Teilleistung 2

Information Retrieval 1 Wintersemester 2015/16

Gruppenmitglieder:
Christoph Keck (3115711)
Alexander Baumgärtner (1800210)
Philipp Wallhäuser (1592856)

Aufgabe 1

Vorgaben

D1 = 100
D2 = 50
Kollektion = 650.000
λ = 0.3
μ = 150

Maximum-Likelihood-Schätzung:

D1: Brauerei: 1/100 Bier: 3/100 Bamberg: 0

P(Q|D1) = (1/100) * (3/100) * 0
P(Q|D1) = 0

D2: Brauerei: 3/50 Bier: 2/50 Bamberg: 3/50

P(Q|D2) = (3/50) * (2/50) * (3/50)
P(Q|D2) = 0.000144

Kollektion gesamt: Brauerei: 2.000/650.00 Bier: 70.000/650.000
Bamberg: 90.000/650.000

P(Q|C) = (2.000/650.00) * (70.000/650.000) * (90.000/650.000)
P(Q|C) = 0.00004588

Jelinek-Mercer Glättung:

D1:
Brauerei: P'(q1|D1) = ((1 - 0.3) * (1/100)) + (0.3 * (2.000/650.000))

P’(q1|D1) = 0.0079

Bier: P’(q2|D1) = ((1 - 0.3) * (3/100)) + (0.3 * (70.000/650.000))
P’(q2|D1) = 0.0533

Bamberg: P’(q3|D1) = ((1 - 0.3) * 0) + (0.3 * (90.000/650.000))
P’(q3|D1) = 0.0415

Ranking-Bewertung: P’(Q|D1) = 0.0079 * 0.0533 * 0.0415
P’(Q|D1) = 0.000017474405

QL(Q,D1) = ln(0.0079) + ln(0.0533) + ln(0.0415)
QL(Q,D1) = -10.95

D2:
Brauerei: P’(q1|D2) = ((1-0.3) * (3/50)) + (0.3 * (2.000/650.000))
P’(q1|D2) = 0.0429

Bier: P’(q2|D2) = ((1 - 0.3) * (2/50)) + (0.3 * (70.000/650.000))
P’(q2|D2) = 0.0603

Bamberg: P’(q3|D2) = ((1 - 0.3) * (3/50)) + (0.3 * (90.000/650.000))
P’(q3|D2) = 0.0835

Ranking-Bewertung: P’(Q|D2) = 0.0429 * 0.0603 * 0.0835
P’(Q|D2) = 0.000216003645

QL(Q,D2) = ln(0.0429) + ln(0.0603) + ln(0.0835)
QL(Q,D2) = -8.44

Dirichlet-Glättung:

D1:
Brauerei: P’(q1|D1) = (1 + (150 * (2.000/650.000))) / (100 + 150)
P’(q1|D1) = 0.0058

Bier: P’(q2|D1) = (3 + (150 * (70.000/650.000))) / (100 + 150)
P’(q2|D1) = 0.0766

Bamberg: P’(q3|D1) = (0 + (150 * (90.000/650.000))) / (100 + 150)
P’(q3|D1) = 0.0831

QL(Q,D1) = ln(0.0058) + ln(0.0766) + ln(0.0831)
QL(Q,D1) = -10.21

D2:

Brauerei: P’(q1|D2) = (3 + (150 * (2.000/650.000))) / (50 + 150)
P’(q1|D2) = 0.0173

Bier: P’(q2|D2) = (2 + (150 * (70.000/650.000))) / (50 + 150)
P’(q2|D2) = 0.0908

Bamberg: P’(q3|D2) = (3 + (150 * (90.000/650.000))) / (50 + 150)
P’(q3|D2) = 0.1188

QL(Q,D2) = ln(0.0173) + ln(0.0908) + ln(0.1188)
QL(Q,D2) = -8.59

Aufgabe 2

Q = “Universität Bamberg”
N = 350.000
n1 = 5000
n2 = 3000
D = 800
dl/avdl = 0.8
fq,i jeweils = 1, da 2 Anfrageterme
fd,1 = 20
fd,2 = 40
k1 = 1.2, b = 0.75, k2 = 125
K = 1.2 * (0.25 + 0.75 * 0.8) = 0.85

BM25 (Q,D) =
ln ((350.000 - 5.000 + 0.5) / (5.000 + 0.5) * (((1.2 + 1)20) / (0.85 + 20)) * (((125+1)1) / (125 + 1)))
+
ln ((350.000 - 3.000 + 0.5) / (3.000 + 0.5) * (((1.2 + 1)40) / (0.85 + 40)) * (((125+1)1) / (125 + 1)))
= 4.89 + 5.37
= 10.26

Aufgabe 3

a) 4359054 → v-Byte: 00000010 00001010 00000111 10001110
2^22, 2^17, 2^15, 2^9, 2^8, 2^7, 2^3, 2^2, 2^1

b) 00000001 00110100 10010000 = 23056
2^14, 2^12, 2^11, 2^9, 2^4

Aufgabe 4

d-Gaps:
4, 5, 1, 9, 3, 7, 4, 6, 9, 13, 8, 2, 1, 1, 4, 9, 10, 1, 5, 5, 5, 3, 2, 1, 8, 7

Skip Pointer:
(19, 4) , (39, 8) , (71, 12) , (86, 16) , (107, 20) , (118, 24)

Eintrag ID 102:
Angefangen wird beim Skip Pointer (107,20). Die Dokumentnummer 107 geht dem Eintrag 20 voran und somit enthält der Eintrag 19 die Nummer 107. Deshalb wird die d-Gap beim Eintrag 19 betrachtet, welche eine 5 beschreibt. Somit muss der Eintrag 18 die gesuchte Dokumentnummer 102 enthalten. Der Start beim Skip Pointer (86,16) wäre ebenfalls möglich, jedoch würde sich die Dauer der Suche dann verlängern.

Aufgabe 5

Neben der Dokument-ID wäre die Anzahl der Wörter für das jeweilige Dokument ein sinnvoller Eintrag. Auch sinnvoll kann es sein zu speichern wie häufig ein Begriff vorkommt und an welcher Stelle im Dokument dieser auftaucht (wievieltes Wort im wievielten Satz). Neben einer positiven Liste in welcher das kontrollierte Vokabular aufgenommen wird, kann es auch sinnvoll sein eine negative Liste zu erstellen in welcher alle Stop-Worte aufgenommen werden, die bspw. besonders häufig vorkommen (Präpositionen etc.) und den Speicherbedarf unnötig erhöhen würden[1]. Ebenso kann es Sinn machen, Zeichenfolgen welche nicht mit in den Index aufgenommen werden sollen zu definieren (etwa Zahlenfolgen).

[1] http://www.iai.uni-bonn.de/III/lehre/vorlesungen/InformationRetrieval/WS04/Vorlesung-041021neu.pdf (S. 16)

Aufgabe 6

Bamberg: 1. d3: 50 2. d1: 20 3. d2: 10 4. d5: 10 5. d4: 0 6. d6: 0
→ Top 2 noch nicht stabil, da 7 * 2 + 2 * 1 + 1 * 2 = 18 noch mögliches Gewicht besteht.

Suche: 1. d3: 50 2. d1: 27 3. d4: 14 4. d6: 14 5. d2: 1 6. d5: 10
→ Top 2 stabil, da nur 2 * 1 + 1 * 2 = 4 noch mögliches Gewicht besteht. Die Top 2 kann dadurch nicht mehr verändert werden!

Aufgabe 7

#combine(#uw:6(Bamberg Universität) #od:1(Godehard Ruppert) Präsident)

Es werden die 3 Suchanfragen “Bamberg Universität” , “Godehard Ruppert” und Präsident kombiniert. Dabei darf der Abstand der beiden Suchbegriffe “Bamberg” und “Universität” maximal 6 betragen, jedoch ist die Reihenfolge der Vorkommen der Begriffe gleichgültig.

Die zweite Anfrage nach dem Präsidenten Godehard Ruppert setzt voraus, dass die beiden Suchbegriffe “Godehard” und “Ruppert” genau in dieser Reihenfolge vorkommen und nichts zwischen diesen beiden Begriffen stehen darf. Nach dem dritten Begriff “Präsident” wird ebenfalls gesucht. Somit müssen alle 3 Ereignisse kombiniert im Dokument eintreten (#combine).

Anfragebearbeitung

Es wird nach den Begriffskombinationen “Bamberg Universität”, “Godehard Ruppert” und “Präsident” im Dokument gesucht. Nur falls alle 3 Begriffe in gewünschter Ordnung vorhanden sind, so wird ein Ergebnis geliefert.
Über invertierte Listen werden den Begriffen die jeweiligen Dokumente zugeordnet.