Seminar: XML-basierte Datenbanken
Prof. Dr.-Ing. Gerhard Weikum
Dipl.-Inform. Anja Jantke

Information Retrieval auf XML-Dokumenten

Michael Koch

Inhaltsverzeichnis

1 Einführung

Information Retrieval dient der Suche von Information in einer Menge von Daten, üblicherweise Dokumenten. Bekannte Beispiele für Information Retrieval Systeme sind die in den letzten Jahren entwickelten Internet-Suchmaschinen. Ein klassisches Anwendungsgebiet ist die Suche in Literaturdatenbanken.

Die üblichen Retrievalmodelle betrachten nur den textuellen Inhalt von Dokumenten. Im Gegensatz dazu nutzt das hier vorgestellte Modell die in XML-Dokumenten enthaltenen Strukturinformationen. Es ermöglicht, in Suchanfragen zusätzlich zu inhaltlichen auch strukturelle Bedingungen zu stellen. Durch die Ausnutzung der Strukturinformation wird die Qualität der Antworten verbessert. Das Modell wurde in [1] beschrieben.

2 Information Retrieval ohne Strukturinformation

2.1 Dokumentretrieval

Information Retrieval wie es hier verstanden wird wird auch als *Dokumentretrieval* bezeichnet. Gesucht wird nach Information in einer Menge von Dokumenten, die Wissen enthalten. Als Information wird dabei das Wissen verstanden, das der Benutzer des Retrieval-Systems gerade benötigt. Der Benutzer stellt eine Anfrage (*Query*) an das System und erhält als Antwort eine Teilmenge der Dokumente.

Die Auswahl der zurückgegebenen Dokumente trifft das zugrundeliegende *Retrievalmodell*. Ein *relevantes* Dokument ist ein Dokument, das (subjektiv) zu einer Anfrage paßt. Bei einem guten Retrievalmodell ist der Anteil an relevanten Dokumenten in der Antwortmenge hoch, und ein hoher Prozentsatz aller relevanten Dokumente aus der Suchmenge wird zurückgegeben.

2.2 Boolesches Retrievalmodell

Ein einfaches Retrievalmodell ist das *Boolesche Retrieval*. Das Modell faßt ein Dokument als eine Menge von *Termen* auf. Terme sind Wörter in der Form, wie sie im Index des Retrievalsystems abgelegt sind. Beispielsweise können bei der Indexierung alle Wörter klein geschrieben werden, oder es werden Füllwörter wie “der” oder “bei” entfernt.

Eine Query besteht aus einer Menge von Termen, die durch boolesche Operatoren wie “UND” verknüpft sind. Die Antwortmenge sind alle Dokumente, die die Terme enthalten und die Verknüpfungsbedingungen erfüllen.

Ein Problem dieses Retrievalmodells ist, das die zurückgegebene Dokumentmenge groß ist, und das die Dokumente in zufälliger Reihenfolge ausgegeben werden. Andere Retrievalmodelle wie das Vektorraummodell berechnen deshalb ein *Ranking*, das schätzt, wie gut ein Dokument zur Query paßt. Die Dokumente werden dann in der Reihenfolge des Rankings zurückgegeben.

2.3 Vektorraummodell

Das Vektorraummodell ([2]) berechnet ein Maß für die Übereinstimmung der Konzepte von Dokumenten. Die Query wird als ein Dokument behandelt, dies ermöglicht eine natürlichspachige Formulierung der Anfrage. Die Dokumente werden in der Reihenfolge der Übereinstimmung mit der Query zurückgegeben.

Für jeden Term in einem Dokument wird ein Termgewicht w_i berechnet, das seine Wichtigkeit im Vergleich mit anderen Termen ausdrückt. Ein Dokument wird als ein Vektor aus den Gewichten aller Terme im Index dargestellt, wobei das Gewicht eines Terms, das nicht im Dokument vorkommt, 0 ist.

$$\vec{d} = \begin{pmatrix} w_1 \\ \vdots \\ w_i \\ \vdots \\ w_n \end{pmatrix} \tag{1}$$

Die Übereinstimmung zweier Dokumente berechnet sich nun als der Abstand zwischen den Dokumentvektoren. Als Abstandsmaß wird üblicherweise das Skalarprodukt benutzt. Da das Gewicht eines Terms, das nicht im Dokument vorkommt, 0 ist und damit nicht zum Skalarprodukt beiträgt, reicht es bei dem Vergleich der Dokumente nur die Terme zu betrachten, die in beiden Dokumenten vorkommen.

$$r = \vec{q} \cdot \vec{d} \tag{2}$$

2.4 Gewichtung der Terme

Im folgenden werden zwei Maße vorgestellt, die die Wichtigkeit eines Terms im Vergleich mit anderen Termen ausdrücken. Sie werden zum Termgewicht w_i kombiniert.

2.4.1 Term Frequency (TF)

Die Idee der Term Frequency ist, das ein Term, der oft in einem Dokument vorkommt, für dieses Dokument wichtig ist, den Inhalt also gut beschreibt. tf_{di} ist die Anzahl der Vorkommen des Terms i im Dokument d. Um zu vermeiden, daß häufig vorkommende Terme zu stark zum Termgewicht beitragen, wird diese Zahl logarithmiert.

$$\mathrm{TF}_{di} = 1 + \ln \mathrm{tf}_{di} \tag{3}$$

2.4.2 Inverse Document Frequency (IDF)

Die Annahme der Inverse Document Frequency ist, ein Term, die in der Dokumentsammlung selten vorkommt, ein Dokument besser repräsentiert. Beispielsweise wird der Term “XML” von der Suchmaschine Altavista[1] in ca. 500000 Dokumenten gefunden, der Term “example” in 5 Millonen. Das heißt, daß “XML” als speziellerer Begriff ein Dokument besser beschreibt.

Die IDF berechnet sich als

$$\mathrm{IDF}_i = \ln \frac{N}{\mathrm{df}_i} \tag{4}$$

N ist die Anzahl der Dokumente in der Sammlung. df_i ist die Anzahl der Dokumente, in der Term i vorkommt.

2.4.3 Termgewicht w_i

Aus TF und IDF berechnet sich das unnormierte Termgewicht

$$\alpha_{di} = \mathrm{TF}_{di} \cdot \mathrm{IDF}_i \tag{5}$$

Mit diesem Maß würden lange Dokumente bevorzugt: Entweder besteht ein langes Dokument aus vielen verschiedenen Termen. Dann ist die Wahrscheinlichkeit, das ein Term sowohl in der Query als auch im Dokument vorkommt, groß. Oder einige Terme kommen sehr oft im Dokument vor, wodurch die Term Frequency ansteigt.

Um Dokumente unterschiedlicher Länge gleich stark zu bewerten, wird der Dokumentvektor normiert.

$$w_i = \frac{\alpha_{di}}{|\mathrm{Dokumentvektor}|} \tag{6}$$

[1] http://www.altavista.com

Methought I heard a voice cry "*Sleep* no more!
Macbeth does murder *sleep*," the innocent *sleep*,
Sleep that knits up the raveled sleave of care...

Shakespeare, Macbeth, 2.2.39-41

Abbildung 1: Beispieldokument

2.5 Beispiel zum Vektorraummodell

Im folgenden Beispiel wird das Termgewicht des Wortes *sleep* im Dokument 2.5 berechnet. Dann wird die Übereinstimmung des Dokuments mit der Query "sleep" bestimmt.

2.5.1 Termgewicht von *sleep*

Sleep kommt im Dokument 4 mal vor. Nach Formel 3 ergibt sich die Term Frequency $\mathrm{TF}_{d,sleep} = 1 + \ln 4 \approx 2,4$.

Um die IDF zu berechnen, müssen Annahmen über die Dokumentmenge gemacht werden. Angenommen, *sleep* kommt in 10 von 100 Dokumenten in der Sammlung vor. Die IDF ist nach Formel 4 $\mathrm{IDF}_{sleep} = \ln \frac{100}{10} \approx 2,3$.

Das unnormierte Termgewicht ist damit $\alpha_{d,sleep} = 2,4 \cdot 2,3 \approx 5,5$. Zur Normierung muß die Länge des Dokumentvektors und dafür das unnormierte Gewicht der anderen Terme im Dokument berechnet werden. Angenommen, $|\vec{d}| = 20$. Nach Formel 6 folgt $w_{d,sleep} = \frac{5,5}{20} \approx 0,28$.

2.5.2 Vergleich mit Query "sleep"

Angenommen, eine Query besteht nur aus dem Wort "sleep". Die TF ist 1; die IDF wird aus der Dokumentsammlung genommen, ist also 2,3. Die Länge des unnormierten Queryvektors ist $1 \cdot 2,3 = 2,3$; $w_{q,sleep} = 1$. Die Übereinstimmung der Query mit dem vorher berechneten Dokumentvektor ist

$$r = \vec{q} \cdot \vec{d} = (1) \cdot (0,28) = 0,28$$

```
<collection>
  ...
  <article>
    <chapter>
      <title>Struktur von XML</title>
      <section>Eine XML-Datei besteht ...</section>
    </chapter>
  </article>
  ...
</collection>
```

Abbildung 2: Ausschnitt aus einem XML-Dokument

3 Retrievalmodell für strukturierte Dokumente

3.1 Ausnutzung der Dokumentstruktur

Das im vorigen Abschnitt vorgestellte Vektorraummodell benutzt zum Ranking nur den Inhalt eines Dokuments. Ein komplexeres Modell kann die Strukturinformationen von XML nutzen um neue Funktionalität gegenüber dem Dokumentretrieval zu implementieren und das Ranking zu verbessern.

Im Folgenden wird ein Retrievalmodell für XML-Dokumente aus [1] vorgestellt, das diese Möglichkeiten nutzt.

3.2 Struktur von XML

Eine XML-Datei besteht aus inhaltlichem Text (*Character Data*) und Markup (*Tags*), das den Text enthält. Die Tagnamen beschreiben den von dem Tag umschlossenen Text (logische Textauszeichnung).

Ein Element besteht aus einem Tag und seinem Inhalt (Text und/oder andere Elemente). Beispielsweise besteht das title-Element in Abbildung 3.1 aus dem <title>-Tag und dem Text "Struktur von XML". Das chapter-Element besteht aus <chapter> und den Elementen title und section. Die hierarchische Verschachtelung drückt Beziehungen zwischen Elementen aus.

Weitere Struktureigenschaften von XML-Dateien sind Attribute und Hyperlinks. Sie werden aber von dem vorgestellten Retrievalmodell nicht genutzt.

FIND <article> WHICH INCLUDES <chapter> WHOSE <title> CONTAINS ‘Struktur’ AND WHOSE <section> CONTAINS ‘XML’

Abbildung 3: Beispielquery

3.3 Eigenschaften des Retrievalmodells

Die Query in Abbildung 3.3 zeigt eine Query, wie sie an das Retrievalsystem gestellt werden könnte. Sie illustriert die vom Retrievalmodell unterstützten Konzepte:

- Retrievalziel ist ein beliebiges Element (*Retrieval Element*)
  (FIND <article>)
  Beim Dokumentretrieval wurde als Suchergebnis eine Menge von Dokumenten zurückgegeben. Hier können Elemente, die in beliebiger Tiefe in einem Dokument vorkommen, als Suchziel angegeben werden. Zurückgegeben werden Links auf die Elemente.

- Angabe von strukturelle Bedingungen
  (INCLUDES <chapter> WHOSE <title>)
  Nur Elemente, die die strukturellen Bedingungen erfüllen, werden zurückgegeben.

- Angabe von inhaltliche Bedingungen
  (<section> CONTAINS ‘XML’)
  Das Element muß den Term enthalten.

- Ausnützung der Dokumentstruktur zur Verbesserung des Rankings.

3.4 Inferenznetzmodell

Mathematische Grundlage für das Retrievalmodell ist ein Inferenznetz. Ein Inferenznetz ist ein gerichteter Graph. Die Knoten repräsentieren Zufallsvariablen mit den Werten ‘Wahr’ oder ‘Falsch’. Eine Kante drückt eine kausale Abhängigkeit des Kind- vom Vater-Knoten aus. Jede Kante wird mit einer bedingten Wahrscheinlichkeit beschriftet, die die Stärke der Abhängigkeit ausdrückt.

Die Stärke der Abhängigkeit eines Knoten von einem beliebigen anderen kann duch die Propagierung der Wahrscheinlichkeiten durch den Graph berechnet werden.

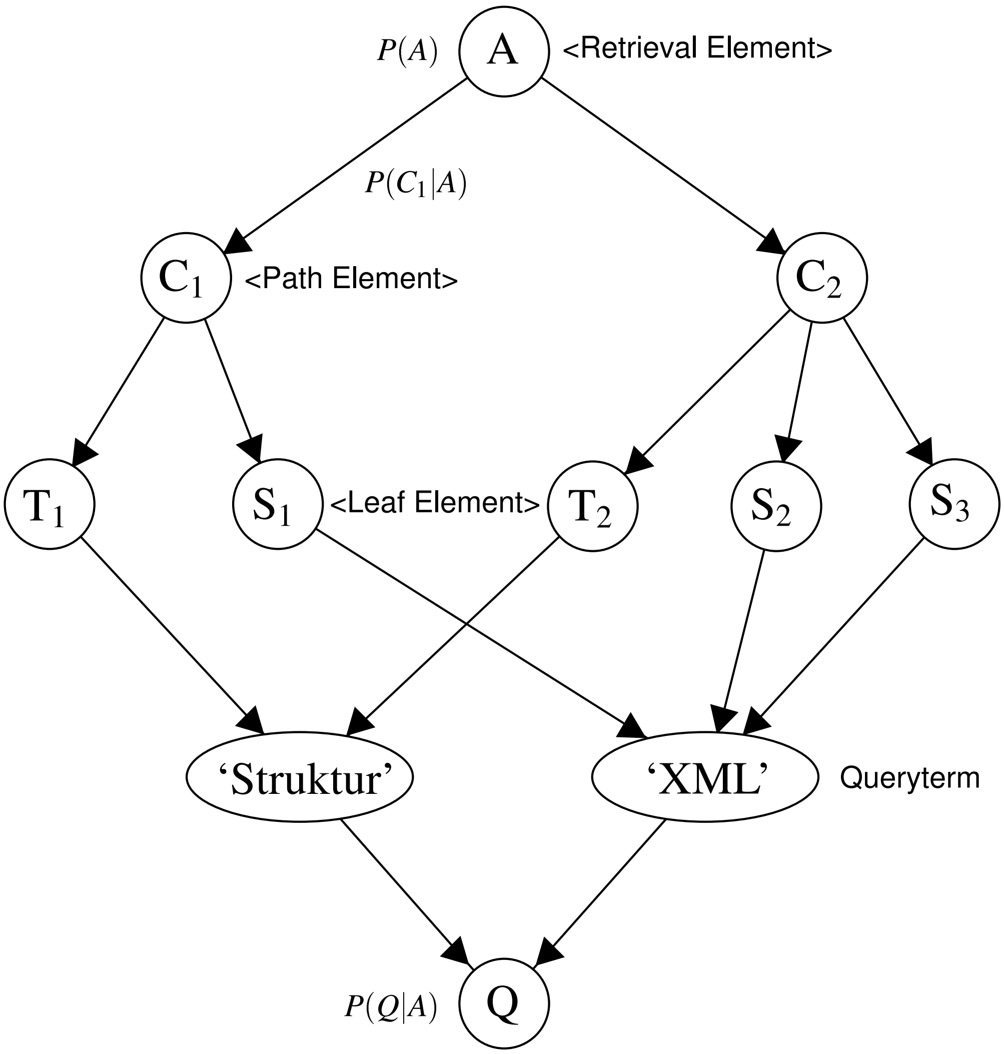

Abbildung 4: Inferenznetz für Query 3.3 und ein Retrieval Element

3.5 Anwendung auf die Suchmenge

Als Rankingwert berechnet das Retrievalmodell für jedes Retrievalelement die Abhängigkeit der Query von jedem Retrieval Element in der Dokumentsammlung. Dazu wird ein Inferenznetz wie in Abbildung 3.4 für jedes Retrieval Element erzeugt. Die Wahrscheinlichkeit $P(Q|A)$ wird als Maß der Abhängigkeit für das Ranking benutzt.

Knoten im Graphen sind:

- Das Retrieval Element und ein Knoten "Query".
- Jeder Queryterm.
- Elemente, die mindestens einen Queryterm enthalten (*Leaf Element*).
- Elemente, die ein Leaf Element enthalten und vom Retrieval Element enthalten werden (*Path Element*).

Wenn im XML-Dokument Element A Element B enthält, ist im Graph Knoten B Kind von Knoten A. Ein Queryterm-Knoten ist Kind aller Leaf Elements, die diesen Term enthalten. Der Knoten "Query" ist Kind aller Queryterm-Knoten.

Wie die Wahrscheinlichkeit $P(Q|A)$ berechnet wird, wird im folgenden Abschnitt erläutert.

3.6 Berechnung der Wahrscheinlichkeiten

Abbildung 3.6 zeigt ein einfaches Netzwerk. Gesucht ist die Wahrscheinlichkeit P(QT|C), die sich mit der folgenden Formel berechnen läßt wenn die Kanten des Graphen mit bedingten Wahrscheinlichkeiten beschriftet sind:

$$P(QT|C) = P(QT|S_1)P(S_1) + P(QT|S_2)P(S_2) \tag{7}$$

Die Wahrscheinlichkeiten $P(S_1)$ und $P(S_2)$ sind nicht bekannt. Da die Knoten aber wiederum vom Knoten C abhängen ($P(S_1) = P(S_1|C)$) kann die Formel rekursiv angewendet werden.

$$P(QT|C) = P(QT|S_1)P(S_1|C)P(C) + P(QT|S_2)P(S_2|C)P(C) \tag{8}$$

Die Formel läßt sich leicht auf den Graph aus Abbildung 3.4 erweitern.

Die Wahrscheinlichkeit des Wurzelknotens C hängt von keinem anderen Knoten ab. Sie kann frei gewählt werden. Im einfachsten Fall kann man alle Retrieval Elements gleich stark gewichten und $P(C) = 1$ setzen. Eine komplexere Möglichkeit ist, die Wahrscheinlichkeit eines Elements zu erhöhen, wenn ein Benutzer

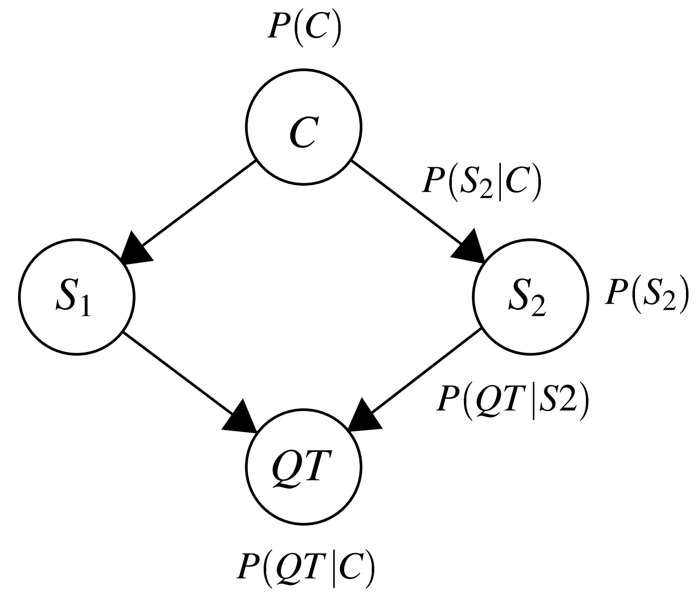

Abbildung 5: Beispielnetzwerk

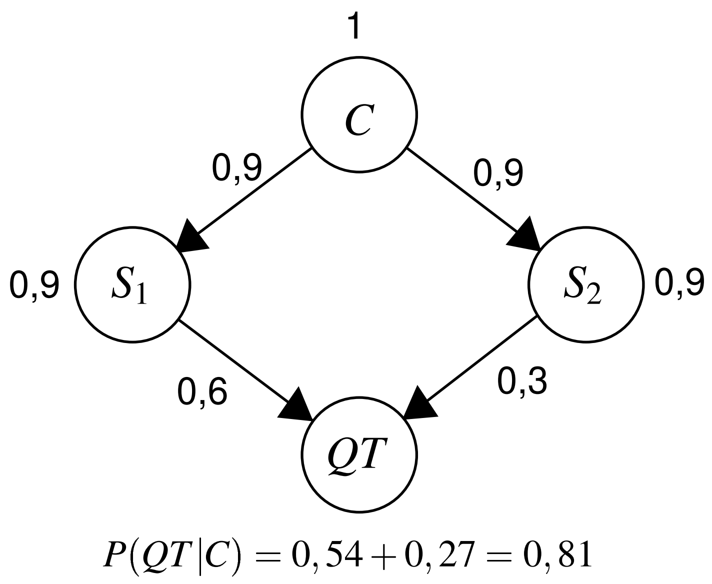

Abbildung 6: Rechenbeispiel

dieses Element als relevant bewertet. Die Bestimmung der bedingten Wahrscheinlichkeiten entlang der Kanten wird im nächsten Abschnitt beschrieben.

Abbildung 3.6 zeigt ein Rechenbeispiel.

3.7 Wahl von $P(B|A)$

Die Wahl der bedingten Wahrscheinlichkeiten entscheidet darüber, wie stark die Knoten zum Ranking beitragen. Hierzu wird die Information über Inhalt und Struktur des Dokuments eingebracht. Dazu werden die im ersten Abschnitt eingeführten Termgewichte und das Vektorraummodell benutzt.

Alle Wahrscheinlichkeiten können bereits zur Indexierungszeit berechnet werden. Bei der Queryausführung müssen nur noch die Wahrscheinlichkeiten entlang der Kanten multipliziert werden.

Es gibt 4 Arten von Kanten im Graph:

3.7.1 B Queryterm, A Leaf Element

Hier soll die Wahrscheinlichkeit ausdrücken, wie gut der Term den Inhalt des Elements beschreibt. Dazu kann das Termgewicht aus Abschnitt 2.4 benutzt werden.

$$P(QT|A) \propto \mathrm{TF}_{QT,A} \cdot \mathrm{IDF}_{QT} \tag{9}$$

Wenn die Struktur der Dokumente in der Sammlung bekannt ist, kann zusätzlich die Art des Tags gewichtet werden. Beispielsweise können Terme in einem <title>-Tag stärker gewichtet werden als Terme in einem <section>-Tag, unter der Annahme, das der Titel den Inhalt des Vaterelements besser beschreibt.

3.7.2 B Leaf Element, A Path Element

Es werden zwei verschiedene Ansätze beschrieben, um die Wahrscheinlichkeit zu bestimmen.

Repräsentativität von B$_i$ für A Mit Hilfe des Vektorraummodells wird der Inhalt von B$_i$ mit dem gesamten Inhalt des Elements A verglichen. Eine hohe Übereinstimmung bedeutet, das B$_i$ den Inhalt von A gut repräsentiert. Die Wahrscheinlichkeit der Kante wird proportional zur Übereinstimmung gewählt. Zusätzlich kann Information über den Tag in die Gewichtung einfließen.

Kohärenz des Inhalts des Kindelements zu den anderen Kindelementen Mit dem Vektorraummodell wird der Inhalt des Elements mit seinen Geschwistern verglichen. Ist die Übereinstimmung niedrig, wird angenommen, das das Element

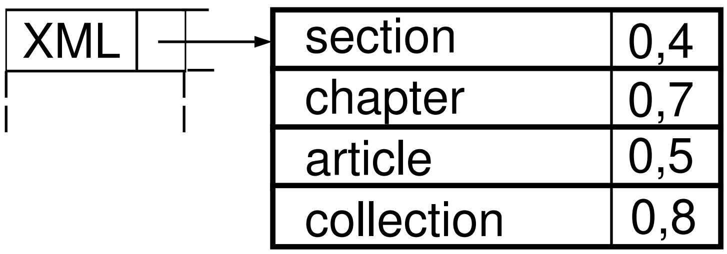

Abbildung 7: Indexstruktur

den Inhalt von A schlecht beschreibt und die Wahrscheinlichkeit an der Kante entsprechend niedriger gewählt.

3.7.3 B Path Element

Hier kann die Wahrscheinlichkeit nach der Art des Tags gewählt werden.

3.7.4 B Knoten "Query"

Die Kanten werden mit 1 beschriftet.

3.8 Implementierung

3.8.1 Invertierter Index

Alle Informationen, die zur Berechnung des Rankings benötigt werden, werden in einem invertierten Index abgelegt. Ein invertierter Index ist ein Index aller Terme aller Dokumente. Für jedes Vorkommen eines Terms in einem Dokument wird der vollständige Pfad bis zum Wurzelelement mit den Wahrscheinlichkeiten der Kanten abgelegt. Abbildung 3.8.1 zeigt einen Indexeintrag für den Term "XML".

Anhand eines Indexeintrags lassen sich einfach die INCLUDES-Strukturbedingungen überprüfen.

3.8.2 Queryausführung

Die Queryausführung läuft in 4 Schritten ab.

- Für jeden Queryterm werden alle Pfade aus dem Index gesucht, die die Strukturbedingungen erfüllen.

- Die bedingte Wahrscheinlichkeit entlang des Pfades bis zum Retrieval Element wird berechnet.
- Aufsummierung der Wahrscheinlichkeiten der Terme, die zum selben Retrieval Element gehören.
- Ausgabe der Retrieval Elements in Reihenfolge des Rankings.

3.9 Evaluierung

Ziel des Retrievalmodells war neben der neuen Funktionalität eine Verbesserung der Retrievalqualität gegenüber dem Dokumentretrieval. Um dies zu überprüfen wurde das Modell implementiert und auf einer Sammlung von SGML-strukturierten Patentdokumenten getestet. Dabei zeigte sich, das das Modell tatsächlich die Retrievalqualität verbessert.

Literatur

[1] Sung Hyon Myaeng, Dong-Hyun Jang, Mun-Seok Kim, and Zong-Cheol Zhoo. A flexible model for retrieval of SGML documents. *Research and Development in Information Retrieval (SIGIR)*, 1998.

[2] G. Salton, editor. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, 1971.