<u>Entropie</u>
ELC1 (1,0)	ELC1 (1,0)
ELC4 (0,949)	ELC4 (0,924)
ELC2 (0,919)	ELC2 (0,885)
ELC6 (0,868)	ELC3 (0,847)
ELC3 (0,854)	ELC6 (0,797)
ELC5 (0,769)	ELC5 (0,705)
ELC7 (0,393)	ELC7 (0,403)

ELC1 wird erwartungsgemäß mit beiden Maßen als am besten bewertet, da dies der Anfrage genau entspricht. Die Stärke des Verfahrens (semantischer Vergleich) läßt sich gut daran ablesen, daß ELC4 besser bewertet wird als ELC2. Schaut man sich die Thesaurus-Struktur an (graphische Darstellung), so sieht man, daß sich der Term "Industrial informatics" mit "Robotics" überschneidet, während er unabhängig ist von "Processing data". Somit deckt ELC4 mehr Gebiet der Anfrage ab, und es macht Sinn, ihn höher zu ranken als ELC2.
Klassische Verfahren hätten beiden ELCs die gleiche Relevanz zugeordnet oder sogar ELC2 höher bewertet als ELC4 (je nach Gewichtung der Terme).

<u>Erweiterung des Algorithmus</u>

Da sich (wie gesehen) die semantischen Begriffe überschneiden, lassen sich naheliegende Terme in die ursprüngliche Anfrage zusätzlich aufnehmen. Manche Begriffe sind nämlich automatisch in der Anfrage enthalten, auch wenn ihre Schlüsselwörter nicht auftauchen.

<u>Beispiel :</u>

Originalanfrage : {R,II,D}
Erweiterte Anfrage : {R,II,D,P,I,C}
Die Begriffe "Processing information" (P), "Informatics" (I) und "Computer" (C) hängen zusammen mit den Begriffen der ursprünglichen Anfrage.

<u>! Vorsicht !</u> : Die Anzahl der resultierenden Dokumentenklassen (ELCs) kann auf diese Art sehr groß werden und das System drastisch verlangsamen ! Aus Sicherheitsgründen überläßt man die Auswahl dieser zusätzlichen Begriffe daher wohl besser dem Benutzer, es ist jedoch auch denkbar, daß das System selbständig eine erweiterte Anfrage konstruiert. Basis für solch eine selbständige Erweiterung könnten dann die für den Benutzer relevanten Dokumente sein (das muß dieser natürlich dann nach Auswertung der Originalanfrage angeben).

Weitere Verbesserungen sind nun denkbar, indem man phonetische Algorithmen zusätzlich benutzt. Falls nämlich ein Wort nicht in der Datenbank gefunden wird, kann man evtl. durch Rechtschreibkorrektur und Phonetik zu dem gesuchten Begriff gelangen. Diese Erweiterungen setzen dann auf dem gegebenen System auf.

5. Zusammenfassung

- Klassische, boolesche Information Retrieval-Systeme sind rein zeichenorientiert und berücksichtigen nicht die Bedeutung von Wörtern. Terme werden hier als unabhängig angesehen.
- Die hier vorgestellte Methode teilt Begriffe in elementare, semantische Einheiten, die atomare Begriffe genannt werden. Die hierzu benötigte Information läßt sich aus einem Thesaurus herleiten.
- Vorteil : Dokumente werden gefunden, bei denen klassische Methoden bisher versagten.
- Darüberhinaus wird mit dem Vektorraummodell Ranking benutzt, so daß der Benutzer nicht durch Informationsfülle erschlagen wird. Eine nach Relevanz geordnete Liste von Dokumenten wird an den (erfreuten !) Benutzer zurückgeliefert.
- Es gibt tatsächlich Systeme, die den Begriffsraum und das VRM benutzen, z.B. das System CB-AIR (Concept Based Advanced Information Retrieval) von Radio Switzerland. Weitere Informationen hierzu findet man in [1].

Anhang : Der Begriff "ELC"

Jede boolesche Formel läßt sich darstellen in ihrer Disjunktiven Normalform (DNF), die eine Disjunktion von Konjunktionen ist.

Beispiel :

(t_1 OR t_2) AND (t_3 OR t_2) AND t_4	$\Leftrightarrow$	(t_1 AND t_2 AND t_3 AND t_4)	OR
		(t_1 AND t_2 AND NOT t_3 AND t_4)	OR
		(t_1 AND NOT t_2 AND t_3 AND t_4)	OR
		(NOT t_1 AND t_2 AND t_3 AND t_4)	OR
		(NOT t_1 AND t_2 AND NOT t_3 AND t_4)	

$\uparrow$

ELCs

Die Konjunktionen heißen hierbei Elementare Logische Konjunktionen (ELC, Elementary logical conjuncts).

Sind also n Terme t_1, t_2, ..., t_n gegeben, so ist ein ELC eine Formel $\bigcap_{i=1}^{n} (\mu_i t_i)$, wobei μ_i entweder die Identität oder der NOT-Operator ist.

Die Menge der Dokumente, die man bekommt wenn ein ELC als Anfrage an eine Datenbank benutzt wird, bezeichnet man als das Resultat des ELCs. Die Resultate verschiedener ELCs einer DNF sind disjunkt.

Literatur

[1] H.P.Giger : Concept Based Retrieval in Classical IR Systems
Swiss Federal Institute of Technology (ETH), Zurich, Switzerland

[2] Z.W.Wang, S.K.M.Wong, Y.Y.Yao : An Analysis of Vector Space Models Based on Computational Geometry
ACM Conference Proceedings Series
SIGIR'92 : 15th International Conference on Information Retrieval
Signatur INF 535/031

[3] G.Salton, M.J.McGill : Introduction to Modern Information Retrieval
McGraw-Hill Book Company
Signatur INF 535/034

[4] U. Pfeifer : HTTPs älterer Bruder (WAIS : Inhaltsorientierte Suche im Internet)
iX 1/95, S. 120-127
Heise-Verlag

[5] M.Schmidt : Nach Maß (WAIS : Text-Retrieval mit Struktur)
iX 4/95, S. 172-182
Heise-Verlag

[6] M.Bärtschi : Term Dependencies in Information Retrieval Models
D.Sc.Thesis ETH, No. 7525, ETH Zurich, 1984

[7] G. Salton : A Blueprint for Automatic Boolean Query Processing
SIGIR Forum, ACM. 17(2):6-24; 1982

[8] G.Salton, E.Fox : Extended Boolean Information Retrieval
Communication of the ACM, 26(11): 1022-1036; 1983

[9] P.Schäuble : Thesaurus Based Concept Spaces
Proc. of the 10th ACMSIGIR Conf. on R&D in Information Retrieval,
ACM, New Orleans: 254-262; 1987

[10] S. Wong : Generalized Vector Space Model for Boolean Query Processing
Proc. of the 8th ACMSIGIR Conf. on R&D in Information Retrieval,
ACM, 18-25; 1985

[11] S. Wong : Generalized Vector Space Model for Boolean Query Processing
Proc. of the 9th ACMSIGIR Conf. on R&D in Information Retrieval,
ACM, Pisa, 175-185; 1986