Date:

20F-0288

i) No, For instance. The Boolean score is 1 when all the query terms occur in a body. Otherwise zero.

ii) Possible values: 0.2, 0.31, 0.49, 0.51, 0.8, 0.69, 1.0

iii)
```
ZONESCORE (list(q))
    float scores [N] = [0]
    constant g[l]
     P ← MERGE (List (q))
    MERGE function will merge the posting list.
     While (p != null) {
          Scores [doc ID[P]] = weightZone (P,g)
               P ← next (P)
     return scores }
```

iv)
```
weightZone (P1, P2, g)
          S ← ()
          Scores (doc ID (P1)) = 0
            for i ← 1 to l
                 S [i] ← BooleanScore (q, docID (P1))
            Scores (doc ID (P1)) = Scores (doc ID (p1)) +
                                   g[i] * S[i]
```

Date:

v) By using equation:
$g = 0.25$

vi) 1 1 R , 2 3/4 NR , 3 3/4 R , 4 0 NR
5 1 R , 6 3/4 R , 7 1/4 NR

vii) In cases where st(dt, qt) and sb(dt, qt) have same values, then score is independent and doesn't play role in optimizing g.

viii) $df_t \geq 1 \Rightarrow idf_t \leq \log N \Rightarrow$ idf always finite

ix) It is 0 if the word is available in every doc. It has the same effect of ignored if we put in stop list.

x)

terms	Doc 1	Doc 2	Doc 3
Car	44.55	6.6	39.6
Auto	6.24	68.64	0
Insurance	0	53.46	46.98
Best	21	0	25.5

xi) Yes

Date:

xiii) It is the no. of bits in boolean representation of idf.

xiv) If jealousy and jealous are stemmed to common word, then their tf's and idf's would be added together.

xv) Doc 1 = (0.897, 0.125, 0, 0.423)
Doc 2 = (0.076, 0.786, 0.613, 0)
Doc 3 = (0.595, 0, 0.706, 0.383)

xvi) Doc 1 = 0.999
Doc 2 = 0.999
Doc 3 = 0.999.

Because they are normalised unit vectors

xvii) Term weights =

i)

Term	Doc 1	Doc 2	Doc 3
car	0.125	0.786	0
Auto	0	0.613	0.786
insurance	0	0.113	0.706
best	0.423	0	0.383

Score 1 = 0.847, Score 2 = 0.689, Score 3 = 0.595
Ranking = 1, 2, 3

ii)

Term	Query	Idf, w(d,q)
Car	1.65	0.478
Auto	2.08	0.662
Insurance	1.62	0.47
best	1.5	0.43

Ranking = 2, 3, 1

Date:

xviii) $\sum_i (q_i - w_i)^2 = \sum_i q_i^2 - 2\sum q_i w_i + \sum w_i^2 = 2(1-\sum q_i w_i)$

Thus:

$\sum (q_i - v_i)^2 < \sum_i (q_i - w_i)^2 \Leftrightarrow 2(1-\sum q_i v_i) < 2(1-\sum q_i w_i) \Leftrightarrow \sum q_i v_i > \sum q_i w_i$

xix)

word	tf	wf	df	idf	qi	tf	wf	di	qi.di
digital	1	1	10000	3	3	1	1	0.52	1.56
videos	0	0	100000	2	0	1	1	0.52	0
cameras	1	1	50000	2.3	2.3	2	1.3	0.68	1.56

Similarity score = 1.56 + 1.56 = 3.12

xx) We can assign weights to query terms according to their idf in the collection

xxi) Omit this term from the query and proceed. Its contribution to the dot products with any document will be zero.

xxii) The weight contribution in any document vector = weights contribution by coyote + weight contribution by insurance = 0 + k. The idf contributed by coyote in query vector need not to be calculated as the idf weights for all the documents is zero coyotes