Information Retrieval und Question Answering
Universität Trier LDV/CL
WS 2009/2010 HS Dialogsysteme
Kai Kugler (kugl2203@uni-trier.de)

1 Information Retrieval

Information Retrieval (IR) ist das maschinelle, inhaltsorientierte Auffinden von relevantem Wissen in meist ungeordneten, großen Datenbeständen (z.B. Textsammlungen, Textkorpora). Die Relevanz der aufzufindenden Information ist abhängig vom aktuellen Wissen des Benutzers, dem aktuellen Problem des Benutzers und der subjektiven Erwartung des Benutzers. Im Gegensatz zu z.B. einer Anfrage an eine Datenbank ist die Frage nicht präzise und formal und das System hat keine oder nur wenig Kenntnisse über den Inhalt der Dokumente (des Datenbestandes). Insbesondere Internetsuchmaschinen verwenden Methoden des IR, um Textdokumente zu indexieren und dem Benutzer auf seine Anfrage hin rangiert zu präsentieren.

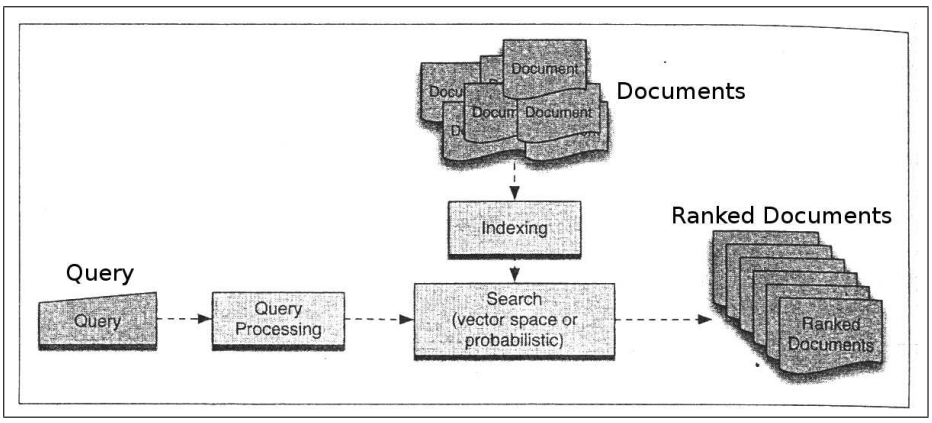

Abbildung 1: IR-System [1] (S. 802)

1.1 Vektorraummodelle

Um Dokumente in einem IR-System zu repräsentieren, werden diese als Vektor der Gewichtung ihrer Merkmale modelliert. Ein einfaches gewichtetes Merkmal eines Dokuments ist z.B. die Häufigkeit der darin vorkommenden Terme (Wörter). **Beispiel**: In einem Zeitungsartikel haben die Terme (Naumann, Kulturbeauftragte, parlamentarisch, Staatssekretär) die Häufigkeiten (11,14,12,9). Dieses Dokument (1) kann also nach Termhäufigkeit dieser vier Terme als Vektor $\vec{d_1} = (11, 14, 12, 9)$ dargestellt werden. Ein anderer Text (2) unseres Korpus habe dagegen die Termhäufigkeiten $\vec{d_2} = (8, 0, 0, 0)$. Das Textkorpus kann entsprechend als Matrix dieser Vektoren dargestellt werden:

$$A = \begin{pmatrix} 11 & 8 \\ 14 & 0 \\ 12 & 0 \\ 9 & 0 \end{pmatrix} \tag{1}$$

Allgemein kann ein Dokument j (aus einem Textkorpus mit n Termen) als Vektor seiner Termgewichte (w) dargestellt werden als:

$$\vec{d_j} = (w_{1,j}, \cdots, w_{n,j}) \tag{2}$$

Ebenso kann eine Anfrage auch als Vektor ihrer Terme dargestellt werden. Eine Anfrage nach (Naumann,Kulturbeauftragte) in unserem Beispiel z.B. als $\vec{q} = (1, 1, 0, 0)$,

oder allgemein:

$$\vec{q} = (w_{1,q}, \cdots, w_{n,q}) \tag{3}$$

Um die Anfrage $\vec{q}$ mit den beiden Dokumenten $\vec{d_1}$ und $\vec{d_2}$ zu vergleichen, wird jeweils der Cosinus der Winkel ($\in [0,1]$) zwischen den Vektoren berechnet. Dieser ergibt bei identischen Vektoren 0, bei orthogonalen 1.

$$sim(\vec{q}, \vec{d_j}) = \frac{\sum_{i=1}^{N} w_{i,q} \times w_{i,j}}{\sqrt{\sum_{i=1}^{N} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,j}^2}} \tag{4}$$

1.2 Termgewichtung

Die Gewichtung der Terme hat einen großen Einfluß auf die Güte des IR-Systems. Im obigen Beispiel wurden die Terme mit der einfachen Häufigkeit gewichtet. Diese ist u.a. abhängig von der Länge der Texte. Zur Indexierung wird deshalb eine andere Gewichtung herangezogen.
Tf-idf (term frequency - inverse document frequency) ist das Produkt aus Termhäufigkeit (tf) und inverser Dokumentenhäufigkeit (idf):

$$w_{i,j} = tf_{i,j} \cdot idf_i = \frac{freq_{i,j}}{max_l(freq_{l,j})} \cdot \log \frac{N}{n_i} \tag{5}$$

($w_{i,j}$ Gewicht des Terms i im Dokument j, $freq_{i,j}$ Häufigkeit des Terms i in Dokument j, $max_l(freq_{l,j})$ Maximalhäufigkeit aller Terme im Dokument, N Anzahl der Dokumente im Korpus (vgl. [2]), n_i Anzahl der Dokumente, in denen Term i vorkommt)
Diese Termgewichtung kann beim Vergleich (Cosinus) der Vektoren benutzt werden:

$$sim(\vec{q}, \vec{d}) = \frac{\sum\limits_{w \in q,d} tf_{w,q}\, tf_{w,d}\, (idf_w)^2}{\sqrt{\sum\limits_{q_i \in q} (tf_{q_i,q} idf_{q_i})^2} \times \sqrt{\sum\limits_{d_i \in d} (tf_{d_i,d} idf_{d_i})^2}} \tag{6}$$

1.3 Termauswahl / Probleme

Bislang haben wir als Terme die in den Texten auftretenden Wortformen betrachtet. Der Beispieltext über den Kulturbeauftragten Naumann führte zu dem Vektor (Naumann, Kulturbeauftragte, ...), da im Text "der Kulturbeauftragte"vorkommt. Eine Anfrage $\vec{q} = (Naumann, Kulturbeauftragter)$ würde so zu einem unbefriedigenden Ergebnis führen.

Stemming

Abhilfe könnte in solchen Fällen eine Grundformenreduktion (Stemming) der Wörter im Text und in der Suchanfrage schaffen. Ein einfaches automatisches Stemming kann jedoch auch zu unerwünschten Ergebnissen führen: Eine Reduktion des Anfrageterms von (Katze) auf (Katz) in der Suche könnte z.B. als Ergebnis Texte über den Comiczeichner Stephan Katz liefern.

Stoppwortliste

Oft kommen auch Stoppwortlisten zum Einsatz. Hochfrequente Wörter von gerignem semantischem Gehalt (..., auch, auf, aus, bei, bin, bis, ...) werden nicht indexiert und in der Anfrage nicht als Terme gewertet. Dies führt jedoch u.a. zu Problemen bei der Phrasensuche. Die Phrase *to be or not to be* würde nach Entfernung der Stoppwörter z.B. auf *not* reduziert.

Synonyme, Polyseme, Homonyme

Beschränkt sich die Termauswahl nur auf die in den Texten vorkommenden Wörter, führt dies natürlich dazu, daß bei einer Anfrage nach *Orange* Texte, in denen *Apfelsine* vorkommt, evtl. nicht gefunden werden. Beide Terme werden zudem einzeln indexiert und gewichtet. Sucht man hingegen nach *Bank*, liefert das System möglicherweise Texte zu Kreditinstituten, Sitzgelegenheiten und Ablagerungen von Sedimentgestein.

1.4 Evaluation von IR-Systemen

Um IR-Systeme zu vergleichen werden die Dokumente (bzgl. einer Anfrage) in relevante und irrelevante Dokumente eingeteilt und überprüft, ob das System in der Lage ist, die richtigen, also relevanten Dokumente auszuwählen. Hierzu werden anhand der vom System als relevant eingestuften (*retrieved*) Dokumente die *precision* und der *recall* berechnet:

$$\begin{aligned} precision &= \frac{|retrieved \cap relevant|}{|retrieved|} \\ recall &= \frac{|retrieved \cap relevant|}{|relevant|} \end{aligned} \tag{7}$$

Beispiel: Das Korpus umfasst 6 Dokumente, davon sind 4 relevant (*relevant*) und 2 irrelevant. Das IR-System liefert für eine bestimmte Anfrage 3 Dokumente (*retrieved*), von denen 2 ($|retrieved \cap relevant|$) korrekt als relevant und eines fälschlicher Weise als relevant eingestuft wurde. Somit ergeben sich eine precision von $\frac{2}{3} = 0.\bar{6}$ und ein recall von $\frac{2}{4} = 0.5$.

Da z.B. Suchmaschinen ihre Ergebnisse rangiert ausgeben, wird dieser Rang in die Berechnung mit einbezogen. Der Vergleich verschiedener Systeme erfolgt meist durch Interpolation der *precision* bei festen Abstufungen des *recall*. Trägt man *precision* und *recall* verschiedener Systeme in einem Graphen auf, sieht man leicht, welches System bei gleichem *recall* die bessere *precision* liefert.

1.5 Verbesserung der Ergebnisse

Für eine Verbesserung der Suchergebnisse, die nicht allein durch eine andere Gewichtung der Terme erreicht wird, kann die Anfrage des Benutzers verändert werden. Dies erreicht man entweder, indem man den Benutzer die Suchergebnisse bewerten lässt, und/oder, indem man die ursprüngliche Anfrage erweitert.

1.5.1 Bewertung durch den Benutzer

Bewertet der Benutzer auf Anfrage die ihm gelieferten Dokumente (relevant/irrelevant), wird der neue Anfragevektor entsprechend in Richtung der relevanten Dokumente (weg von den irrelevanten) verschoben.
Sei $\vec{q_i}$ der ursprüngliche Anfragevektor, $\vec{r}$ (relevant) und $\vec{s}$ (irrelevant) Dokumente, R die Anzahl der relevanten Dokumente und S die der irrelevanten, $\beta, \gamma \in [0, 1]$. Der gedrehte Anfragevektor geht dann aus dem ursprünglichen hervor:

$$\vec{q}_{i+1} = \vec{q}_i + \frac{\beta}{R} \sum_{j=1}^{R} \vec{r}_j - \frac{\gamma}{S} \sum_{k=1}^{S} \vec{s}_k \tag{8}$$

Wobei β angibt, wie weit der Anfragevektor in die Richtung relevanter Dokumente verschoben wird und γ, wie stark er von irrelevanten Dokumenten weggeschoben wird. Diese Parameter werden experimentell bestimmt.

1.5.2 Erweiterung des Anfragevektors

Neben dem Verschieben des Anfragevektors kann dieser auch mit bestimmten Termen erweitert werden. Hierbei kann es sich um Terme handeln, die...

- aus den vom dem Benutzer als relevant bewerteten Dokumenten hervorgehen
- aus dem Umfeld (z.B. Textfenster) der Anfrageterme stammen (passage retrieval)
- aus einem externen Thesaurus stammen (Synonyme, Hyponyme, Hyperonyme...) (z.B. aus WordNet)
- den Anfragetermen (z.B. korpusspezifisch) 'ähnlich' sind (Termclustering, signifikante Kookkurrenzen) = Erstellen eines eigenen Thesaurus

2 Factoid Question Answering

Factoid Question Answering befasst sich mit dem Verarbeiten von Fragen, die als Antwort einen einfachen Fakt (Eigenname, Zahlenwert etc.) erwarten. Die Fragen werden beantwortet, indem mit Methoden des IR kurze Textsegmente aus Korpora extrahiert und diese dem Benutzer präsentiert werden. Der Prozeß des QA lässt sich grob in drei Phasen einteilen: Question Processing, passage retrieval und answer processing.

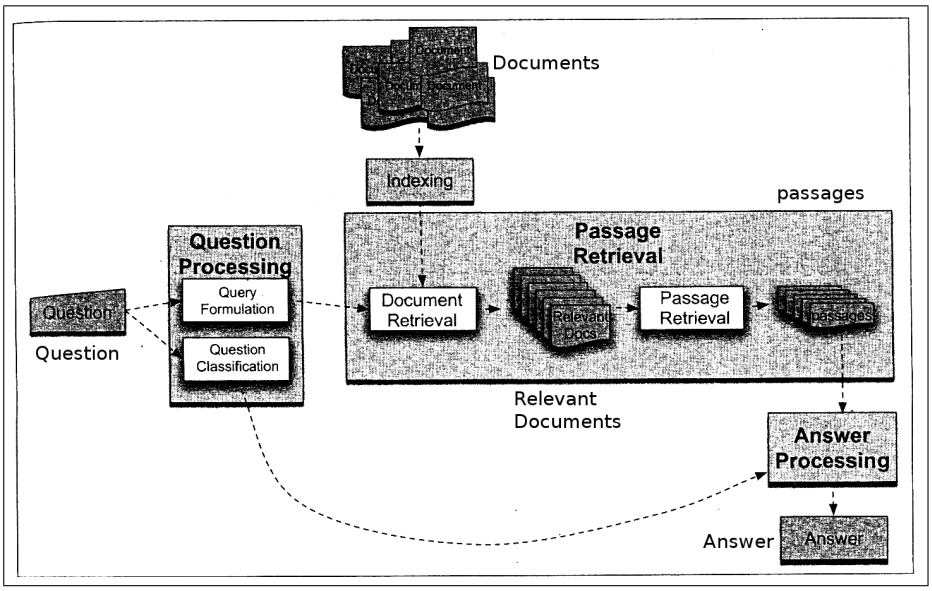

Abbildung 2: question processing, passage retrieval, answer processing [1] (S. 813)

2.1 Question Processing

Die Vorverarbeitung der Anfrage hat zum einen das Ziel, diese in eine Form bringen (query formulation), die beim späteren IR möglichst erfolgsversprechende Textpassagen auffindet, zum anderen soll ermittelt werden, welcher Antworttyp auf die Frage erwartet wird (question classification), wonach also gefragt wird.

Query formulation

Um auch in kleineren Korpora Textpassagen zu finden, die eine mögliche Antwort auf die Frage enthalten, lohnt es sich, die Anfrage mit z.B. morphologischen Varianten und Daten aus Thesauri (Synonyme etc.) zu erweitern. Eine weitere Methode ist das regelbasierte Umformulieren des Fragestrings in Einheiten, die in einer entsprechenden Antwort vorkommen könnten.

$$\begin{aligned} \textit{Where is A} &\rightarrow \textit{A is located in} \\ \textit{where is the valley of kings?} &\rightarrow \textit{the valley of kings is located in} \end{aligned} \tag{9}$$

Question Classification

Durch die Klassifikation des erwarteten Antworttyps lassen sich der Suchaufwand verringern und die möglichen Ergebnisse einschränken. Bei einer Frage nach einem Personennamen kann die Suche entsprechend auf (kurze) Textpassagen beschränkt werden, in denen Eigennamen vorkommen, eine Frage nach einer Definition erfordert wahrscheinlich einen großeren Textausschnitt als Antwort.
Die Antworttypen werden in Taxonomien geordnet, die z.B. manuell oder mit maschinellen Lernverfahren als Regeln erstellt werden. Eine Regel für den Antwortyp PERSON wäre z.B.:

$$who\,\{is|was|are|were\}PERSON \tag{10}$$

Hinweise auf den Typ geben das Fragewort (*Wer*...), bestimmte Schlüsselwörter (*Welche Stadt ...*) oder Eigennamen. Um diese in der Frage ausfindig zu machen, werden z.B. POS-Tagging und Abgleiche mit Namens- und Ortsregistern verwendet.

2.2 Passage Retrieval

Beim passage retrieval wird die vorverarbeitete Frage einem IR-System übergeben, das aus dem Textkorpus relevante Dokumente zurück liefert, die bestenfalls eine Antwort auf die Frage beinhalten. Diese Dokumente werden in Textpassagen (Sätze, Absätze) unterteilt, diese Abschnitte werden anhand des erwarteten Antworttyps gefiltert und anschließend rangiert.
Wird z.B. als Antwortfakt ein Personenname erwartet, können alle Passagen, die keine Personennamen enthalten, aussortiert werden.

2.3 Answer Processing

Im letzten Schritt des factiod question answering wird die Antwort auf die ursprüngliche Frage aus den gefundenen und rangierten Textpassagen extrahiert. Hierzu bieten sich insbesondere die Methoden des Mustervergleichs mit einem Antwortmuster (answer-type pattern extraction) und die des N-gram tiling an.

Answer-type pattern extraction

Beim musterbasierten Extrahieren wird das erwartete Textmuster der Antwort mittels regulärer Ausdrücke mit den gefundenen Textpassagen verglichen. Hierbei spielt der erwartete Antworttyp eine große Rolle: Für eine Frage nach einer Person (answer-type HUMAN) kommen als Antwort nur noch entsprechende Entitäten in Frage (z.B. ermittelt mit einem named entity tagger).
Die schwierigere Antwort auf eine Frage nach einer Definition kann anhand bestimmer Muster extrahiert werden: Diese Muster können zum einen manuell eingepflegt werden, zum anderen kommen maschinelle Lernverfahren zum Einsatz, die aus

Question	Pattern	Answer
What is a <caldera>?	<QP>, a <AP>	the <caldera>, a <volcanic crater>

vorgegebenen Relationen und einer Textbasis solche Muster erzeugen können. Mit einer z.B. vorgegebenen Relation (person-name *rightarrow* year-of-birth) und gegebenen Anfangsdaten (mozart:1756) sucht der Lernalgorithmus in den Texten nach Sätzen mit diesen Informationsschlüsseln und erzeugt bestenfalls ein oder mehrere präzise Muster. Für die Relation (person-name *rightarrow* year-of-birth) könnten dies z.B.
`<NAME> (<BD> - <YD>)`
oder
`<NAME> was born on <BD>`
sein. Um eine Antwort erfolgsversprechend zu extrahieren bietet es sich an, verschiedene Extraktionsverfahren zusammen mit anderen Hinweisen zu kombinieren.

N-gram tiling

Eine weitere, oft im Zusammenhang mit den von Internetsuchmaschinen gefundenen Textpassagen ist das Kacheln mit N-grammen. Die N-gramme der zerlegten Textpassagen werden anhand ihrer Vorkommenshäufigkeit und anhand des Anfragemusters gewichtet, mit dem Antworttyp verglichen und jene von hohem Rang überlappend zu einer Antwort zusammengefügt.

Literatur

[1] Jurafsky, Dan, Martin, James H.: Speech and language processing : an introduction to natural language processing, computational linguistics and speech recognition. London ^{2}2009.

[2] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. New York 1999.