14/April/2024

Assignment-5

CSCE-5200

Uday Bhaskar Valapadasu
11696364

Ans:-

No. For instance, the boolean score from the title zone could be 1 when at least half of the query terms occur in that zone and zero otherwise whereas from the body zone, the boolean score is 1, when all query terms occur in the body and zero otherwise.

2 Ans:

The distinct score values a document can receive are combinations of these weights based the match occurs:

1) No Match : 0
2) Match in zone 1 only : 0.2
3) Match in zone 2 only : 0.31
4) Match in Zone 3 only : 0.49
5) Match in Zones 1 and 2 : 0.2 + 0.31 = 0.51
6) Match in Zones 1 and 3 : 0.2 + 0.49 = 0.69
7) Match in Zones 2 and 3 : 0.31 + 0.49 = 0.8
8) Match in all zones : 0.2 + 0.31 + 0.49 = 1.0

∴ possible values: 0.2, 0.31, 0.51, 0.69, 0.49, 0.8, 1.0

3 Ans:

The IDF of a term is always finite because it is calculated as the logarithm of the ratio of the total number of documents to the number of documents that contain the term, which prevents it from beginning infinite.

Even if a term appears in every document,

making the denominator equal to numerator, the log of 1 is 0, which is finite.

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

4 Ans:

It is 0. For a word that occurs in every document, putting it on the stoplist has the same effect as idf weighting: the word is ignored. The IDF of a term that occurs in every document is 0, because the log base 10 of $\frac{N}{N}$ (where N is the total number of documents) is 0. This comparison highlights the rationale behind using stopwords list; terms occur in almost every document (like "the", "is", "at") are not useful for distinguishing between documents in search queries, much like terms with an IDF of 0.

5 Ans:-

Yes, the tf-idf weight of a term in a document can exceed 1. This happens because the term frequency (tf) can be any non-negative integer, and if the term is sufficiently rare (thus having a high IDF), the product of tf and idf (tf-idf) can be greater than 1.

6 Ans:

	Doc1	Doc2	Doc 3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

Figure 6.9

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

Figure 6.8

The tf-idf weighting scheme assigns to term 't' a weight in document 'd' given by

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t$$

1) For the term 'car'

→ Doc 1 : $27 \times 1.65 = 44.55$

→ Doc 2 : $4 \times 1.65 = 6.6$

→ Doc 3 : $24 \times 1.65 = 39.6$

2.) For the term "auto":

→ Doc 1: 3 × 2.08 = 6.24

→ Doc 2: 33 × 2.08 = 68.64

→ Doc 3: 0 × 2.08 = 0

3.) For the term "insurance":

→ Doc 1: 0 × 1.62 = 0

→ Doc 2: 33 × 1.62 = 53.46

→ Doc 3: 29 × 1.62 = 46.98

4.) For the term "best":

→ Doc 1: 14 × 1.5 = 21

→ Doc 2: 0 × 1.5 = 0

→ Doc 3: 17 × 1.5 = 25.5

These calculations give you the tf-idf weights for the given terms across the three documents.

x x