

Aufgabe 1

Gegeben sei eine Datenbank, die die Dokumente d1 bis d6 enthält und diese wie folgt durch die

Indexterme t1 bis t8 repräsentiert:

d1 : {t1, t4, t6, t7}

d2 : {t2, t4, t8}

d3 : {t1, t3, t4}

d4 : {t2, t6, t7}

d5 : {t1, t4}

d6 : {t1, t3, t6}.

a) Bestimmen Sie den zu dieser Datenbank gehörigen invertierten Index („Inverted File Index“).

Lösung: Inverted File Index:

t1 → d1, d3, d5, d6

t2 → d2, d4

t3 → d3, d6

t4 → d1, d2, d3, d5

t6 → d1, d4, d6

t7 → d1, d4

t8 → d2

b) Welche Treffer liefert das Boolesche Modell für die folgenden Anfragen?

„t1 AND t4“

„(t3 OR t4) AND NOT t6“

„(t1 AND t6) OR NOT t6“

Lösung:

„t1 AND t4“: {d1, d3, d5}

„(t3 OR t4) AND NOT t6“: {d2, d3, d5}

„(t1 AND t6) OR NOT t6“: {d1, d2, d3, d5, d6}

d) Geben Sie zu jeder der folgenden Treffermengen eine (möglichst kurze) Anfrage an, die die jeweilige Treffermenge liefert.

{d2}, {d6}, {d3, d5}

Anfrage für die Treffermenge {d2}:

t8; t2 AND t4; t2 AND t8; t4 AND t8; t8 AND NOT t1....

Anfrage für die Treffermenge {d6}:

t3 AND t6; t6 AND NOT t7; t3 AND NOT t4; t1 AND t3 AND t6....

Anfrage für die Treffermenge {d3, d5}:

t1 AND NOT t6...

Aufgabe 2

Es sollen $w_1, w_2, \dots$ unterschiedliche Wörter sein. Die Dokumente $d_1, \dots, d_5$ sollen folgende Wortfolgen darstellen:

$d_1 : w_5, w_1, w_9, w_3, w_8, w_2$

$d_2 : w_9, w_8, w_3$

$d_3 : w_2, w_3, w_8, w_7$

$d_4 : w_9, w_1, w_8, w_2, w_3$

(a) Welche Antwortmengen ergeben sich beim Booleschen Retrieval für folgende Anfragen:

$q_1 : w_2 \text{ AND } (w_8 \rightarrow w_1)$

$q_2 : w_7 \text{ OR } (w_1 \rightarrow \neg w_3)$

$q_3 : (w_8 \rightarrow w_2) \rightarrow (w_1 \rightarrow \neg w_3)$

$q_1 : \{d_1, d_4\}$

$q_2 : \{d_2, d_3\}$

$q_3 : \{d_2, d_3\}$

(b) Glauben Sie dass die Implikation in realen Systemen üblicherweise implementiert wird?

Nein. Zu einem steht die Implikation oft für den Ausdruck „wenn A, dann B“. Von diesem Sprachgebrauch wird normalerweise eine inhaltliche Zusammenhänge (Kausalität der zeitlichen Abfolge) erwartet. In Suchverfahren wird nur der wahrheitsfunktionale Zusammenhang berücksichtigt, d.h. die Implikation ist genau dann falsch, wenn der Wenn-Teil wahr ist und der Dann-Teil falsch ist. In jedem anderen Fall ist die Implikation wahr. Deswegen liefert die Anfrage „wenn A, dann B“ eine große Antwortmenge als Ergebnis, die viele nicht relevante Dokumente enthält. Zu anderen ist es möglich die Implikation aussagenlogisch äquivalent durch eine Kombination mit OR und NOT auszudrücken. Zu beachten ist aber, im Allgemeinen wird der NOT-Operator nur in Verbindung mit dem AND-Operator eingesetzt, um bestimmte Dokumente aus einer Ergebnismenge auszuschließen und nicht etwa eine ganze Menge (sehr groß) an eine NOT Bedingung auszuliefern.

Aufgabe 3

Schreiben Sie einen Algorithmus für das Zusammenführen zweier Postinglisten zur Auswertung einer OR-Query - analog dem AND-Algorithmus aus der Sitzung.

```
OR(p1,p2)
answer ← <>
while p1 ≠ NIL OR p2 ≠ NIL
if p1 ≠ NIL
    then if p2 ≠ NIL
        do if docID(p1) = docID(p2)
            then ADD(answer, docID(p1))
            p1 ← next(p1)
            p2 ← next(p2)
        else if docID(p1) < docID(p2)
            then ADD (answer, docID(p1))
            p1 ← next(p1)
            else ADD (answer, docID(p2))
            p2 ← next(p2)
        else ADD(answer, docID(p1))
        p1 ← next(p1)
    else if p2 ≠ NIL
        ADD(answer, docID(p2))
        p2 ← next(p2)
return answer
```

Aufgabe 4 wurde so nicht gestellt aber schauen Sie sich die Lösung einmal an.

Implementieren Sie folgenden ersten Teil eines sehr einfachen IR Systems.

- (a) Ein Teilprogramm soll den Shakespearertextkorpus von der Webseite in einzelne Stücke zerlegen, jedem Dokument eine ID zuordnen, den Text ausgeben und den Dokumententitel ausgeben können
- (b) Eine Methode für eine lineare Suchmöglichkeit über die Texte mit Rückgabe der Dokumenten-id
- (c) Implementieren Sie eine Klasse, die eine binäre Term-Dokumentenmatrix für einen Textkorpus wie er in a eingelesen wurde aufbaut.

Lösung siehe Vorschläge vom Kommunikationsinstitut Köln