1)

 a)

O stemming substitui as palavras pela sua forma base (como se trunca-se a palavra) pelo que, ao fazê-lo, poderá substituir por uma palavra que não tem o mesmo significado na query. Como substitui por uma palavra que pertence ao dicionário de palavras do sistema, acaba por aumentar o recall, pois este consegue retornar mais documentos relevantes, embora possa diminuir a precision, pois estes documentos retornados poderão não ser os mais relevantes para a query. Logo a afirmação é falsa.

 b)

O stemming substitui as palavras pela sua forma base (como se trunca-se a palavra) pelo que, ao fazê-lo, poderá substituir por uma palavra que não tem o mesmo significado na query. Como substitui por uma palavra que pertence ao dicionário de palavras do sistema, acaba por aumentar o recall, pois este consegue retornar mais documentos relevantes, embora possa diminuir a precision, pois estes documentos retornados poderão não ser os mais relevantes para a query. Logo a afirmação é verdadeira.

 c)

O stemming substitui as palavras pela sua forma base (como se trunca-se a palavra) pelo que, ao fazê-lo, irá manter ou diminuir o tamanho do vocabulário. Logo a afirmação é falsa.

 d)

De modo a melhorar os resultados do sistema de retrieval, o stemming poderá ser invocado tanto quando se indexa como quando se processa a query. Logo a afirmação é falsa.

2)

 a)

O idf (Inverted Document Frequency) analisa o quão rara uma palavra é na coleção. Uma palavra que apareça em todos os documentos da coleção terá um idf com valor 0.

As stop-words costumam aparecer em todos os documentos pelo que o seu valor de idf tenderá a ser 0 também.

 b)

O tf-idf dá-nos a importância de um termo num dado documento em função da raridade deste termo na coleção. Em termos matemáticos isto traduz-se em multiplicar-se o tf pelo idf, ou seja, tf (term frequency no documento) \* log(nº documentos / nº documentos em que o termo aparece). Assim, percebe-se que o valor de tf-idf poderá exceder 1.

3)

Although many people think York University is located in England or New York, it is actually located in Toronto

4)

 a)

Como os sistemas foram testados com apenas uma query, a MAP é igual a AP.

Sistema 1: (1/4) \* (1 + 2/3 + 3/9 + 4/15) = 17/30 = 0.57

Sistema 2: (1/4) \* (1/2 + 2/5 + 3/9 + 4/12) = 47/120 = 0.39

O sistema 1 tem um maior MAP.

 b)

Sim. Quanto mais tarde cada documento relevante for retornado, pior irá ser a precision e consequentemente a average precision e a mean average precision.

 c)

Sistema 1: 3/4

Sistema 2: 3/4

 d)

--

5)

 a)

--

 b)

precision = 1/5

recall = 1/2

f1 = 2 / (5 + 2) = 2/7

 c)

precision = 5/5 = 1

recall = 5/10 = 1/2

f1 = 2 / (1 + 2) = 2/3

7)

 a)

 Doc1 Doc2 Doc3

 tf idf tf-idf tf idf tf-idf tf idf tf-idf

car 27 1.62 43.74 4 1.62 6.48 24 1.62 38.88

auto 3 2.05 6.15 33 2.05 67.65 0 2.05 0

insurance 0 1.59 0 33 1.59 52.47 29 1.59 46.11

best 14 1.47 20.58 0 1.47 0 17 1.47 24.99

 b)

Doc1 = (q \* d1) / (||q|| \* ||d1||) = (1 \* 6.15) + (0) / (...)

Doc2 = (q \* d2) / (||q|| \* ||d2||) = (1 \* 67.65) + (1 \* 52.47) / (...)

Doc3 = (q \* d3) / (||q|| \* ||d3||) = (0) + (1 \* 46.11) / (...)

8)

 a) É pelo tf-idf e cosseno

 A B

this 1 0

update 1 0

is 1 0

designed 1 0

to 1 1

reduce 1 0

rankings 1 1

for 2 1

low-quality 1 0

sites 3 2

which 1 0

are 2 0

low-value 1 0

add 1 0

users 1 0

copy 1 0

content 1 0

from 1 0

other 1 0

websites 1 0

or 1 0

that 1 1

just 1 0

not 1 0

very 1 0

useful 1 0

we 0 1

can't 0 1

make 0 1

a 0 1

major 0 1

improvement 0 1

without 0 1

affecting 0 1

many 0 1

it 0 1

has 0 1

be 0 1

some 0 2

will 0 2

go 0 2

up 0 1

and 0 1

down 0 1

 b)

A: (1 \* 0 + 1 \* tf-idf) / (sqrt(2) \* sqrt(30))

B: (1 \* 0 + 1 \* tf-idf) / (sqrt(2) \* sqrt(27))

 c)

O tf-idf dá-nos a importância de um termo num dado documento em função da raridade deste termo na coleção. Em termos matemáticos isto traduz-se em multiplicar-se o tf pelo idf, ou seja, tf (term frequency no documento) \* log(nº documentos / nº documentos em que o termo aparece).

9)

d1: Jackson talented entertainers all time

d2: Michael Jackson anointed himself King Pop

 d1 d2 c p(t|d1) p(t|d2) p(t|c)

Jackson 1 1 2 1/4 1/6 2/10

talented 1 0 1 1/4 0 1/10

entertainers 1 0 1 1/4 0 1/10

time 1 0 1 1/4 0 1/10

Michael 0 1 1 0 1/6 1/10

anointed 0 1 1 0 1/6 1/10

himself 0 1 1 0 1/6 1/10

King 0 1 1 0 1/6 1/10

Pop 0 1 1 0 1/6 1/10

 a)

--

 b)

d1: (0.5 \* 0 + 0.5 \* 1/10) \* (0.5 \* 1/4 + 0.5 \* 2/10)

d2: (0.5 \* 1/6 + 0.5 \* 1/10) \* (0.5 \* 1/6 + 0.5 \* 2/10)

 c)

d1: ((0 + 100 \* 1/10) / (4 + 100)) \* ((1 + 100 \* 2/10) / (4 + 100))

d2: ((1 + 100 \* 1/10) / (6 + 100)) \* ((1 + 100 \* 2/10) / (6 + 100))

12)



13)



15)

 a)

 tf

the 2

martian 1

has 1

landed 1

on 1

latin 1

pop 1

sensation 1

ricky 1

martin 1

P(the) = 2/11 ; P(martian) = 1/11

 b)

 tf

the 1

martian|the 1

has|martian 1

landed|has 1

on|landed 1

the|on 1

latin|the 1

pop|latin 1

sensation|pop 1

ricky|sensation 1

martin|ricky 1

P(sensation|pop) = 1/11 ; P(pop|the) = 0

16)

 d1 d2 d3 d4 col

click 4 2 0 1 7

go 1 0 0 0 1

the 1 0 0 0 1

shears 1 0 0 1 2

boys 1 0 0 0 1

metal 0 0 1 1 2

here 0 0 1 1 2

Total 8 2 2 4 16

 a)

d1 d2 d3 d4

click 0.46875 0.71875 0 0.34375 d2 > d1 > d4 > d3

shears 0.125 0 0 0.1875 d4 > d1 > d2/d3

Final 0.05859 0.71875 0 0.06445 d2 > d4 > d1 > d3

 b)

d2 > d4 > d1 > d3

17)

 a)

É usado no LMJM. Tem um efeito não escalado. É o número de vezes que o termo ocorre no documento.

 b)

É usado no LMJM. Tem um efeito não escalado. É o número de vezes que o termo ocorre na coleção.

 c)

Não é usado no LMJM. É o número de documentos que contêm o termo.

 d)

--

19)

 a)

qm = alfa \* q0 + beta \* (1/|Dr|) \* sum(d€Dr) - gamma \* (1/|Dnr|) \* sum(d€Dnr)

qm = 1\*q0 + 0.75\*docA - 0.25\*docB (>= 0)

 q0 docA docB qm

this 0 1 0 0.75

update 0 1 0 0.75

is 0 1 0 0.75

designed 0 1 0 0.75

to 0 1 1 0.5

reduce 0 1 0 0.75

rankings 0 1 1 0.5

for 0 2 1 1.25

low-quality 0 1 0 0.75

sites 1 3 2 2.75

which 0 1 0 0.75

are 0 2 0 1.5

low-value 0 1 0 0.75

add 0 1 0 0.75

users 0 1 0 0.75

copy 0 1 0 0.75

content 0 1 0 0.75

from 0 1 0 0.75

other 0 1 0 0.75

websites 0 1 0 0.75

or 0 1 0 0.75

that 0 1 0 0.75

just 0 1 0 0.75

not 0 1 0 0.75

very 0 1 0 0.75

useful 0 1 0 0.75

we 0 0 1 0

can't 0 0 1 0

make 0 0 1 0

a 0 0 1 0

major 0 0 1 0

improvement 0 0 1 0

without 0 0 1 0

affecting 0 0 1 0

many 0 0 1 0

it 0 0 1 0

has 0 0 1 0

be 0 0 1 0

and 0 0 1 0

some 0 0 2 0

will 0 0 1 0

go 0 0 1 0

 b)

Como o algoritmo Rocchio é baseado no feedback do utilizador, poderá originar alguns problemas como:

 - assumir que o utilizador tem conhecimento suficiente na query inicial: o utilizador poderá não ser muito preciso no que deseja e também poderá enganar-se a escrever ou mesmo confundir termos;

 - assumir que os protótipos de relevância são "bem comportados": os termos nos documentos relevantes são todos semelhantes e os termos dos não relevantes são semelhantes também entre si, sendo os termos que aparecem nos relevantes todos diferentes dos que aparecem nos não relevantes.

20)

Quais as formas de obter relevance judgements, coleção dos dados, (ver capítulo avaliação). Que tipo de métricas usar? Como arranjar queries? Como criar um ambiente de teste? Como obter relevance judgments? – ter várias pessoas a responder, mas isso implica gastar mais dinheiro portanto ou cortamos no nº de queries ou no nº de docs a avaliar. Como escolher o nº de docs a avaliar?

Temos 100 queries e para cada uma 100 documentos e vamos avaliar com utilizadores. Vamos usar todas as 100 queries ou não? Os relevance judgments podem não sair com qualidade porque a query pode ser ambígua – o agreement deve ser superior a 60% para a query ser aceitável e 80% para ser efetivamente boa (só usamos essas)

21)

CombSum: use score of the document on the different lists as the main ranking score



CombMNZ: multiplies the number of ranks where the document occurs by the sum of the scores obtained across all lists.



BordaFuse: for each rank, the document gets a score corresponding to its (inverse) position on the rank. The fused rank is based on the sum of all per-rank scores.

Reciprocal Rank Fusion (RRF): the reciprocal rank fusion weights each document with the inverse of its position on the rank.

