Relevanzgewichtung in komplexen Ähnlichkeitsanfragen

Nadine Schulz Ingo Schmitt

{nschulz|schmitt}@iti.cs.uni-magdeburg.de

Zusammenfassung: Eine Möglichkeit, um Nutzerpräferenzen in heutigen Multimedia- und Information Retrieval Systemen abzubilden, besteht in der Gewichtung von Anfragetermen. Die meisten Arbeiten auf diesem Gebiet sind in ihrer Anwendung auf das Gewichten einfacher Anfragen beschränkt. Aus diesem Grund wird in diesem Papier die Relevanzgewichtung auf verschiedenen Ebenen von komplexen Anfragen betrachtet. Es werden zwei Alternativen zur Formulierung von Gewichten in komplexen Anfragen demonstriert. Des Weiteren werden Transformationsregeln für gewichtete Anfragen untersucht, die eine Optimierung der Anfragebearbeitung ermöglichen.

1 Einführung und Motivation

In den vergangenen 20 Jahren entwickelte sich das Gewichten von Anfragetermen als eine Möglichkeit, um Nutzerpräferenzen in Multimedia und Information Retrieval Systemen abzubilden [DP86, FW97, SFW83, Sun98, WK79, Yag87]. Die Vergabe von Relevanzgewichten für bestimmte Anfrageterme in Fuzzy oder Multimedia-Anfragen ermöglicht dem Nutzer eine flexible Spezifikation von Präferenzen. Für einfache Anfragen, also Anfragen, die jeweils nur aus der Konjunktion bzw. Disjunktion von Anfragetermen bestehen, können derzeitige Verfahren [Yag87, DP86, FW97, Sun98] zur Handhabung von Relevanzgewichten erfolgreich eingesetzt werden, obwohl sie verschiedene Schwachstellen, wie in [Sun98, FW00] diskutiert, aufweisen. Bei diesem Typ von Anfragen wird die Gewichtung eines Anfrageterms jeweils gegenüber allen anderen Termen in der Anfrage betrachtet.

Im Gegensatz zu den einfachen Anfragen stehen die komplexen Anfragen. Hier können einzelne Terme oder Teilanfragen gegenüber anderen Teilanfragen gewichtet werden. Damit ist in komplexen Anfragen eine Gewichtung auf verschiedenen Ebenen möglich. Zwei Arten von komplexen Anfragen können unterschieden werden. Heterogene komplexe Anfragen bestehen sowohl aus Konjunktionen als auch aus Disjunktionen von Anfragetermen. Homogene komplexe Anfragen hingegen bestehen jeweils nur aus der Konjunktion bzw. Disjunktion von Anfragetermen.

Derzeitig gibt es keine Lösung für die Verarbeitung von Gewichten auf verschiedenen Ebenen in komplexen Anfragen. Ziel ist es daher, in komplexen Anfragen eine Gewichtung auf allen Ebenen zu ermöglichen und Transformationsregeln für die Optimierung dieser Anfragen bereitzustellen. Es wird die Entwicklung eines neuen Ansatzes, der als Multi-

Level-Gewichten bezeichnet wird, angestrebt.

Ein weiterer Aspekt, der bei der Entwicklung unseres Ansatzes betrachtet werden soll, ist die syntaktische Umformung von Anfragen in andere, für die Optimierung besser geeignete Anfrageformen durch das Retrieval System. Die syntaktische Transformation von Anfragen in logisch äquivalente Anfragen ist eine wichtig Eigenschaft von Retrieval Systemen mit Booleschen Operationen. Während die Transformation von ungewichteten Booleschen Anfragen ohne Probleme möglich ist, stellt die Transformation von gewichteten Anfragen ein bisher offenes Problem in Retrieval Systemen dar [Boo78, WK79]. Bei der Transformation muss sichergestellt werden, dass für syntaktisch unterschiedliche jedoch semantisch äquivalente, gewichtete Anfragen stets das gleiche Anfrageergebnis berechnet wird. Zur Umgehung dieses Problems werden die Nutzer derzeitig oftmals bei der Formulierung ihrer Anfragen eingeschränkt, indem sie beispielsweise ihre Anfrage direkt in disjunktiver Normalform (DNF) bzw. konjunktiver Normalform (KNF) formulieren müssen [HV01, Pas99]. Diese Herangehensweise stellt für den Nutzer eine zu starke Einschränkung dar. Aus diesem Grund bildet die Entwicklung von geeigneten logischen Transformationsregeln für das Multi-Level-Gewichtungsmodell einen Schwerpunkt in diesem Papier.

Im Weiteren ist die Arbeit wie folgt gegliedert. In Abschnitt 2 wird das zugrundeliegende Anfragemodell sowie, die in diesem Papier verwendeten Notationen dargelegt. Abschnitt 3 beschreibt zwei Alternativen zur Formulierung von gewichteten Anfragen. Weiterhin werden in Abschnitt 4 Anforderungen der logischen Umformung von gewichteten Anfragen erläutert. Es wird auf entsprechende Umformungsregeln für gewichtete, komplexe Anfragen eingegangen. Die Evaluierung dieser Anfragen erfolgt durch gewichtete Scoring-Funktionen, die mit dem Ansatz von Fagin und Wimmers [FW97] generiert werden. Abschließend wird ein Überblick über offenen Probleme und weiteren Forschungsaufgaben gegeben.

2 Anfragemodell

Eine Anfrage X besteht aus n atomaren Suchbedingungen x_i, die durch die Junktoren *und* ($\wedge$) und *oder* ($\vee$) miteinander verknüpft sind. Weiterhin, besteht die Möglichkeit Anfrageterme zu negieren:

$$X := x \mid (X\ [\wedge|\vee]\ X) \mid \neg X.$$

Eine komplexe Anfrage, wie in Abbildung 1 dargestellt, umfasst verschiedene Ebenen l_j mit $j = 0, \ldots, m$, so dass $m + 1$ die Gesamtanzahl der Ebenen angibt. Jede Ebene l_j enthält n_j verschiedene Teilanfragen $s_{j,i}$ mit $i = 1, \ldots, n_j$, wobei jede *und/oder*-Teilanfrage aus zwei Operanden besteht. Zur Bestimmung der direkten Kinder einer Teilanfrage $s_{j,i}$ wird die Funktion $child(s_{j,i})$ verwendet.

Nutzerpräferenzen können durch nummerische Gewichte θ_i mit $\theta_i \in [0, 1]$ und $\sum_{i=1}^{n} \theta_i = 1$ ausgedrückt werden. Typischerweise wird jedes Gewicht θ_i jeweils dem atomaren Anfrageterm x_i zugeordnet. Die Funktion $weight()$ ermittelt das Gewicht eines atomaren Anfrageterms oder einer Teilanfrage.

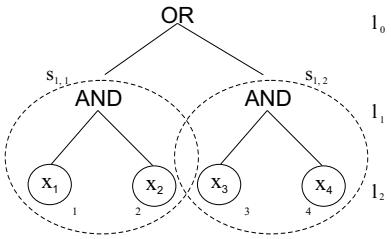

Abbildung 1: Anfragebaum einer komplexen Anfrage mit den Ebenen l_j, Teilanfragen $s_{j,i}$ und Gewichten θ_i

Bei der Evaluierung einer Anfrage X wird für jedes Objekt in der Datenbank und jede atomare Suchbedingung x_i ein Score μ_i im Intervall $[0, 1]$ berechnet, der ausdrückt inwieweit sich das Objekt die Suchbedingung x_i erfüllt. Je höher der Score μ_i ist, desto höher ist der Grad der Ähnlichkeit zwischen den Objekten bezüglich der Suchbedingung x_i. Anschließend wird für jedes Objekt der Gesamtscore μ berechnet. Dafür werden mittels der Junktoren *und* und *oder* jeweils zwei Fuzzy Mengen kombiniert und letztendlich eine neue Fuzzy-Menge als Gesamtscore für jedes Objekt in der Datenbank ermittelt. Die Kombination der Fuzzy Mengen für ein Objekt wird mittels einer Scoring-Funktion

$$S_X : [0, 1]^n \rightarrow [0, 1]$$

für eine Anfrage X realisiert. Typischerweise basiert S_X auf T-Normen und T-Conormen. Eine Konjunktion von Anfragetermen wird evaluiert unter Verwendung einer T-Norm und eine Disjunktion von Anfragetermen dementsprechend unter Verwendung einer T-Conorm (siehe [Zad65]). Der Einsatz von Fuzzy-basierten Scoring-Funktionen ist weit verbreitet, wie z. B. in den Ansätzen von [HV01, ORC+98, CBGM97]. Andere Scoring-Funktionen, wie beispielsweise probabilistische Funktionen, finden ebenso Anwendung [FG01].

Im Falle einer gewichteten Anfrage, wobei jeder Anfrageterm x_i mit einem Gewicht θ_i versehen ist, müssen die Gewichte der Anfrageterme mit in die Scoring-Funktion S_X einfließen. Es ergibt sich die gewichtete Scoring-Funktion

$$S_X^{\Theta} : [0, 1]^n \times [0, 1]^n \rightarrow [0, 1].$$

3 Multi-Level Gewichtung

Typischerweise wird jedes Gewicht einem atomaren Anfrageterm zugeordnet. Diese Einschränkung der Gewichtung auf atomare Terme ist in komplexen Anfragen nicht praktikabel, da dieses nicht dem Denkmuster des Nutzers entspricht und dieser daher die Auswirkungen seiner Gewichtung oft nicht nachvollziehen kann. Vielmehr sollte der Nutzer die Freiheit haben, Gewichte auf allen Ebenen einer komplexen Anfrage zu spezifizieren.

3.1 Implizites versus Explizites Gewichten

Für die Gewichtung von komplexen Anfragen adaptieren wir den Ansatz von [SSS02]. Demnach werden folgende zwei Alternativen unterschieden:

1. Implizites Gewichten: der Nutzer ordnet, wie auch bei einfachen Anfragen, jedem atomaren Term x_i ein Gewicht θ_i zu. Dabei gilt als globale Beschränkung $\sum_{i=1}^{n} \theta_i = 1$. Unter Berücksichtigung der gegebenen Gewichte für die atomaren Terme und der Semantik der Anfrage sind die Gewichte für die Teilanfragen $s_{j,i}$ auf einer höheren Ebene damit implizit bestimmt. Sie können aus den zugrundeliegenden atomaren Termgewichten ermittelt werden. Das Gewicht für eine Teilanfrage $s_{j,i}$ wird mittels einer Funktion f aus den Gewichten der Kinder der Teilanfrage berechnet. Einer Teilanfrage $s_{j,i}$ wird das Gewicht $\theta_{j,i} = f(weight(child(s_{j,i}))$ zugeordnet. Die implizite Gewichtung wird mit Θ bezeichnet.

2. Explizites Gewichten: der Nutzer spezifiziert explizit ein Gewicht θ_i für jeden atomaren Term x_i und ein Gewicht $\theta_{j,i}$ für jede Teilanfrage $s_{j,i}$. Die explizite Gewichtung wird mit Θ^E bezeichnet. Im Gegensatz zum impliziten Gewichten gibt es lokale Beschränkungen. Für jede Teilanfrage $s_{j,i}$ gilt, dass die Gewichte ihrer Kinder sich zu 1 summieren: $\forall s_{j,i} : \sum_{\theta_{j,t} \in weight(child(s_{j,i}))} \theta_{j,t} = 1$. Auf Grund der lokalen Beschränkung entspricht das Gewicht für die Teilanfrage $s_{0,1}$ immer 1, so dass das Gewicht $\theta_{0,1}$ nicht mit in die Gewichtung Θ^E aufgenommen wird.

Beide Alternativen der Formulierung von Gewichten unterscheiden sich in der Art der Spezifikation der Gewichte sowie in der Anzahl der zugrundeliegenden Beschränkungen. Jedoch verfügen sie über die gleiche Mächtigkeit. Die Gewichtung der Anfrageterme und Teilanfragen kann vom Anwender ohne Berücksichtigung der Bedingungen bei der Anfrageformulierung vorgenommen werden. Intern erfolgt eine Normalisierung der Gewichte, so dass die lokalen bzw. globalen Beschränkungen vom System eingehalten werden.

Als Initialgewichtung für eine Anfrage wird angenommen, dass alle Gewichte auf den einzelnen Ebenen gleich sind, was einer ungewichteten Anfrage entspricht. Dies bietet dem Nutzer die Möglichkeit, Gewichte nur für bestimmte Anfrageterme anzugeben. Eine andere Möglichkeit für die Festlegung der Initialgewichte besteht in der Verwendung von Nutzerprofilen.

In diesem Ansatz werden nummerische Gewichte verwendet. Jedoch ist es ebenso denkbar, statt dessen Fuzzy-Gewichte zu verwenden, die durch linguistische Variablen, wie z. B. *sehr wichtig*, *unwichtig*, usw. ausgedrückt werden können [HV01].

Im Folgenden zeigen wir die Evaluierung einer komplexen Anfrage mittels einer gewichteten Scoring-Funktion. Hierfür muss die Gewichtung in expliziter Form vorliegen. Daher ist es gegebenenfalls notwendig, eine implizite Gewichtung in die entsprechende explizite Gewichtung zu überführen. Für diese Umformung der Gewichtungen verwenden wir den in [SSS02] vorgestellten Ansatz. Die Überführung der beiden Gewichtungsformen ineinander ist invers und stellt somit eine Bijektion dar.

3.2 Gewichtete Multi-Level Scoring-Funktionen S_X^Θ

Für die Evaluierung von komplexen, gewichteten Anfragen sind Scoring-Funktionen vonnöten, die eine Handhabung von Gewichten auf den verschiedenen Ebenen ermöglichen. Hierfür wird auf gewichtete Scoring-Funktionen für einfache Anfragen zurückgegriffen [Yag87, DP86, FW97, Sun98], welche bereits erfolgreich eingesetzt werden. Da in komplexen Anfragen jede *und/oder*-Teilanfrage $s_{j,i}$ einer einfachen Anfrage entspricht kann durch die rekursive Anwendung der Scoring-Funktionen, die ursprünglich für einfache Anfragen entwickelt wurden, eine Gewichtung auf mehreren Ebenen einer komplexen Anfrage gewährleistet werden.

Wir greifen hier auf den Ansatz von Fagin und Wimmers zurück [FW97]. Dieser erlaubt die Gewichtung einer beliebigen zugrundeliegenden Scoring-Funktion S_X mit $\Theta = \langle\theta_1, \ldots, \theta_n\rangle$, wobei jedes Gewicht θ_i einem Anfrageterm x_i zugeordnet ist. Unter Beachtung der Annahmen $\theta_i \in [0,1]$, $\sum_{i=1}^{n} \theta_i = 1$ und $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_n$, welche mittels der Kommutativität erreicht wird, kann eine gewichtete Scoring-Funktion für einfache, gewichtete Anfragen mit folgender Formel generiert werden:

$$\begin{aligned} S_X^\Theta(\mu_1, \ldots, \mu_n) &= (\theta_1 - \theta_2) S_X(\mu_1) + 2 * (\theta_2 - \theta_3) S_X(\mu_1, \mu_2) \ldots + \ldots \\ &\quad n * \theta_n S_X(\mu_1, \ldots, \mu_n) \end{aligned} \tag{1}$$

Eine gewichtete Scoring-Funktion für komplexe Anfragen kann mit Hilfe der Formel (1) rekursiv gebildet werden. Scoring-Funktionen für komplexe Anfragen werden als Multi-Level Scoring-Funktionen bezeichnet. Für die Generierung einer Multi-Level Scoring-Funktion ist es notwendig, dass die Gewichtung für komplexe Anfragen in expliziter Form vorliegt bzw. in diese überführt wird.

4 Transformation von Anfragen

Eine Fähigkeit von Retrieval Systemen mit Booleschen Operationen ist die syntaktische Transformation von Anfragen in spezielle Anfrageformen. Durch die syntaktische Veränderung der internen Darstellung der Anfrage ist eine Minimierung des Ressourcenverbrauchs und damit eine performante Anfragebearbeitung möglich. Beispielsweise kann eine Anfrage syntaktisch so umgeformt werden, dass der Score für eine Teilanfrage effizient in einem System und der Score für eine andere Teilanfrage in einem anderen System ermittelt werden kann. Eine syntaktische Transformation von Anfragen setzt voraus, dass die Evaluierung äquivalenter Anfrageformen immer das selbe Ergebnis liefert.

4.1 Transformation ungewichteter Anfragen

Grundlage für die Transformation von Anfragen bilden logische Transformationsregeln. Für eine Boolesche Formel der Aussagen- bzw. der Fuzzy-Logik, müssen die allgemei-

nen Transformationsregeln, wie Kommutativität, Assoziativität, Distributivität, De Morgan, Idempotenz und Involution, gelten [Boo78].

Die Transformation von ungewichteten Anfragen beruht auf diesen Transformationsregeln und ist ohne Probleme möglich. Demgegenüber stellt die Transformation von gewichteten Anfragen jedoch ein bisher offenes Problem dar, da bei der Anfrageevaluierung der Einfluss der Gewichte berücksichtigt werden muss. Aus diesem Grund werden Nutzer in verschiedenen Systemen gezwungen, ihre gewichteten Anfragen in der DNF bzw. CNF zu formulieren [HV01, Pas99]. Diese Herangehensweise ist für den Nutzer nicht praktikabel. Vielmehr sollte der Nutzer die Freiheit besitzen, seinen Informationsbedarf in einer beliebigen Anfrageform auszudrücken. Die Last der logischen Umformung sollte daher nicht dem Nutzer, sondern dem Retrieval System im Rahmen der Anfrageoptimierung auferlegt werden.

Im folgenden Abschnitt beschreiben wir eine Möglichkeit der Transformation von gewichteten Anfragen. Unser Ansatz kann gewährleisten, dass für syntaktisch unterschiedliche Anfrageformen einer gewichteten Anfragen gleiche Ergebnisse ermittelt werden.

4.2 Transformation von gewichteten Anfragen

Für die Evaluierung gewichteter Anfragen verwenden wir gewichtete Scoring-Funktionen, die mittels der Faginschen Formel aus Abschnitt 3.2 generierten werden. Dafür werden hier konkret die Scoring-Funktionen MIN für Konjunktionen und MAX für Disjunktionen von Anfragetermen verwendet. Bei Fagin und Wimmers wird ein Gewicht jeweils einem Anfrageterm zugeordnet, so dass sich ein Paar (x_i, θ_i) mit dem Term x_i und dem Gewicht θ_i ergibt. Bei komplexen Anfragen gehen wir davon aus, dass die Gewichtung in expliziter Form vorliegt.

Für gewichtete, einfache Anfragen, die mit einer Faginschen Scoring-Funktion evaluiert werden, gelten die Eigenschaften Kommutativität, Involution, De Morgan und Idempotenz:

- Kommutativität: $((x_1, \theta_1) \wedge (x_2, \theta_2)) = ((x_2, \theta_2) \wedge (x_1, \theta_1))$,
  $((x_1, \theta_1) \vee (x_2, \theta_2)) = ((x_2, \theta_2) \vee (x_1, \theta_1))$.

  Bei der Berechnung des Gesamtscores mittels einer gewichteten Scoring-Funktion werden unter Ausnutzung der Kommutativität die Anfragen so umgeformt, dass die Bedingung der Faginschen Formel $\theta_1 \geq \ldots \geq \theta_n$ gilt.

- Idempotenz: Wenn $x_1 = x_2$, dann gilt: $((x_1, \theta_1) \wedge (x_2, \theta_2)) = x_1 = x_2$,
  $((x_1, \theta_1) \vee (x_2, \theta_2)) = x_1 = x_2$.

- Involution: $\neg\neg(x_1, \theta_1) = (x_1, \theta_1)$.

- De Morgan: $\neg((x_1, \theta_1) \wedge (x_2, \theta_2)) = ((\neg x_1, \theta_1) \vee (\neg x_2, \theta_2))$,
  $\neg((x_1, \theta_1) \vee (x_2, \theta_2)) = ((\neg x_1, \theta_1) \wedge (\neg x_2, \theta_2))$.

Die Gültigkeit dieser Eigenschaften kann sehr einfach bewiesen werden. Aus Platzgründen haben wir die Beweise an dieser Stelle nicht mit angegeben und verweisen auf [SSnt]. Bei einfachen, gewichteten Anfragen sind Distributivität und Assoziativität nicht von Bedeutung, jedoch bei komplexen, gewichteten Anfragen. Um bei diesem Typ von Anfragen die Distributivität und Assoziativität zu gewährleisten ist es notwendig, Gewichte zu modifizieren. Im Folgenden gehen wir auf diese beiden Transformationsregeln näher ein.

Distributivität

Es gilt:

$$\begin{aligned}((((x_1,\theta_1)\wedge(x_2,\theta_2)),\theta_{1,1})\vee(x_3,\theta_3)) &= ((((x_1,\theta_{1'})\vee(x_3,\theta_{3'})),\theta_{1,1'})\\ &\quad\wedge(((x_2,\theta_{2'})\vee(x_3,\theta_{3''})),\theta_{1,2'})),\\ ((((x_1,\theta_1)\vee(x_2,\theta_2),\theta_{1,1}))\wedge(x_3,\theta_3)) &= ((((x_1,\theta_{1'})\wedge(x_3,\theta_{3'}),\theta_{1,1'})\\ &\quad\vee((x_2,\theta_{2'})\wedge(x_3,\theta_{3''})),\theta_{1,2'})).\end{aligned}$$

In unseren Betrachtungen unterscheiden wir zwischen einer *und/oder*-Anfrage ($X_{\wedge,\vee}$) sowie einer *oder/und*-Anfrage ($X_{\vee,\wedge}$). Wichtig bei der logischen Transformation von Anfragen ist, dass sowohl für die ursprüngliche Anfrage X als auch für die transformierte Anfrage X' gleiche Ergebnisse ermittelt werden. Aus diesem Grund wird bei der distributiven Transformation einer komplexen Anfrage die Modifikation der Gewichte erforderlich. Wir gehen davon aus, dass $\theta_1 \geq \theta_2$ und $\theta_{1,1} \geq \theta_3$ gilt. Die Gewichte für die transformierte Anfrage können dann wie folgt aus den Gewichten der ursprünglichen Anfrage berechnet werden:

$$\theta_{3'} = \theta_3$$

$$\theta_{1,2'} = \begin{cases} \frac{1-(1-2\theta_3)(1-2\theta_2)}{2} & X_{\wedge,\vee}:\mu_1\geq\mu_3\geq\mu_2 \text{ oder } X_{\vee,\wedge}:\mu_2\geq\mu_3\geq\mu_1 \\ \theta_2 & \text{sonst} \end{cases}$$

$$\theta_{3''} = \begin{cases} \frac{\theta_3}{1-(1-2\theta_2)(1-2\theta_2)} & X_{\wedge,\vee}:\mu_1\geq\mu_3\geq\mu_2\wedge \\ & \quad(1-2\theta_2)\mu_1+2\theta_2\mu_2<\mu_3 \text{ oder} \\ & X_{\vee,\wedge}:\mu_1\geq\mu_3\geq\mu_2\wedge \\ & \quad(1-2\theta_2)\mu_1+2\theta_2\mu_2>\mu_3 \\ 0 & X_{\wedge,\vee}:\mu_1\geq\mu_3\geq\mu_2\wedge \\ & \quad(1-2\theta_2)\mu_1+2\theta_2\mu_2\geq\mu_3 \text{ oder} \\ & X_{\vee,\wedge}:\mu_1\geq\mu_3\geq\mu_2\wedge \\ & \quad(1-2\theta_2)\mu_1+2\theta_2\mu_2\leq\mu_3 \\ \theta_3 & \text{sonst} \end{cases}$$

Auf Grund der lokalen Beschränkungen können die Gewichte $\theta_{1'}, \theta_{2'}$ sowie $\theta_{1,1'}$ aus den bereits ermittelten Gewichten berechnet werden, z. B. $\theta_{1'} = 1 - \theta_{3'}$.

Der folgende Beweis für den Fall $\mu_1 \geq \mu_2 \geq \mu_3$ und einer *oder/und*-Anfrage zeigt die Korrektheit unserer ermittelten Gewichte. Die Beweise für die anderen Fälle sind analog. Unsere Ergebnisse haben wir ebenfalls experimentell bestätigt.

Beweis: Es soll gelten $S_X^{\Theta^E} = S_{X'}^{\Theta^E}$. Unter Verwendung der Faginschen Formel ergibt sich für X:

$$\begin{aligned} S_{s_{1,1}}^{\Theta} &= (1-2\theta_2)\mu_1 + 2\theta_2 MAX(\mu_1,\mu_2) = \mu_1 \\ S_X^{\Theta^E} &= (1-2\theta_3)S_{s_{1,1}}^{\Theta} + 2\theta_3 MIN(S_{s_{1,1}}^{\Theta},\mu_3) \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 \end{aligned}$$

und für X':

$$\begin{aligned} S_{s_{1,1}}^{\Theta} &= (1-2\theta_{3'})\mu_1 + 2\theta_{3'} MIN(\mu_1,\mu_3) \\ S_{s_{1,2}}^{\Theta} &= (1-2\theta_{3''})\mu_2 + 2\theta_{3''} MIN(\mu_2,\mu_3) \\ S_{X'}^{\Theta^E} &= (1-2\theta_{1,2'})S_{s_{1,1}}^{\Theta} + 2\theta_{1,2'} MAX(S_{s_{1,1}}^{\Theta}, S_{s_{1,2}}^{\Theta}) \\ &\quad \text{mit } \theta_{3'} = \theta_3, \theta_{3''} = \theta_3, \theta_{1,2'} = \theta_2 \\ &= S_{s_{1,1}}^{\Theta} \\ &= (1-2\theta_3)\mu_1 + 2\theta_3\mu_3 = S_X^{\Theta^E} \qquad q.e.d. \end{aligned}$$ □

Assoziativität

Es gilt:
$((((x_1,\theta_1) \wedge (x_2,\theta_2)),\theta_{1,1}) \wedge (x_3,\theta_3)) = ((x_1,\theta_{1'}) \wedge (((x_2,\theta_{2'}) \wedge (x_3,\theta_{3'})),\theta_{1,1'}))$,
$((((x_1,\theta_1) \vee (x_2,\theta_2)),\theta_{1,1}) \vee (x_3,\theta_3)) = ((x_1,\theta_{1'}) \vee (((x_2,\theta_{2'}) \vee (x_3,\theta_{3'})),\theta_{1,1'}))$

Die Herangehensweise zur Bestimmung der Gewichte für die transformierte Anfrage ist hier analog zur distributiven Transformation. Die Gewichte für die transformierte Anfrage können wie im Folgenden angegeben berechnet werden. Die übrigen Gewichte, also $\theta_{1'}$ und $\theta_{2'}$, werden auf Grund der lokalen Beschränkungen aus den bereits ermittelten Gewichten berechnet.

$$\theta_{1,1'} = \begin{cases} \theta_3 & X_\wedge : \mu_2 \geq \mu_1 \geq \mu_3 \text{ oder } X_\vee : \mu_3 \geq \mu_1 \geq \mu_2 \\ \frac{1-(1-2\theta_3)(1-2\theta_2)}{2} & X_\wedge : \mu_1 \geq \mu_2 \geq \mu_3, \mu_1 \geq \mu_3 \geq \mu_2 \wedge \\ & \quad (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 > \mu_3 \text{ oder} \\ & X_\vee : \mu_3 \geq \mu_2 \geq \mu_1, \mu_2 \geq \mu_3 \geq \mu_1 \wedge \\ & \quad (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 < \mu_3 \\ \theta_2 & \text{sonst} \end{cases}$$

$$\theta_{3'} = \begin{cases} 1 & X_\wedge : \mu_2 \geq \mu_1 \geq \mu_3 \text{ oder } X_\vee : \mu_3 \geq \mu_1 \geq \mu_2 \\ \frac{\theta_3}{2\theta_{1,1'}} & X_\wedge : \mu_1 \geq \mu_2 \geq \mu_3 \text{ oder } X_\vee : \mu_3 \geq \mu_2 \geq \mu_1 \\ 1 - \frac{(1-2\theta_3)\theta_2}{2\theta_{1,1'}} & X_\wedge : \mu_1 \geq \mu_3 \geq \mu_2 \wedge (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 > \mu_3 \\ & \text{oder} \\ & X_\vee : \mu_2 \geq \mu_3 \geq \mu_1 \wedge (1-2\theta_2)\mu_1 + 2\theta_2\mu_2 < \mu_3 \\ \theta_2 & \text{sonst} \end{cases}$$

Für die weitere Vereinfachung von gewichteten Anfragen geben Fagin und Wimmers zwei Regeln unter Berücksichtigung des Einflusses der Gewichte an [FW97]:

- Wenn ein atomarer Anfrageterm oder eine Teilanfrage mit Null gewichtet ist, dann kann dieser Term bzw. diese Teilanfrage von der Anfrage entfernt werden. Das Entfernen hat keinen Einfluss auf das Anfrageergebnis.
- Sind alle Gewichte gleich, also $\theta_i = 1/n$, entspricht die gewichtete Anfrage der ungewichteten Anfrage. Die Evaluierung der Anfrage kann ohne Berücksichtigung der Gewichte erfolgen, da in diesem Fall der gleiche Ergebniswert wie für die gewichteten Anfrage berechnet wird.

5 Zusammenfassung und Ausblick

Es wurde eine Methode zur Spezifikation von Relevanzgewichten für Anfrageterme in komplexen Anfragen beschrieben. Dieser Ansatz wird als Multi-Level-Gewichten bezeichnet, wobei zwei Alternativen der Gewichtung unterschieden werden. Bei der impliziten Gewichtung wird jedem atomaren Anfrageterm ein Gewicht zugeordnet. Gewichte für die Teilanfragen in komplexen Anfragen werden dann aus den gegebenen Gewichten intern berechnet. Die explizite Gewichtung gibt dem Nutzer die Freiheit, sowohl atomare Terme als auch Teilanfragen explizit mit einem Gewicht zu versehen. Eine Überführung beider Alternativen ineinander ist möglich und notwendig für die Evaluierung einer komplexen Anfrage mittels einer gewichteten Multi-Level Scoring-Funktion.

Die Anfrageoptimierung, welche auf logischen Transformationsregeln basiert, ist in einem Retrieval System wesentlich. Es wurde darauf eingegangen, wie Multi-Level-gewichtete Anfragen logisch umgeformt werden können. Dafür wurden Transformationsregeln angegeben, die eine logische Umformung von gewichteten, komplexen Anfragen erlauben. Die Evaluierung der gewichteten Anfragen erfolgt hier auf Basis der gewichteten Scoring-Funktionen von Fagin und Wimmers. Wir haben gezeigt, wie die Gewichte bei einer distributiven und assoziativen Umformung einer gewichteten, komplexen Anfrage modifiziert werden müssen, so dass sowohl für die ursprüngliche als auch für die transformierte Anfrage stets gleiche Gesamtscores ermittelt werden.

Für unsere weitere Arbeit stehen weitere Untersuchungen zur logischen Umformung von gewichteten, komplexen Anfragen im Vordergrund. Ferner interessiert uns, inwieweit eine Vereinfachung der Formeln zur Berechnung der Gewichte möglich ist. Ein weiteres Ziel ist es, die Transformationsregeln für gewichtete Anfragen in ein Retrieval System einzubetten, um dadurch eine Anfrageoptimierung implementieren zu können.

Literaturverzeichnis

[Boo78] A. Bookstein. On the perils of merging Boolean and weighted retrieval systems. *Journal of the American Society for Information Science, 29*, Seiten 156–158, 1978.

[CBGM97] C. Carson, S. Belongie, H. Greenspan, und J. Malik. Region-based image querying. In *Proc. of the IEEE Workshop CVPR '97 Workshop on Content-Based Access of Image and Video Libraries, Puerto Rico*, Seiten 42–49, 1997.

[DP86] D. Dubois und H. Prade. Weighted Minimum and Maximum Operations in Fuzzy Set Theory. *Information Science 39*, Seiten 205–210, 1986.

[FG01] N. Fuhr und K. Gro'sjohann. XIRQL: A Query Language for Information Retrieval in XML Documents. In *Proceedings of the 24th Annual International Conference on Research and development in Information Retrieval, ACM, New York*, Seiten 172–180, 2001.

[FW97] R. Fagin und E.L. Wimmers. Incorporating User Preferences in Multimedia Queries. In *Proc. 6th International Conference on Database Theory*, Seiten 247–261. Springer-Verlag, LNCS 1186, Delphi, 1997.

[FW00] R. Fagin und E.L. Wimmers. A Formula for Incorporating Weights into Scoring Rules. *Theorectical Computer Science*, 239:309–338, 2000.

[HV01] E. Herrera-Viedma. An Information Retrieval System with Ordinal Linguistic Weighted Queries Based on Two Weighting Semantics. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9:77–88, 2001.

[ORC+98] M. Ortega, Y. Rui, K. Chakrabarti, K. Porkaew, S.Mehrotra, und T.S. Huang. Supporting Ranked Boolean Similarity Queries in MARS. *Knowledge and Data Engineering*, 10(6):905–925, 1998.

[Pas99] G. Pasi. A logical formulation of the Boolean model and of weighted Boolean models. In *Proceedings of the Workshop on Logical and Uncertainty Models for Information Systems (LUMIS 99), University College London, Inghilterra*, 1999.

[SFW83] G. Salton, E.A. Fox, und H. Wu. Extended Boolean Information Retrieval. *Communications of the ACM*, 26(11):1022–1036, 1983.

[SSnt] N. Schulz und I. Schmitt. Logical Transformation Rules for Complex Weighted Queries. *Preprint, Otto-von-Guericke Univeristät, Magdeburg*, erscheint.

[SSS02] I. Schmitt, N. Schulz, und G. Saake. Multi-Level Weighting in Multimedia Retrieval Systems. In *Proceedings of the 2nd Int. Workshop on Multimedia Data Document Engineering (MDDE'02), Prague, Czech Republic*, Seiten 353–364. Springer-Verlag, LNCS 2490, 2002.

[Sun98] S.Y. Sung. A Linear Transform Scheme for Combining Weights into Scores. *Technical Report, Rice University*, 1998.

[WK79] W. G. Waller und D. H. Kraft. A mathematical model for a weighted Boolean retrieval system. *Information Processing and Management*, 15(5):235–245, 1979.

[Yag87] R. R. Yager. A note on weighted queries in information retrieval systems. *Journal of the American Society for Information Science,38*, Seiten 23–24, 1987.

[Zad65] L. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.