CS 533 - Information Retrieval Systems
Class Notes (31.03.2014 - 02.04.2014)

prepared by Selcuk Emre Solmaz

1. Term Weighting Components

- Term frequency component (TFC)
- Collection frequency component (CFC)
- Normalization component (NC)

1.1. Term frequency component (TFC)

b : Binary (0, 1)

t : Row frequency (t_f)

n : Augmented normalization term frequency : $0,5 + 0,5\frac{t_f}{max(t_f)}$

Example:

document vector = (5 0 1 2)

b = (1 0 1 1)

t = (5 0 1 2)

n = (1 0 0,6 0.7)

1.2. Collection frequency component (CFC)

x: No change (use original TFC)

f : Inverse collection frequency component

$$ln\frac{m}{t_{g_j}} + 1$$

(m: number of documents, t_{g_j} : term generality for term *j* (no of unique docs. containing term *j*))

p: probability of inverse collection frequency factor

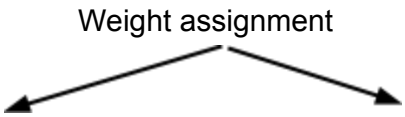
$$ln\frac{m-t_g+1}{t_g}$$

1.3. Normalization component (NC)

x: No change (use the weight implied by TFC and CFC)
c: use normalization

$$c = \frac{1}{\sqrt{\sum w_i^2}}$$

2. Different term weight combinations

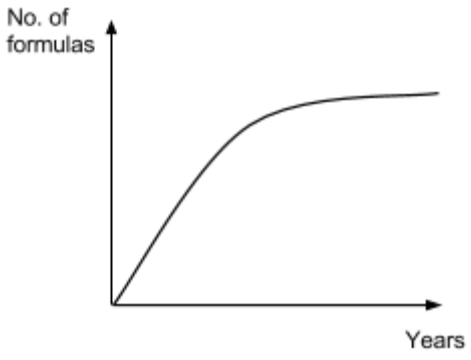


document vector			query vector		
b, t, n	TFC	3	b, t, n	TFC	3
x, f, p	CFC	3	x, f, p	CFC	3
x, c	NC	2	x	NC	1*

* Normalizing or not normalizing do not change document ranking.

3 x 3 x 2 x 3 x 3 x 1 = 162

In the literature there were 1800 (unique one 287) different matching functions.



In Salton's paper [1], they used several different text collections (ACM, TIMES, INSPEC, NPC, 12864 docs (largest at that time)). They found the best weighting combination is tfc.nfx <doc.query>.

Example:

						max t_f
$D =$	2	0	1	2	0	2
	0	2	1	3	1	3
	2	0	0	1	1	2
	1	0	0	0	1	1
	2	1	0	1	0	2

t_g:	4	2	2	4	3

Consider d_1: use tfc

TFC -> t => (2 0 1 2 0)

CFC -> $f : ln\frac{m}{t_{g_j}} + 1$

d_1 ->	f	1,22	1,92	1,92	1,22	1,51
d_1 ->	t.f	2x1,22	0x1,92	1x1,92	2x1,22	0x1,51

NC -> $c : \frac{1}{\sqrt{w_i^2}} = \frac{1}{\sqrt{2{,}44^2 + 0 + 1{,}92^2 + 2{,}44^2 + 0}} = \frac{1}{3{,}95}$

d_1 ->	t.f.c	$\frac{2{,}44}{3{,}95}$	0	$\frac{1{,}92}{3{,}95}$	$\frac{2{,}44}{3{,}95}$	0
d_1 ->	t.f.c	0,62	0	0,49	0,62	0

Query weight : n f x

example query: Q : (1 0 0 2 0)

Q:

TFC	n	$0,5 + 0.5\frac{1}{2}$	0	0	$0,5 + 0.5\frac{2}{2}$	0
	n	0,75	0	0	1	0
CFC	f	1,22	1,92	1,92	1,22	1,51
	nf	0,75x1,22	0	0	1x1,22	0
	nf	0,92	0	0	1,22	0

D matrix using tfc for doc weights would be like this:

$D \approx$

0,62	0	0,49	0,62	0
0	0,66	0,33	0,63	0,26
0,78	0	0	0,39	0,48
0,63	0	0	0	0,78
0,73	0,58	0	0,37	0

Sim(q,d_1) = 0,62 x 0,92 + 0 x 0 + 0,49 x 0 + 0,62 x 1,22 + 0 x 0 = 1,33
Sim(q,d_2) = 0,77
Sim(q,d_3) = 1,20
Sim(q,d_4) = 0,58
Sim(q,d_5) = 1,12

Ranking : $d_1 > d_3 > d_5 > d_2 > d_4$

3. Questions

1) In previous example, the t.f.c calculation of d_1 is given. Make the t.f.c calculation for d_2.

Answer:

Consider d_2: use tfc

TFC -> t => (0 2 1 3 1)

CFC -> $f : ln\frac{m}{t_{g_j}} + 1$

d_2 ->	f	1,22	1,92	1,92	1,22	1,51
d_2 ->	t.f	0x1,22	2x1,92	1x1,92	3x1,22	1x1,51

NC -> $c : \frac{1}{\sqrt{w_i^2}} = \frac{1}{\sqrt{0 + 3{,}84^2 + 1{,}92^2 + 3{,}66^2 + 1{,}51^2}} = \frac{1}{5{,}84}$

d_2 ->	t.f.c	0	$\frac{3,84}{5,84}$	$\frac{1,92}{5,84}$	$\frac{3,66}{5,84}$	$\frac{1,51}{5,84}$
d_2 ->	t.f.c	0	0,66	0,33	0,63	0,26

2) Calculate TFC value for query (0 0 1 1 5)

TFC	n	0	0	$0,5 + 0.5\frac{1}{5}$	$0,5 + 0.5\frac{1}{5}$	$0,5 + 0.5\frac{5}{5}$
	n	0	0	0,60	0,60	1

3) Calculate CFC values for query (0 0 1 1 5) for d_1 in the previous example.

CFC	f	1,22	1,92	1,92	1,22	1,51
	nf	0	0	0,60x1,92	0,60x1,22	1x1,51
	nf	0	0	1,15	0,73	1,51

4) Show D matrix using tfc for document weights.

$D \approx$

0,62	0	0,49	0,62	0
0	0,66	0,33	0,63	0,26
0,78	0	0	0,39	0,48
0,63	0	0	0	0,78
0,73	0,58	0	0,37	0

5) Calculate and rank the similarities between the query and the documents.
Sim(q,d_1) = 0,62 x 0 + 0 x 0 + 0,49 x 1,15 + 0,62 x 0,73 + 0 x 1,51 = 1,02

$Sim(q,d_2) = 0 \times 0 + 0{,}66 \times 0 + 0{,}33 \times 1{,}15 + 0{,}63 \times 0{,}73 + 0{,}26 \times 1{,}51 = 1{,}23$
$Sim(q,d_3) = 0{,}78 \times 0 + 0 \times 0 + 0 \times 1{,}15 + 0{,}39 \times 0{,}73 + 0{,}78 \times 1{,}51 = 1{,}46$
$Sim(q,d_4) = 0{,}63 \times 0 + 0 \times 0 + 0 \times 1{,}15 + 0 \times 0{,}73 + 0{,}78 \times 1{,}51 = 1{,}18$
$Sim(q,d_5) = 0{,}73 \times 0 + 0{,}58 \times 0 + 0 \times 1{,}15 + 0{,}37 \times 0{,}73 + 0 \times 1{,}51 = 0{,}27$

Ranking : $d_3 > d_2 > d_4 > d_1 > d_5$

4. Related Papers

- Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. http://dl.acm.org/citation.cfm?id=54260
- Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. http://dl.acm.org/citation.cfm?id=291008

5. References

[1] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* 24, 5 (August 1988), 513-523.