Qtd de

IDF = log (N/DF)

WTF → por documento → log TF + 1

Soma de palavras (Por palavra)

↳ Por documento → IDF · WTF + IDF · WTF₂

TF.IDF = (log TF + 1) · IDF

Vetorial

↳ V com (IDF₁, IDF₂)

V doc 1 (TF.IDF₁, TF.IDF₂)

↳ modulo $\sqrt{(TF.IDF_1)^2 + (TF.IDF_2)^2}$

normalizado = (TF.IDF₁ / modulo , TF.IDF₂ / modulo)

V com normalizado 1 entrada mesma V com normal 2 entrada 2

Com X = IDF₁ · V DOC 1 , IDF₂ · V doc 2

TF e IDF

Em ambos os modelos vistos no módulo 1, matriz termo-documento e índice invertido, as buscas de resultados ocorrem pelo modelo booleano.

Nesta busca posso ter situações conhecidas como Banquete - Inanição (Fome), que são o retorno de um número muito amplo de resultados ou a absoluta falta de resultados.

Por exemplo:

Para a matriz Termo-documento

Matriz Termo-Documento

	Doc1	Doc2	Doc3
alheio	0	1	0
bom	0	0	1
errado	1	1	1
gente	1	1	0

A busca "gente OR bom" retorna todos os documentos.

A busca "gente AND bom" retorna nenhum documento.

Para tentar resolver esta situação cientistas da computação propuseram criar pesos para os termos do dicionário, baseados na ideia de que algumas palavras devem possuir mais relevância que outras devido as suas ocorrências nos documentos e no Corpus.

Duas abordagens são elaboradas:

TF, W

DF, IDF

TF – W

A primeira leva em consideração a frequência de cada termo nos documentos, chamada de TF (termo, documento) (term frequency / frequência do termo). A frequência de termo TF(t,d) do termo t em um documento d é definido como o número de vezes que t ocorre em d;

Por exemplo:

Para o documento doc01:"O gatão, a gatona, o gato, a gata e seus gatinhos comeram muita ração." temos os termos gato e ração com as frequências de 5 e 1 respectivamente.

Neste caso a TF(gato,doc01)=5 e TF(ração,doc01)=1

Com certeza neste documento gato possui mais importância que ração. Mas considerar que possui 5 vezes mais importância é exagerado.

A frequência de termo literal não é o que queremos:

- A relevância não aumenta proporcionalmente com a frequência de termo.

Para amenizar esta diferença foi proposto o TF ponderado

Conhecido como W(t, d) = (IF TF(t,d) > 0, 1 + $\log_{10}$ TF(t,d), 0)

IF frequência: 0 → 0, 1 → 1, 2 → 1.3, 10 → 2, 1000 → 4, etc.

Pontuação para um documento em consulta: soma dos W(t,d) para todos os termos t da consulta em d:

Logs básicos:

Log 0 não existe; Log 1 = 0; Log 2 = 0,30; Log 3 = 0,48;

Log 4 = 0,60; Log 5 = 0,70; Log 6 = 0,78; Log 7 = 0,84;

Log 8 = 0,90; Log 9 = 0,95; Log 10 = 1; Log 100 = 2; Log 1000 = 3.

Calculando W(t,d) para os termos da matriz Termo-Documento

	Doc1	Doc2	Doc3
alheio	0	15	0
apressado	145	0	0
bom	0	0	1231
errado	12	8	120
gente	338	155	0

Trocando as frequencias pelo W(t,d)

$W(t, d) = (IF\ \ TF(t,d) > 0, 1 + \log_{10} TF(t,d), 0)$

	Doc1	Doc2	Doc3
alheio	0	1+log 15 2,18	0
apressado	1+log145 3.16	0	0
bom	0	0	1+log1231 4,09
errado	1+log12 2,08	1+log8 1,90	1+log120 3,08
gente	1+log338 3,53	1+log155 3,19	0
Consulta: errado Rank	2.08 2o	1,90 3o	3.08 1o
Consulta: errado gente Rank	2,08+3,53 5,61 1o	1,90+3,19 5,09 2o	3,08+0 3,08 3º

No livro indico ler as seções 6.2, 6.2.1 e 6.2.2.

Exercicio:

Calcule W(t,d) para os termos da matriz Termo-Documento
$W(t, d) = (IF\ \ TF(t,d) > 0, 1 + \log_{10} TF(t,d), 0)$

	Doc1	Doc2	Doc3
Alheio	7	45	340
apressado	1145	540	30
Bom	50	606	4123
Errado	31	0	420
Gente	15	15	400
Consulta: Alheio bom Rank:			
Consulta: Errado gente apressado Rank:			

IDF, Sacola de Palavras, TF.IDF

A segunda abordagem leva em consideração a raridade de um termo no Corpus. Considera-se que quando consultamos um termo raro no Corpus, os documentos que o contém devam possuir maior importância.

Primeiro criou-se o DF (Document Frequency) que indica em quantos documentos o termo aparece, DF (termo). Este valor indica que quanto menor ele for maior será sua raridade.

Para que tenhamos um índice que represente a raridade como fator de grandeza, criou o IDF (inverse DF) cuja fórmula é IDF(termo)= log N/DF(t), onde N é o número total de documentos no Corpus.

Calculando IDF(t) para os termos da matriz Termo-Documento, considerando nosso Corpus com 3 documentos.

IDF(termo)= log N/DF(t)

	DF	Doc1	Doc2	Doc3
alheio	1	0	15	0
apressado	1	145	0	0
bom	1	0	0	1231
errado	3	12	8	120
gente	2	338	155	0

	IDF	Doc1	Doc2	Doc3
alheio	Log 3/1 = 0,48	0	15	0
apressado	0,48	145	0	0
bom	0,48	0	0	1231
errado	Log 3/3 = 0	12	8	120
gente	Log 3/2 = 0,18	338	155	0
Errado		0	0	0
Errado gente		0+0,18 0,18	0+0,18 0,18	0+0 0

Sacola de palavras

Com estes dois parâmetros foi possível se trabalhar com uma nova estrutura de dicionário e de busca.

Esta estrutura é chamada de sacola de palavras e se assemelha muito ao índice invertido, porém possui dois campos adicionais. O primeiro é um IDF(t) calculado para cada termo e ligado ao termo e o segundo é um W(t,d) calculado para cada posting e ligado a ele.

Ao se fazer uma busca, selecionamos apenas os termos da busca e as listas a eles associados.

BUSCA RANQUEADA

Temos então 3 formas de ranquear os documentos constantes nestas listas:

Pelo IDF: sendo o peso de cada documento o resultado da somatória dos IDFs de todos os termos em que estão.

Pelo W: sendo o peso de cada documento dado pela somatória dos W de todos os postings daquele documento. Visto anteriormente.

Pelo TF.IDF (W.IDF): sendo a somatória de todos os W dos postings multiplicado pelo IDF do termo.

Nos tópicos anteriores de TF e IDF aplicamos o ranqueamento dos dois primeiros tipos.

Calcule os ranqueamentos para a tabela abaixo, usando tf.idf.

	IDF	Doc1	Doc2	Doc3
alheio		0	15	320
apressado		145	30	0
bom		220	0	1231
errado		12	8	120
gente		338	155	20
Consulta: bom errado Rank				
Consulta: alheio gente apressado Rank				

Tente ranquear os exemplos vistos anteriormente.

Veja o vídeo sobre Sacola de palavras https://www.youtube.com/watch?v=IRKDrrzh4dE

Modelo Vetorial

Neste modelo de RI usa-se o TF.IDF como principal parâmetro de ranqueamento, mas de forma diferenciada da sacola de palavras.

Pensou-se em poder classificar o grau de proximidade de um documento com a consulta feita.

O modelo proposto usa a ideia de vetores e cosseno, onde quando o cosseno entre dois vetores é maior implica que o ângulo entre eles é menor, significando que são mais similares.

Neste modelo temos:

- O plano de coordenadas é formado por eixos, em que cada eixo representa os valores de TF.IDF de cada termo constante na consulta;
- Os documentos são os vetores da intersecção de todas as coordenadas dos termos neste plano de coordenadas;
- A consulta é um vetor dos IDFs dos termos da consulta;
- O grau de similaridade dos vetores com a consulta é dado pelos cossenos dos ângulos entre estes vetores. Quanto maior o cosseno, maior a similaridade. Este método é conhecido como Cosseno de Similaridade.

Então, para ranquearmos um conjunto de documentos em relação a uma consulta:

- Para cada documento calculamos seu vetor;
- Para a consulta calculamos seu vetor;
- Calculamos os módulos dos vetores dos documentos e da consulta;
- Normalizamos os vetores dos documentos e da consulta;
- Calculamos o cosseno de cada par consulta-documento;
- Ordenamos estes cossenos em ordem decrescente, classificando os documentos.

Exemplificando:

$W(t, d) = (IF\ \ TF(t,d) > 0, 1 + \log_{10} TF(t,d), 0)$

$IDF(termo) = \log_{10} N/DF(t)$

	Doc1	Doc2	Doc3
alheio	0	15	0
apressado	145	3	200
bom	30	200	1231
errado	12	8	120
gente	338	155	0

Logs básicos:

Log 0 não existe; Log 1 = 0; Log 2 = 0,30; Log 3 = 0,48;

Log 4 = 0,60; Log 5 = 0,70; Log 6 = 0,78; Log 7 = 0,84;

Log 8 = 0,90; Log 9 = 0,95; Log 10 = 1; Log 100 = 2; Log 1000 = 3.

Ranqueando a consulta bom, errado

Calculando TF, IDF e TF.IDF, N = 10, N é o número de documentos do Corpus.

$IDF(termo) = \log_{10} N/DF(t)$

$W(t, d) = (IF\ \ TF(t,d) > 0, 1 + \log_{10} TF(t,d), 0)$

	IDF	TF.IDF		
		Doc1	Doc2	Doc3
bom	0,52	1,29	1,72	2,13
errado	0,52	1,08	0,99	1,60

Calculando vetores dos documentos

Vdoc1 = (1,29, 1,08)

Vdoc2 = (1,72, 0,99)

Vdoc3 = (2,13, 1,60)

Calculando vetor de consulta

Vcons = (0,52, 0,52)

Calculando módulos

Mdoc1=(1.29^2 + 1,08^2)1/2 =1,66+1,17=2,83 = 1,68

Mdoc2= (2,96+0,98)^0,5=3,94^0,5 = 1.98

Mdoc3= (4,54+2,56)^0,5=7,10^0,5 = 2.66

Mcons=(0,27+0,27)^0,5=0,54^0,5= 0,73

Normalizando vetores

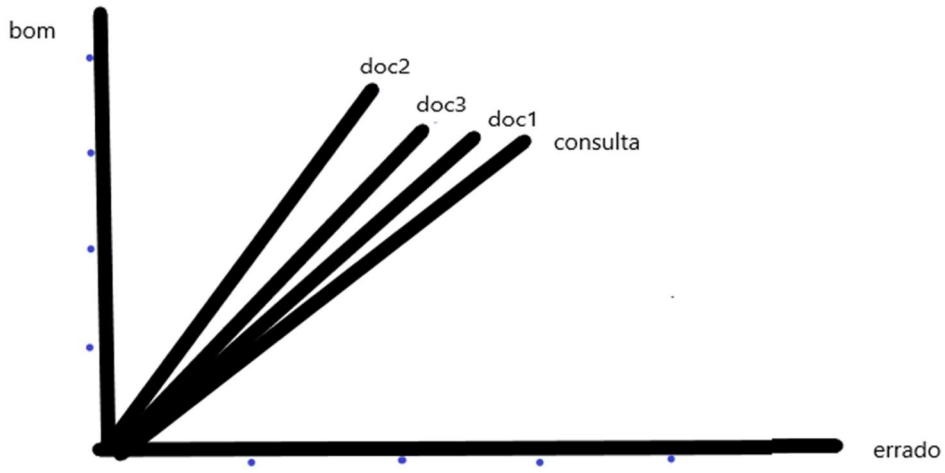
Vdoc1n=(1,29/1,68, 1,08/1,68)= (0,77, 0,64)

Vdoc2n=(1,72/1,98, 0,99/1,98)= (0,86, 0,49)

Vdoc3n=(2,13/2,66, 1,60/2,66)=(0,80, 0,60)

Vconsn=(0,52/0,73, 0,52/0,73)=(0,71, 0,71)

Visualizando os vetores



Calculando os cossenos de cada par consulta-documento

Cosdoc1cons=0,77*0,71+0,64*0,71= 0,546+0,454=1,00

Cosdoc2cons=0,86*0,71+0,49*0,71=0,61+0,35=0,96

Cosdoc3cons=0,80*0,71+0,60*0,71=0,568+0,426=0,994

Ranqueando

1º doc1, 2º doc3, 3º doc2

Exercício:

Dados os TFs.

$W(t, d) = (IF\ \ TF(t,d) > 0, 1 + \log_{10} TF(t,d), 0)$

$IDF(termo) = \log_{10} N/DF(t) \quad N=10$

	IDF	Doc1	Doc2	Doc3	Doc4
alho		0	12	344	4544
cebola		23	0	32	332
feijao		323	230	30	223
pepino		223	333	5545	0
rabanete		323	230	0	0

Calcule os ranqueamentos usando o modelo vetorial para as consultas:

alho,cebola

pepino,rabanete

alho, feijão,rabanete

Mais um exemplo resolvido:

Calcule o ranqueamento para a consulta apressado, gente para os dados abaixo, N=20:

$W(t, d) = (IF\ \ TF(t,d) > 0, 1 + \log_{10} TF(t,d), 0)$

$IDF(termo) = \log_{10} N/DF(t)$

	IDF	Doc1	Doc2	Doc3
apressado	Log 20/3 0.82	TF 145 W 3,16 TF.IDF 2.59	TF 3 W 1,47 TF.IDF 1,20	TF 200 W 3,30 TF.IDF 2,70
gente	Log 20/2 1	TF 338 W 3,52 TF.IDF 3,52	TF 155 W 3,19 TF.IDF 3.19	TF 0 W 0 TF.IDF 0

Calcular vetores dos documentos vetor(tf.idf termo1, tf.idf termo2)

Vdoc1=(2.59, 3.52)

Vdoc2=(1.20, 3.19)

Vdoc3=(2.70, 0)

Calcular vetor de consulta

Cons=(0.82, 1)

Calcular módulos

Mdoc1=(2.59^2 +3.52^2)^0.5 = (6.7 + 12.39)^0.5= 19.09=4.36

Mdoc2=(1,44+ 10.17)^0.5 = 11.61^0.5 = 3.40

Mdoc3=(7.29 + 0)^0.5 = 7.29^0.5 = 2.70

Mcons=(0.67 +1)^0.5= 1.67^0.5 = 1.29

Normalizar vetores

Vdoc1n=(2.59/4.36, 3.52/4.36)=(0.59, 0.80)

Vdoc2n=(1.20/3.40, 3.19/3.40)=(0.35 , 0.93)

Vdoc3n=(2.70/2.70, 0/2.70)=(1, 0)

Consn=(0.82/1.29, 1/1.29)=(0.63, 0.77)

Calcular os cossenos de cada par consulta-documento

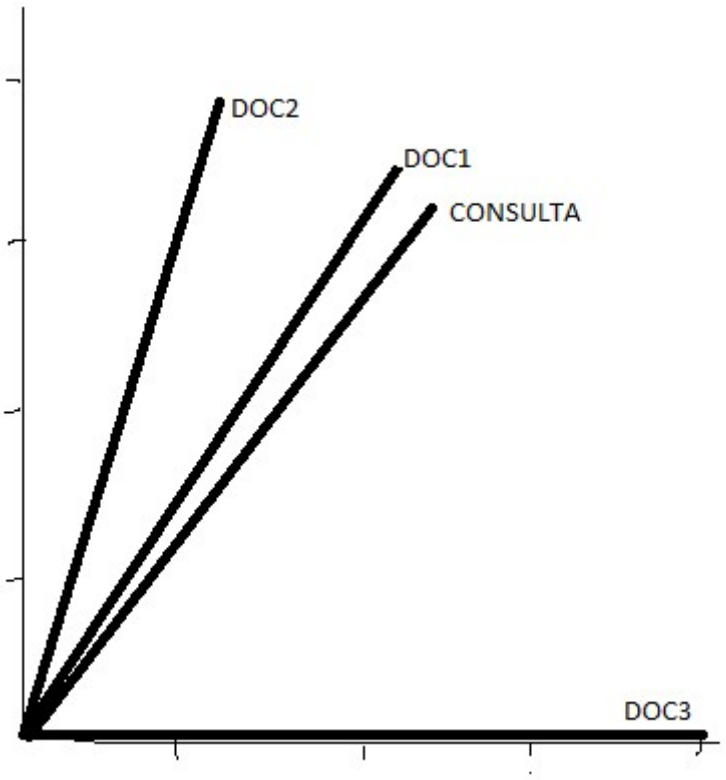
Cos(vdoc1n, consn)=0,59*0,63 + 0,80*0,77= 0,37 + 0,61=0,98

Cos(vdoc2n, consn)=0,35*0,63 + 0,93*0,77= 0,22+0,71=0,93

Cos(vdoc3n, consn)=1*0,63 + 0*0,77= 0,63 +0= 0,63

Fazer ranqueamento

1º Doc1, 2º doc2, 3º doc3



Mais exercícios:

$W(t, d) = (IF\ \ TF(t,d) > 0, 1 + \log_{10} TF(t,d), 0)$

$IDF(termo) = \log_{10} N/DF(t)$

	IDF	Doc1	Doc2	Doc3	Doc4
alheio	0,30	0	15	0	220
apressado	0	145	3	200	54
bom	0,12	30	200	1231	0
errado	0	12	83	120	400
gente	0,30	338	155	0	0

Calcule os ranqueamentos usando o modelo vetorial para as consultas:

A – alheio, gente

B – apressado, errado

C – alheio, bom, gente