1.Cvičení

1.Príklad

Nájdite dotaz v tvare [vyraz_1 vyraz_2] (bez úvodzoviek), ktorý Google neinterpretuje ako súčin jednotlivých výrazov dotazu. Tzn. dotaz, keď bude položený Googlu, vráti aspoň jeden výsledok, ktorý obsahuje len jeden výraz z dotazu.

a) Prvých 20 výsledkov označte číslami 0, 1, 2 podľa toho, či výsledok obsahuje oba výrazy, len jeden z výrazov alebo ani jeden výraz.

b) Podľa tohto usúďte, či Google interpretuje všetky dotazy implicitne ako Booleovský prienik výrazov z dotazu.

Řešení: Řešení si každý musí vyzkoušet sám.

2. Príklad

Vytvorte invertovaný index, zostavený pre následujúcu kolekciu dokumentov:

Doc 1 new home sales top forecasts
Doc 2 home sales rise in july
Doc 3 increase in home sales in july
Doc 4 july new home sales rise

Řešení: napíšeme tabulku a za každé slovo napíšeme ve kterých dokumentech se vyskytuje.

new	1,4
home	1,2,3,4
sales	1,2,3,4
top	1
forecasts	1
rise	2,4
in	2,3
july	2,3,4
increase	3

3. Príklad

Nižšie je časť indexu s pozíciami v tvare term: doc1: <pos1, pos2, pos3, ..>; doc2<pos1, pos2, ..>

angels: 2: <36,174,252,651>; 4: <12,22,102,432>; 7: <17>;
fools: 2: <1,17,74,222>; 4: <8,78,108,458>; 7: <3,13,23,193>;
fear: 2: <87,704,722,901>; 4: <13,43,113,433>; 7: <18,328,528>;
in: 2: <3,37,76,444,851>; 4: <10,20,110,470,500>; 7: <5,15,25,195>;
rush: 2: <2,66,194,321,702>; 4: <9,69,149,429,569>; 7: <4,14,404>;
to: 2: <47,86,234,999>; 4: <14,24,774,944>; 7: <199,319,599,709>;
tread: 2: <57,94,333>; 4: <15,35,155>; 7: <20,320>;
where: 2: <67,124,393,1001>; 4: <11,41,101,421,431>; 7: <15,35,735>;

Ktoré dokumenty zodpovedajú následujúcim dotazom, kde každý výraz v uvodzovkách je frázový dotaz (phrase query)?
a) “fools rush in”
b) “fools rush in” AND “angels fear to tread”.
Na ktorých pozíciách tieto dotazy matchujú?
c) V uvedenom indexe je chyba, kde?

Řešení: Hledáme ve kterých dokumentech jdou slova za sebou. V závorkách jsou indexy pozice slov

a) 2 (1-3), 4 (8-10), 7 (3-5, 13-15)

b) 4 (8-10 AND 12-15)

c) V dokumentu 7 je na indexu 15 jak „where“ tak „in“

4. Príklad

Odporučte stratégiu spracovania dotazu (tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes) vzhľadom na nasledujúce velkosti postings zoznamov:
eyes 213312
kaleidoscope 87009
marmalade 107913
skies 271658
tangerine 46653
trees 316812

Řešení: Použil bych seznam s přeskakovači. Ty se nám vyplatí ale jen na AND, proto bych zpracovával AND jak nejdříve to bude možné. Přeskakuje se po $\sqrt{n}$.

5. Príklad

Máme dotaz zložený z dvoch výrazov. Postings zoznam jedného výrazu je zložený z nasledujúcich 16 položiek:
[4,6,10,12,14,16,18,20,22,32,47,81,120,122,157,180]
a druhý výraz ma postings zoznam len jednoprvkový:
[47].

Zistite koľko porovnaní a prečo je potreba vykonať na prienik týchto dvoch postings zoznamov s nasledujúcimi stratégiami:
a. použitie štandardných postings zoznamov
b. použitie postings zoznamov uložených s preskakujúcimi odkazmi, s dĺžkou preskoku sqrt(P)

Řešení: Nejde vlastně o nic jiného než o procházení seznamu.

a) **je potřeba projít 11 prvků seznamu, než se dojde ke shodě.**
b) **je potřeba 6 projití. Skok 4-12, skok 12-18, skok 18-32, skok 32-120, skok zpět 120-32, krok na 47**
c)

6. Príklad

Pomocou matice vypočítajte Levenshteinovu vzdialenosť slov jablko a malina.

Řešení: Jednoduše řečeno, kolik písmen musíme změnit/přidat/odebrat, aby se ze slova Astalo slovo B. Pro jednoduchost můžeme počítat, že pro všechny operace je stejná váha.

jablko	jablko	jablko	jablko	jablko	jablko	jablko
malina	jalina	malina	malina	malina	malina	malina
0	1	1	2	3	4	5

7. Príklad

a) Nájdite dve rozdielne napísané podstatné mená (anglicky), ktoré majú rovnaký soundex kód.

b) Nájdite dve foneticky podobné podstatné mená (anglicky), ktoré majú rozdielny soundex kód

Řešení: Soundex index je čtyř místný kód. Vytváří se pomocí následujícího postupu.

Slovo „BOOKS“

1. **Necháme první písmeno – „B“**
2. **Nahradíme skupiny písmen – „B0022“**
3. **Odstraníme duplicity – „B02“**
4. **Odstraníme nuly – „B2“**
5. **Změníme velikost na 4 (ořežeme nebo doplníme nuly) – „B200“**

Skupiny písmen:
A, E, I, O, U, H, W, Y -> 0
B, F, P, V -> 1
C, G, J, K, Q, S, X, Z -> 2
D,T -> 3
L -> 4
M, N -> 5.
R -> 6

a) **Trash – T620**
   Thrash – T620
b) **Kill – K400**
   Bill – B400

8. Príklad

Vypíšte prvky slovníka permuterm indexu ktoré su generované slovom mama.

Řešení: Napíšeme za slovo znak „$“ a napíšeme všechny permutace slova

mama$
ama$m
ma$ma
a$mam
$mama

9. Príklad

Aké kľúče sú použiteľné na nájdenie termu s*ng v permuterm wildcard indexe.

Řešení: Doplníme dolar a budeme permutovat tak dlouho, dokud nepřesuneme hvězdičku na konec.

s*ng$
***ng$s**
ng$s*

Hledá se tedy klíč začínající na „ng$s“

10. Príklad

Pre n = 2 a 1 ≤ T ≤ 30, vykonajte krok za krokom simuláciu algoritmu 4.7 (Introduction to Information Retrieval). Vytvorte tabulku, ktorá pre každý okamih v čase, v ktorom je spracovaných T = 2*k tokenov (1 ≤ k ≤ 15), ukazuje ktoré zo štyroch indexov I0, ..., I3 sú používané. Prvé tri riadky tabulky sú uvedené nižšie:

	I3	I2	I1	I0
2	0	0	0	0
4	0	0	0	1
6	0	0	1	0

Řešení: /*chybí*/

2.Cvičení

1.Príklad

Kolekcia dokumentov obsahuje 4 slová: a, b, c, d. Vzájomná frekvencia slov je a > b > c > d. Celkový počet tokenov v kolekcii je 5000. Predpokladajte, ze pre túto kolekciu presne platí Zipfov zákon. Aké su frekvencie vyššie uvedených štyroch slov?

Řešení:
f = a > b > c >d
Aplikujeme Zipa
1/1k + 1/2k +1/3k +1/4k = 5000
Převedeme na společný jmenovatel
12/12k + 6/12k + 4/12k + 3/12k = 5000
Z toho vzejde že
25/12k = 5000, k = 2400
Po dosazení nám vyjde že
a = 2400, b = 1200, c = 800, d = 600

2. Príklad

γ-kódy je neefektívny pre veľké čísla (napr. 1000 alebo 10 000) pretože kódujú dĺžku offsetu v unárnom kóde. δ-kódy používajú gamma kód pre zakódovanie tejto dĺžky.

γ-kód je definovaný ako
unárny_kód(dĺžka(ofset(G))),ofset(G)

δ-kód je definovaný ako
γ(dĺžka(ofset(G+1))),ofset(G+1)

Napríklad δ-kód pre G=6 je 10,0,11. 10,0 je γ-kód pre dĺžku (v tomto prípade 2). Kódovanie ofsetu (11) je rovnaké ako v prípade γ-kódu pre G = 7.

Vypočítajte γ- a δ- kódy pre 1, 2, 3, 4, 31, 63, 127, 1023.

Řešení:

Napřed potřebujeme unární kód čísla, což je sled n jedniček ukončených nulou. Pak offset, což je binární zápis čísla, bez první jedničky. A length, což je počet bitů offsetu, reprezentováno unárně. γ-kód je pak „length, offset“. δ-kód vezme první složku (length) a zakóduje ji do gamma kódu, druhá složka zůstává stejná. S tím rozdílem že vstupní číslo se bere o jedno větší

1:
- **Gamma = 0**
- **Delta = 0**

2:
- **Gamma = 10,0**
- **Delta = 10,0,1**

3:
- **Gamma = 10,1**
- **Delta = 10,0,00**

4:
- **Gamma = 110,00**
- **Delta = 10,0,01**

a tak dále….

3. Príklad

Vypočítajte variabilný byte- a γ- kód pre postings zoznam <777, 17743, 294068, 31251336>. Používajte medzery namiesto docID tam kde je to možné. Binárne kódy napíšte v 8 bitových blokoch.

Řešení: Zajímají nás rozdíly indexů a jejich γ- kódy.

777 = 00000110 10001001b

17743 = 00000001 00001010 11001111b

17743 – 777 = 00000001 00000100 11000110b

γ = 111111111111110 00001001000110

a tak dále…

4. Príklad

Posúďte tabuľku s frekvenciami slov troch dokumentov Doc1, Doc2, Doc3 nižšie. Vypočítajte tf-idf váhy termov *car*, *auto*, *insurance*, *best*, pre každý dokument. Idf hodnoty termov sú uvedené v tabuľke.

	Doc1	**Doc2**	**Doc3**	**idf**
car	27	4	24	1.65
auto	3	33	0	2.08
insurance	0	33	29	1.62
best	14	0	17	1.5

Řešení: Jednoduše se pronásobí řádky tabulky hodnotou IDf.

	Doc1	**Doc2**	**Doc3**
car	44,55	6,6	39,6
auto	6,24	68,64	0
insurance	0	53,46	46,98
best	21	0	25,5

5. Príklad

Vypočítajte normalizované Euclidovské vektory pre každý dokument z predchádzajúceho príkladu, kde každý vektor má štyri komponenty, jednu pre každý zo štyroch termov.

Řešení: Jde v podstatě jen o to, vytvořit normalizované vektory z dokumentů.

S = $\sqrt{44,55^2 + 6,24^2 + 21^2} = 49,6$
Doc1_car = 44,5 / 49,6 = 0,897
Doc1_auto = 6,24 / 49,6 = 0,126
Doc1_insurance = 0
Doc1_best = 21 / 49,6 = 0,423
Doc1 = (0,897; 0,126; 0; 0,423)

a stejně pro další dokumenty…

6. Príklad

S váhami slov ako boli vypočítané v predchádzajúcom príklade, oznámkujte tri dokumenty podľa vypočítaného skóre pre dotaz car insurance, pre každý z nasledujúcich prípadov váženia slov:
a) váha termu je 1 ak sa v dotaze nachádza, inak 0
b) Euclidovské normalizované idf

Řešení:

a) q = „car insurance“

Doc1: (1+0+0+0) = 1

Doc2: (1+0+1+0) = 2

Doc3: (1+0+1+0) = 2

b) = „car insurance“

Doc1: 0,897+0+0+0 = 0,897

Doc2: 0,0756 + 0 + 0,6127 +0 = 0,6883

Doc3: 0,5953 + 0 + 0,7062 +0 = 1,3015

7. Príklad

Vypočítajte vektor-space podobnosť medzi dotazom "digital cameras" a dokumentom "digital cameras and video cameras" doplnením prázdných stĺpcov v tabulke nižšie. Predpokladajte N = 10 000 000, logaritmické váženie termov (stĺpce wf) pre dotaz aj dokumenty, idf váženie len pre dotaz a kosínovú normalizáciu len pre dokument.
"And" považujte za STOP slovo. Napíšte počty termov do tf stĺpca.
Aké je konečné skóre podobnosti?

		Query				Document			Product
	df	tf	wf	idf	q_i=wf-idf	tf	wf	d_i=normalized wf	q_i*d_i
digital	10 000								
video	100 000								
cameras	50 000								

Řešení:

q1 = „digital cameras“

q2 = „digital cameras and video cameras“

tf = počet slov

wf = log(tf) +1

idf = log(N/df)

		Query				Document			Product
	df	**tf**	**wf**	**idf**	**q_i=wf-idf**	**tf**	**wf**	**d_i=normalized wf**	**q_i*d_i**
digital	**10 000**	**1**	**1**	**3**	**3**	**1**	**1**	**0,52**	**1,36**
video	**100 000**	**0**	**0**	**2**	**0**	**1**	**1**	**0,52**	**0**
cameras	**50 000**	**1**	**1**	**2,3**	**2,3**	**2**	**1,3**	**0,68**	**1,56**

8. Príklad

Ukážte, že pre dotaz *affection* je radenie skóre troch dokumentov z tabuľky nižšie v opačnom poradí ako pre dotaz *jealous gossip*. Dotaz je vážený normalizáciou tf.

	SaS	PaP	WH
affection	0.996	0.993	0.847
jealous	0.087	0.120	0.466
gossip	0.017	0	0.254

Řešení: /*chybí*/

3. Cvičení

1. Príklad

IR systém vráti 8 relevantných dokumentov a 10 nerelevantných dokumentov.
Dohromady je v kolekcii 20 relevantných dokumentov. Aká je presnosť a úplnosť (precision, recall)
systému pri tomto vyhľadávaní?

Řešení: Je potřeba spočítat precision a recall. Tyto dva se počítají podle následujících vzorců.

Precision = $\frac{RELEVANTNI}{RELEVANTNI+NERELEVANTNI}$

Recall = $\frac{NALEZENE\ RELEVANTNI}{VSECHY\ RELEVANTNI}$

P = 8/18 = 4/9

R = 8/20 = 2/5

2. Príklad

Nasledujúci zoznam písmen R a písmen N reprezentuje relevantné (R) a nerelevantné (N) dokumenty vrátené v usporiadanom zozname 20 výsledkov ako odpoveď na dotaz z kolekcie 10 000 dokumentov. Prvý (najrelevantnejší) výsledok zoznamu je naľavo. Tento zoznam obsahuje 6 relevantných dokumentov.
Predpokladajte, že kolekcia obsahuje dohromady 8 relevantných dokumentov ku dotazu.
R R N N N N N R N R N N N R N N N R

a) Aká je presnosť systému na prvých 20 výsledkoch?
b) Aká je F_1 na prvých 20 výsledkoch?
c) Aká je neinterpolovaná presnosť systému pri 25% pokrytí?
d) Aká je interpolovaná presnosť systému pri 33% pokrytí?
e) Predpokladajte, že týchto 20 dokumentov je úplný zoznam výsledkov systému. Aký je MAP systému pre tento dotaz?

Teraz predpokladajte, že systém vrátil všetkých 10 000 dokumentov v zoradenom zozname a hore je uvedených prvých 20 vrátených výsledkov.

f) Aký najvyšší možný MAP môže tento systém dosiahnuť?
g) Aký najnižší možný MAP môže tento systém dosiahnuť?
h) Pri sade experimentov bolo vyhodnotených len prvých 20 výsledkov. Výsledok (e) bol použitý na na aproximovanie rozsahu (f)-(g). Aká veľká môže byť chyba pre výpočet MAP pri počítani (e) namiesto (f) a (g) pre tento dotaz?

Řešení:

a) **P = 6/20 = 3/10**

b) $F_1 = \frac{2*P*R}{P+R} = \frac{2*\frac{3}{10}*\frac{3}{4}}{\frac{3}{10}+\frac{3}{4}} = \frac{9}{21} = \frac{3}{7}$

c) **Hledáme množinu všech P, u kterých je R = 25% = ¼. Půjdeme postupně, od prvního.**
1. **P = 1/1 = 1; R= 1/8**
2. **P = 2/2 = 1; R = 2/8 = ¼ - toto nás zajímá**
3. **P = 2/3; R= 2/8 = ¼ - takto to bude pokračovat až k dalšímu N**

Pro R = 25% => P = {1; 2/3; 1/8; 2/5; 1/3; 2/7; 2/8}

d) **Stejný postup jako u „c“, s tím že hledáme pro R = 1/3.**

e) **Musíme spočítat MAP podle vzorce** $MAP = \frac{Suma\ P\ u\ relevantních\ nálezů}{Počet\ P}$.

MAP $= \frac{1+1+\frac{3}{9}+\frac{4}{11}+\frac{5}{15}+\frac{6}{20}}{6} = 0,555$

f) **Předpokládáme, že seznam bude pokračovat …RRNNN…**

MAP $= \frac{1+1+\frac{3}{9}+\frac{4}{11}+\frac{5}{15}+\frac{6}{20}+\frac{7}{21}+\frac{8}{22}}{6} = 0,503$

g) **Předpokládáme, že seznam bude končit …NNNRR**

MAP $= \frac{1+1+\frac{3}{9}+\frac{4}{11}+\frac{5}{15}+\frac{6}{20}+\frac{7}{9999}+\frac{8}{10000}}{6} = 0,417$

3. Príklad

Nižšie je tabuľka ukazujúca ako dvaja znalci ohodnotili relevanciu množiny 12 dokumentov k nejakej informačnej potrebe (0=nerelevantné, 1=relevantné).
Predpokladajme, že ste vyvinuli IR systém, ktorý pre tento dotaz vráti dokumenty {4, 5, 6, 7, 8}.

DocID	Judge 1	Judge 2
1	0	0
2	0	0
3	1	1
4	1	1
5	1	0
6	1	0
7	1	0
8	1	0
9	0	1
10	0	1
11	0	1
10	0	1

a) Vypočítajte Kappa mieru zhody medzi týmito znalcami.
b) Vypočítajte presnosť, pokrytie a F_1 vášho systému, ak je dokument relevantný len ak sa na ňom zhodli obaja znalci.
c) Vypočítajte presnosť, pokrytie a F_1 vášho systému, ak je dokument relevantný ak si to myslí aspoň jeden zo znalcov.

Řešení:

a) **Kappa míra se počítá pomocí vzorce** $k = \frac{(P(A)-P(E))}{(1-P(E))}$ **; kde** $P(A) = \frac{shoda\ všech\ sudích}{počet\ prvků}$ **a** $P(E) = p(rel)^2 + p(notrel)^2$**; kde** $p(rel) = \frac{suma\ všech\ \text{"ano" všech sudích}}{počet\ položek * počet\ sudích}$ **a** $p(notrel) = \frac{suma\ všech\ \text{"ne" všech sudích}}{počet\ položek * počet\ sudích}$

k = $k = \frac{\frac{1}{3}-\frac{1}{2}}{1-\frac{1}{2}} = -\frac{1}{3}$; $P(A) = \frac{4}{12} = \frac{1}{3}$; $P(E) = \left(\frac{6+6}{24}\right)^2 + \left(\frac{6+6}{24}\right)^2 = \frac{1}{2}$

b) $F_1 = \frac{2*\frac{2}{12}*1}{\frac{2}{12}+1} = \frac{1/3}{7/6} = \frac{6}{21}$

c) $F_1 = \frac{2*\frac{10}{12}*1}{\frac{10}{12}+1} = \frac{5/3}{11/6} = \frac{30}{33} = \frac{10}{11}$

4. Príklad

Užívateľov prvotný dotaz je "cheap CDs cheap DVDs extremely cheap CDs". Užívateľ preskúma dva dokumenty d1 a d2. Ohodnotí dokument d1 "CDs cheap software cheap CDs" ako relevantný a d2 "cheap thrills DVDs" nerelevantný. Predpokladajme, že používame jednoduchú tf bez dĺžkovej normalizácie vektorov. Použitím Rocchio relevance feedbacku aký by bol prepracovaný vektor dotazu po zvážení relevance feedbacku?
alpha=1, beta=0.75, gamma=0.25

Řešení: Napíšeme tabulku, kde budou rozepsány četnosti slov dotazu a dokumentů.

	Cheap	CDs	DVDs	extremly	software	thrill
dotaz	**3**	**2**	**1**	**1**	**0**	**0**
d1	**2**	**2**	**0**	**0**	**1**	**0**
d2	**1**	**0**	**1**	**0**	**0**	**1**

Pak zkonstruujeme vektor s následujícím postupem: $(alfa * vektor\ dotazu) + \frac{(beta*suma\ vektor\ relevantni)}{počet\ relevantních} - \frac{(gamma*suma\ vektor\ nerelevantni)}{počet\ nerelevantních} = vysledna\ relevance$

1*(3,2,1,1,0,0)+(0,75*(2,2,0,0,1,0))/1-(0.25*(1,0,1,0,0,1))/1=(4,25; 3,5; 0,75; 1; 0,75; -0,25)

5. Príklad

Prečo je pozitívny feedback pravdepodobne lepší ako negatívny feedback pre IR systém? Prečo je možno lepšie použiť na feedback len jeden nerelevantný dokument ako ich použiť viac?

Řešení: Protože závěry je-li nebo není-li lepší systém s negativním feedbackem jsou dosti nepřesné, a pozitivní feedback nám stačí. Jako negativní je lepší použít jen jeden nejvyšší.

6. Príklad

Prečo je prírastková relevancia viacej realistické merítko užívateľskej spokojnosti?
Udajte príklad kde neprírastková metrika ako napríklad presnosť alebo úplnosť je zavádzajúce merítko užívateľskej spokojnosti a naopak prírastková je lepšia?

Řešení: Řekl bych, že je to tehdy, kdy uživatel hledá spíše podle toho, který dokument vidělo více uživatelů.

4. Cvičení

1. Príklad

Každý z dvoch webových vyhľadávacích systémov A a B zo svojich indexov generujú veľké množstvo stránok rovnomerne náhodne. 30% stránok z A sa nachádza v indexe B a 50% stránok z B
sa nachádzav indexe A. Aký je pomer stránok medzi systémami A a B?

Řešení:

x|A| = y|B|
0,3|A| = 0,5|B|
$\frac{|A|}{|B|} = \frac{5}{3}$

2. Príklad

Každý z dvoch webových vyhľadávacích systémov A a B zbierajú (crawl) náhodnú, ale rovnako veľkú podmnožinu Webu. Niektoré zozbierané stránky sú duplikáty - presné textové kópie na rôznych URL.
Predpokladajte, že sú duplikáty distribuované rovnomerne medzi stránkami zozbierané systémom A aj B. Ďalej predpokaldajte, že duplikát má presne dve kópie - žiadne stránky nemajú viac ako dve kópie. A indexuje stránky bez eliminace duplikátov, kdežto B indexuje len jednu kópiu duplikovaných stránok. Tieto dve náhodné podmonožiny majú rovnakú veľkosť pred odstránením duplikátov.
Ak sa 45% stránok z A nachádza v indexe B, a 50% stránok z B v indexe A, aká veľká časť Webu sa skladá zo stránok, ktoré nemajú duplikáty?

Řešení:
$x|A| = y\left(|B| - \frac{D}{2}\right)$
$0,45|A| = 0,5(|B| - \frac{D}{2})$
$0,9|A| = |B| - \frac{D}{2}$
$1,8|A| = 2|B| - D$
$-0,2|A| = D$ // **|A| a |B| jsou si rovny**
$D = 0,2 * 0,5 = 0,1$ // **|A| je polovina**
D = 10% webu

3. Príklad

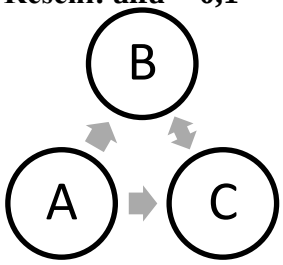
Daný je nasludjúci web graf.
V = {a, b, c}, E = {a -> b, a -> c, b -> c, c -> b}

Vypočítajte PageRank, hub skóre a autoritatívne skóre pre každú z troch stránok. Zoraďte sstránky podľa jednotlivých skóre a pozorujte prípadné väzby.

Pre výpočet PageRank môžte predpokladať, že sa v každom kroku náhodnej prechádzky teleportujeme na náhodnú stránku s pravdepodobnosťou 0.1 a s rovnomernou distribúciou stránok, na ktoré sa teleportujeme.

Pre huby a autority normalizujte skóre tak, aby maximum bolo 1.

Řešení: alfa = 0,1



$$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \text{(každý řádek musí dávat součet 1)} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} * (1 - a) =$$

$$\begin{pmatrix} 0 & \frac{9}{20} & \frac{9}{20} \\ 0 & 0 & \frac{9}{10} \\ 0 & \frac{9}{10} & 0 \end{pmatrix} \text{nuly nahradíme } \left(\frac{\text{alfa}}{\text{počet uzlů}}\right) = \begin{pmatrix} \frac{1}{30} & \frac{9}{20} & \frac{9}{20} \\ \frac{1}{30} & \frac{1}{30} & \frac{9}{10} \\ \frac{1}{30} & \frac{9}{10} & \frac{1}{30} \end{pmatrix} * 30 =$$

$$= \begin{pmatrix} 1 & 90 & 90 \\ 1 & 1 & 180 \\ 1 & 180 & 1 \end{pmatrix}$$

Dále vypočítám vlastní vektor matice

$$\begin{bmatrix} 1-\lambda & 90 & 90 \\ 1 & 1-\lambda & 180 \\ 1 & 180 & 1-\lambda \end{bmatrix} = \cdots$$ **Vlastní vektor je (1/30, 29/60, 29/60)**

4. Príklad

Priemerný vstupný stupeň všetkých uzlov vybraného grafu webu je 9. Čo môžeme povedať o priemernom výstupnom stupni všetkých uzlov tohto grafu?

Řešení:

$$\boldsymbol{A} = \begin{pmatrix} \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \end{pmatrix} \boldsymbol{A^T} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} \end{pmatrix}$$

$\vec{\boldsymbol{h}} = \boldsymbol{A} * \boldsymbol{A^T} * \vec{\boldsymbol{h}} = \begin{pmatrix} \mathbf{2} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} \end{pmatrix} * \vec{\boldsymbol{h}} = (\mathbf{4}, \mathbf{2}, \mathbf{2})$ **– Hub, odkazy na autority**

$\vec{\boldsymbol{a}} = \boldsymbol{A^T} * \boldsymbol{A} * \vec{\boldsymbol{a}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{2} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{2} \end{pmatrix} * \vec{\boldsymbol{a}} = (\mathbf{0}, \mathbf{3}, \mathbf{3})$ **– Autority, obsah jako takový**

5. Cvičení

1. Príklad

Pre XML dokument uvedený nižšie napíšte XPath výrazy.

a) Vráťte všetky názvy (title elementy), ako kurzu, tak oddelenia.
b) Vráťte názvy kurzov, ktoré majú v názve výraz "programming"
c) Vráťte priezviská inštruktorov učiacich aspon jeden kurz, ktorý má vo svojom popise slovo "software".
d) Vráťte priezviská profesorov učiacich aspon jeden kurz, ktorý má vo svojom popise slovo "software".

```
<Course_Catalog>
  <Department Code="CS">
    <Title>Computer Science</Title>
    <Chair>
      <Professor>
        <First_Name>Jennifer</First_Name>
        <Last_Name>Widom</Last_Name>
      </Professor>
    </Chair>
    <Course Number="CS106A" Enrollment="1070">
      <Title>Programming Methodology</Title>
      <Description>Introduction to the engineering of computer applications
emphasizing modern software engineering principles.</Description>
      <Instructors>
        <Lecturer>
          <First_Name>Jerry</First_Name>
          <Middle_Initial>R.</Middle_Initial>
          <Last_Name>Cain</Last_Name>
        </Lecturer>
        <Professor>
          <First_Name>Eric</First_Name>
          <Last_Name>Roberts</Last_Name>
        </Professor>
        <Professor>
          <First_Name>Mehran</First_Name>
          <Last_Name>Sahami</Last_Name>
        </Professor>
      </Instructors>
    </Course>
    <Course Number="CS106B" Enrollment="620">
      <Title>Programming Abstractions</Title>
      <Description>Abstraction and its relation to programming.</Description>
      <Instructors>
        <Professor>
          <First_Name>Eric</First_Name>
          <Last_Name>Roberts</Last_Name>
        </Professor>
        <Lecturer>
          <First_Name>Jerry</First_Name>
          <Middle_Initial>R.</Middle_Initial>
          <Last_Name>Cain</Last_Name>
        </Lecturer>
      </Instructors>
      <Prerequisites>
        <Prereq>CS106A</Prereq>
      </Prerequisites>
    </Course>
    </Department>
</Course_Catalog>
```

Řešení: základní příkazy XPath – „//“ znamená cokoliv, contains(kde, co)

a) **//Title**
b) **//Course[Contains(Title,‘Programing‘)]/Title**
c) **//Course[Contains(Description,‘Software‘)]/Instructors/Last_Name**
d) **//Course[Contains(Description,‘Software‘)]/Instructors/Profesor/Last_Name**

2. Príklad

Vypočítajte podobnosť medzi dotazmi a im zodpovedajúcimi cestami v dokumente z Príkladu 1.
a) //Instructors//Last_Name#Cain
b) //Course/Instructors/Lecturer/Last_Name#Cain

Řešení: Potřebujeme určit kontextovou podobnost, což je funkce $CR = \frac{1+\text{počet uzn\r{u}}\ v\ dotazu}{1+\text{počet uzl\r{u}}\ v\ dokumentu,\ \ pro\ kter\acute{e}\ to\ sed\acute{i}}$**, nebo pokud je dotaz špatný tak je** $CR = 0$**.**

a) $\frac{1+2}{1+6} = \frac{3}{7}$
b) $\frac{1+4}{1+6} = \frac{5}{7}$

3. Príklad

Spočítajte, koľko štruktúrnych termov (structural terms, dvojíc kontext/term <c,t>) je v XML strome na nižšie.

```
<Course>
  <Title>Programming Abstractions</Title>
  <Description>Abstraction and its relation to programming</Description>
  <Instructors>
    <Professor>
      <First_Name>Eric</First_Name>
      <Last_Name>Roberts</Last_Name>
    </Professor>
  </Instructors>
</Course>
```

Řešení: Musí se spočítat pro každé slovo, všechny možné podstomy.

`Programming Abstractions` **– dvě slova**
`<Title>Programming Abstractions</Title>` **– první podstrom**
`<Course>`
`<Title>Programming Abstractions</Title>` **– druhý podstrom**
`</Course>`

2*(1+1) = 4 – A dále pro každé další slova
6*(1+1) = 12
1*(1+1+1+1) = 4
1*(1+1+1+1) = 4
4+12+4+4=24

4. Príklad

Ktorý z dokumentov uvedených nižšie má rovnaké alebo rozdielne bag of words reprezentácie pre Bernouliho a multinomický model? Aké sú rozdiely

Doc1: He moved from London, Ontario, to London, England.
Doc2: He moved from London, England, to London, Ontario.
Doc3: He moved from England to London, Ontario.

Řešení:

Podle Bernouliho jsou všechny stejné, protože Bernouli nezahrnuje duplikáty.
{London, London} = {London}

Podle multinomického modelu jsou různé, protože zahrnuje duplikáty
{London, London} ≠ {London}

5. Príklad

Na základe dát z tabuľky nižšie
a) odhadnite multinomické Naive Bayes klasifikátory,
b) aplikujte ich na testovací dokument,
c) odhadnite Bernoulli Naive Bayes klasifikátor,
d) aplikujte ho na testovací dokumnet
Nemusíte odhadovať parametre, ktore na klasifikáciu dokumnetu nie sú potrebné.

	docID	obsah dokumentu	je v c = China?
trénovacia množ.	1	Taipei Taiwan	áno
	2	Macao Taiwan Shanghai	áno
	3	Japan Sapporo	nie
	4	Sapporo Osaka Taiwan	nie
testovacia množ.	5	Taiwan Taiwan Sapporo	?

Řešení:

Klasifikátory se určují pro každé slovo podle

$$\boldsymbol{P}(\boldsymbol{slovo}|+/-) = \frac{(\boldsymbol{počet\ výskytů\ v + nebo-})\ + \mathbf{1}}{(\boldsymbol{počet\ slov\ v + nebo-}) + (\boldsymbol{počet\ různých\ slov\ v\ sad\check{e}})}$$

$$\boldsymbol{P}(+/-) = \frac{(\boldsymbol{počet + nebo - dokumentu})}{\boldsymbol{počet\ všech\ dokumentu}}$$

Pro konkrétní aplikaci pak

$$\boldsymbol{P}(+/-|\boldsymbol{test}) = \boldsymbol{P}(+/-) * (\boldsymbol{všechny\ váhy\ slov\ v\ testu\ pronásobené})$$

a)
P(Taiwan|+) = (2+1)/(5+7) = 1/4
P(Saporro|+) = (0+1)/(5+7) = 1/12
P(+) = 2/4 = 1/2
P(Taiwan|-) = (1+1)/(5+7) = 1/6
P(Saporro|-) = (2+1)/(5+7) = 1/4
P(-) = 2/4 = 1/2

b)
P(+|test) = 1/2 * 1/4 * 1/4 * 1/12 = 0,0026
P(-|test) = 1/2 * 1/6 * 1/6 * 1/4 = 0,0035

c)
P(Taiwan|+) = (2+1)/(2+2) = 3/4
P(Saporro|+) = (0+1)/(2+2) = 1/4
P(+) = 2/4 = 1/2
P(Taiwan|-) = (1+1)/(2+2) = 1/2
P(Saporro|-) = (2+1)/(2+2) = 3/4
P(-) = 2/4 = 1/2

d)
P(+|test) = 1/2 * 3/4 * 1/4 = 0,0937
P(-|test) = 1/2 * 1/2 * 3/4 = 0,1875