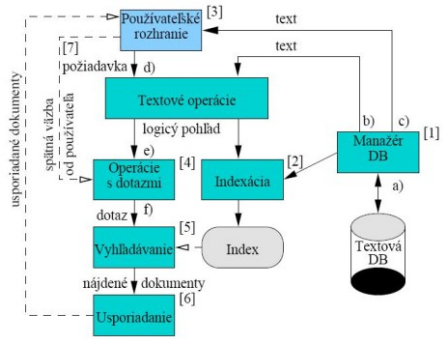
Manažment znalostí (2)

OBSAH PREDNÁŠKY

- Proces vyhľadávania informácií (information retrieval – IR)
  - stručné zopakovanie základných informácií z prvej prednášky
- Klasické modely pre IR
  - Boolovský
    - Stručné zopakovanie
  - Vektorový
    - Rôzne typy váhovania indexových termov
    - Výpočet podobnosti dopytu a dokumentu
  - Pravdepodobnostný
    - Základné východiská

Proces vyhľadávania informácií (1)



Formálna definícia IR modelu

- **Model je štvorica (D, Q, F, $R(q,d_j)$)**, kde
  - D je množina reprezentácií dokumentov d_j kolekcie
  - Q je množina reprezentácií používateľských otázok q
  - F je spôsob (matematický aparát) modelovania reprezentácií dokumentov, otázok a ich vzťahov
  - $R(q,d_j)$ je ohodnocovacia funkcia, ktorá priradí dvojici $(q, d_j) \in Q \times D$ reálne číslo. Toto ohodnotenie (ranking) potom určuje usporiadanie dokumentov vrátených ako odpoveď systému na používateľskú otázku q

Boolovský model

- **Boolovský model**: $w_{ij} \in \{0,1\}$; dopyt je podmnožina indexových termov pospájaných logickými spojkami *AND*, *OR* alebo *NOT*
- Jedna z možných foriem vnútornej reprezentácie dokumentov pre boolovskom modeli je incidenčná matica term-dokument

Príklad boolovskej reprezentácie: incidenčná matica term-dokument

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

- Stĺpce predstavujú jednotlivé dokumenty (divadelné hry)
- Riadky predstavujú vybrané termy
- Hodnota 1 znamená, že dané slovo sa vyskytuje v danej hre, ináč 0

Príklad boolovského dopytu a jeho vyhodnotenia

- Takže pre každý term máme vektor binárnych hodnôt dĺžky rovnej počtu dokumentov v korpuse
- Ak chceme napr. vyhľadať hry, v ktorých sa vyskytuje *Brutus*, *Caesar* ale nevyskytuje sa tam *Calpurnia*, môžeme sformulovať dopyt:

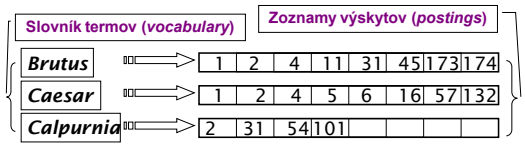
  Brutus AND Caesar BUT NOT Calpurnia

  110100 *AND* 110111 *AND* 101111
- Jeho výsledok získame bitovým súčinom, t.j.

  110100 *AND* 110111 *AND* 101111 = **100100**
- Takže danej podmienke vyhovujú hry *Antony and Cleapatra* a *Hamlet*

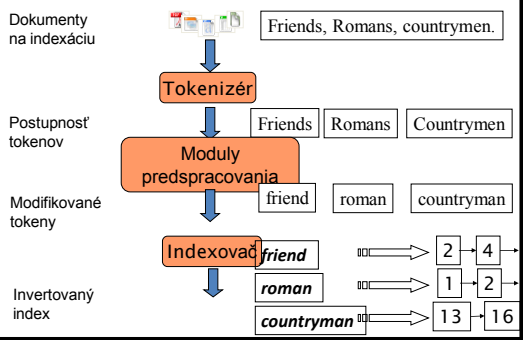
Invertovaný index (1)

- Pre každý term si musíme uchovať zoznam všetkých dokumentov, v ktorých sa vyskytuje



- Potrebujeme teda zoznamy výskytov s premenlivou dĺžkou (v pamäti reprezentované spojkovým zoznamom, alebo poliami s premenlivou veľkosťou)

Konštrukcia invertovaného indexu



Činnosť indexovača (1)

1. Vytvorí postupnosť párov typu:
   - (modifikovaný token, ID dokumentu)

Dokument 1: I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Dokument 2: So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Činnosť indexovača (2)

2. Usporiadanie zoznamu:
   - podľa abecedy, a potom
   - podľa ID dokumentu

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Činnosť indexovača (3)

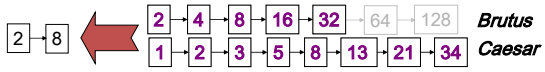
3. Viacnásobné výskyty termu v dokumente sa zlúčia do jedného
4. Rozdelí sa slovník termov a zoznamy výskytov
5. Pridá sa informácia o frekvencii výskytov jednotlivých termov

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

term	doc. freq.	→	postings list
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Vyhodnotenie dopytu nad invertovaným indexom

- Uvažujme jednoduchý dopyt *BRUTUS* AND *CAESAR*
- Postup vyhodnotenia tohto dopytu:
  1. Nájdi v slovníku termov *BRUTUS* a získaj jeho zoznam výskytov
  2. Nájdi v slovníku termov *CAESAR* a získaj jeho zoznam výskytov
  3. Sprav prienik oboch zoznamov výskytov



- Ak sú dĺžky vstupných zoznamov výskytov x a y, potom operácia ich zlúčenia má časovú zložitosť $O(x + y)$
- Dôležitým predpokladom je, že zoznamy výskytov sú usporiadané podľa ID dokumentu

Algoritmus pre nájdenie prieniku dvoch usporiadaných zoznamov

INTERSECT(p_1, p_2)
answer ← <>
while $p_1 \neq$ NIL and $p_2 \neq$ NIL
  do if *docID*(p_1) = *docID*(p_2)
    then ADD(*answer*, *docID*(p_1))
      p_1 ← *next*(p_1)
      p_2 ← *next*(p_2)
    else if *docID*(p_1) < *docID*(p_2)
      then p_1 ← *next*(p_1)
      else p_2 ← *next*(p_2)
return *answer*

Algoritmus pre vyhodnotenie konjunktívnych dopytov

INTERSECT(<t_1, . . . , t_n>)
  terms ← SortByIncreasingFrequency(<t_1, . . . , t_n>)
  result ← *postings*(*first*(*terms*))
  terms ← *rest*(*terms*)
  while *terms* ≠ NIL and *result* ≠ NIL
    do *result* ← INTERSECT(*result*, *postings*(*first*(*terms*)))
      terms ← *rest*(*terms*)
  return *result*

14

Boolovský model - sumár

- **Výhody**
  - Jasný formalizmus
  - Jednoduchosť
- **Nevýhody**
  - Presná zhoda výskytu termov otázky v dokumente môže viesť k príliš veľkému (*OR*) alebo naopak príliš malému (*AND*) počtu dokumentov v odpovedi
    - Nevadí pri strojovom spracovaní, nevhodné pre používateľov
  - Dokumenty nemožno usporiadať podľa stupňa relevancie k otázke
    - Pri usporiadaní podľa relevancie počet dokumentov v odpovedi nie je problémom (stačí uvažovať prvých *k*)
  - Neberie sa do úvahy frekvencia výskytu jednotlivých termov otázky v dokumente

15

Zohľadnenie frekvencie výskytu termu v dokumente – vektorový model (1)

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

- Táto reprezentácia nezohľadňuje poradie slov v dokumente => tzv. "*bag of words*" model
- Ako využiť informáciu o frekvencii výskytu termu tf_{ij} pre výpočet miery relevancie k dopytu?

16

Zohľadnenie frekvencie výskytu termu v dokumente (2)

- Frekvencia výskytu nie je presne to, čo chceme
  - dokument s 10-timi výskytmi termu bude viac relevantný ako dokument s jedným výskytom, ale nie 10-krát relevantnejší
  - Relevancia nerastie proporcionálne s frekvenciou výskytu
- Dá sa však použiť logaritmus frekvencie výskytu tak, aby:
  - pre 0 → 0, pre 1 → 1, pre 2 → 1.3, pre 10 → 2, pre 1000 → 4 ...

$$w_{i,j} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{i,j}, & \text{if } \mathrm{tf}_{i,j} > 0 \\ 0, & \text{iná č} \end{cases}$$

- Pre výpočet podobnosti dokumentu *d* voči dopytu *q* možno použiť vzťah:

$$sim(q,d) = \sum_{t \in q \cap d} (1 + \log_{10} \mathrm{tf}_{t,d})$$

17

Zohľadnenie frekvencie výskytu termu v celej kolekcii dokumentov

- Okrem frekvencie výskytu termu v danom dokumente je dôležitá aj jeho frekvencia v celej kolekcii.
  - Zriedkavé termy nesú viac informácie ako často sa vyskytujúce termy, čo chceme vyjadriť aj číselne
- Dokumentová frekvencia df_i termu *i* je počet dokumentov v kolekcii, v ktorých sa tento term nachádza
  - df_i je nepriamo úmerná informatívnosti daného termu pre vyhľadávanie. Platí tiež, že $df_i \leq N$
- Definujeme preto inverznú dokumentovú frekvenciu:

$$\mathrm{idf}_i = \log_{10}\left(\frac{N}{df_i}\right)$$

- Každý term t_i v kolekcii má teda jednu hodnotu idf_i =>
- Pri jednoslovných dopytoch idf_i nemá vplyv na výsledné usporiadanie dokumentov podľa relevancie

18

Váhovanie *tf-idf*

- Najlepšia známa váhovacia schéma pre vyhľadávanie informácií je súčin váh *tf* a *idf*

$$w_{ij} = (1 + \log \mathrm{tf}_{i,j}) \times \log_{10}\left(\frac{N}{df_i}\right)$$

- Výpočet podobnosti dokumentu voči dopytu:

$$sim(q,d) = \sum_{t \in q \cap d} \mathrm{w}_{t,d}$$

19

Matica term-dokument pri použití tf-idf váhovania

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5,25	3,18	0	0	0	0,35
Brutus	1,21	6,1	0	1	0	0
Caesar	8,59	2,54	0	1,51	0,25	0
Calpurnia	0	1,54	0	0	0	0
Cleopatra	2,85	0	0	0	0	0
mercy	1,51	0	1,9	0,12	5,25	0,88
worser	1,37	0	0,11	4,15	0,25	1,95

- Dokumenty sú teda vektory reálnych hodnôt v T - rozmernom priestore ($d_j \in R^T$), resp.
- Každý dokument je jeden bod v tomto priestore

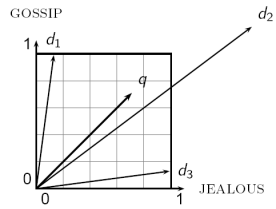
20

Dopyty ako vektory

- Tak ako dokumenty, aj dopyty možno reprezentovať vektormi v *T*-rozmernom priestore
- Usporiadanie dokumentov podľa relevancie k dopytu je potom usporiadaním blízkosti ich vektorov k vektorom daného dopytu
- Blízkosť vektorov = podobnosť vektorov
- Blízkosť ≈ opak vzdialenosti
- Použiť Euclidovskú vzdialenosť?
  - Nie je dobrý nápad, lebo má veľkú hodnotu pre vektory rôznej dĺžky

21

Prečo vzdialenosť nie je v tomto prípade dobrá miera

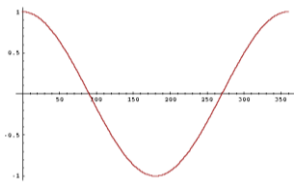


- Preto sa ako miera podobnosti pri vyhľadávaní informácií používa uhol medzi vektorom dokumentu a vektorom dopytu

22

Kosínusová miera podobnosti

- Uvedené dve tvrdenia sú ekvivalentné:
  - Usporiadať dokumenty podľa klesajúcej hodnoty uhla medzi dopytom a dokumentom
  - Usporiadať dokumenty podľa stúpajúcej hodnoty kosínusu uhla medzi dopytom a dokumentom
- Kosínus je monotónne klesajúca funkcia na intervale [0°, 180°]



Normalizácia dĺžky vektora

- Vektor môže byť normalizovaný predelením všetkých jeho zložiek jeho dĺžkou, t.j. $|\vec{x}| = \sqrt{\sum_i x_i^2}$
- Výsledkom normalizácie je, že vektor má jednotkovú dĺžku
- Kosínusová podobnosť medzi dopytom a dokumentom:

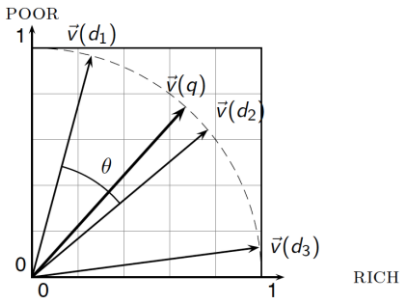
$$sim(q,d) = \cos(\vec{q},\vec{d}) = \frac{\vec{q} \bullet \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|} \bullet \frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|r|} q_i d_i}{\sqrt{\sum_{i=1}^{|r|} q_i^2} \sqrt{\sum_{i=1}^{|r|} d_i^2}}$$

- Pre vektory normalizované na jednotkovú dĺžku vektora je kosínusová vzdialenosť jednoducho skalárnym súčinom normalizovaných vektorov *q* a *d*

$$sim(q,d) = \cos(\vec{q},\vec{d}) = \vec{q} \bullet \vec{d} = \sum_{i=1}^{|r|} q_i d_i$$

24

Ilustrácia kosínusovej miery podobnosti



Rôzne varianty tf-idf schémy

Term frequency		Document frequency		Normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha < 1$
L (log ave)	$\frac{1+\log(tf_{t,d})}{1+\log(ave_{t \in d}(tf_{t,d}))}$				

- Používané sumárne označenie pre popis daného vyhľadávacieho stroja *ddd.qqq*
- Napr. označenie ***ltc.lnn*** znamená:
  - dokument: logaritmickú *tf*, *idf* a kosínusovú normalizáciu
  - dopyt: logaritmickú *tf*, bez *idf* a bez normalizácie

Príklad *tf-idf* výpočtu pre *lnc.ltc*

- Dokument: *„car insurance auto insurance“*
- Dopyt: *„best car insurance“*

Term	Dopyt						Dokument				Súčin
	tf_{ij}	tf_{ij}-log	df_i	idf_i	w_{ij}	nor mal.	tf_{ij}	tf_{ij}-log	w_{ij}	nor mal.	
auto	0	0	5000	2.3	0	0	1	1	1	0.52	0
best	1	1	50000	1.3	1.3	0.34	0	0	0	0	0
car	1	1	10000	2.0	2.0	0.52	1	1	1	0.52	0.27
insurance	1	1	1000	3.0	3.0	0.78	2	1.3	1.3	0.68	0.53

- Viete povedať aká je hodnota *N*?
- Miera podobnosti sim = 0 + 0 + 0.27 + 0.53 = **0.8**

Vektorový model - sumár

- **Výhody**
  - Schéma váženia termov podľa frekvencie ich výskytu zvyšuje výkonnosť vyhľadávania
  - Vyhľadá aj dokumenty ktoré len čiastočne vyhovujú zadanej otázke
  - Usporiadanie nájdených dokumentov podľa stupňa ich relevancie
- **Nevýhody**
  - Predpoklad nezávislosti indexových termov síce neplatí, ale prakticky ide väčšinou iba o lokálne závislosti malých skupín termov

Pravdepodobnostný model (1)

- Tento model sa snaží odhadnúť pravdepodobnosť, že používateľ bude považovať daný dokument d_j za relevantný k svojej otázke q
- Preto nutne musí predpokladať, že:
  - Táto pravdepodobnosť závisí len od otázky q a dokumentu d_j
  - Existuje podmnožina všetkých dokumentov, ktorú používateľ preferuje ako odpoveď na svoju otázku q, tzv. ideálna odpoveď R
  - Dokumenty z R budú predikované ako relevantné ku q, ale všetky ostatné dokumenty mimo R budú nerelevantné
- $w_{ij} \in \{0,1\}$ aj $w_{iq} \in \{0,1\}$; q je podmnožina indexových termov

Pravdepodobnostný model (2)

$p(R/\bar{d}_j)$ Je pravdepodobnosť toho, že dokument d_j je relevantný ku otázke q

$p(\bar{R}/\bar{d}_j)$ Je pravdepodobnosť toho, že dokument d_j nie je relevantný ku otázke q

$$sim(d_j,q) = \frac{p(R/\bar{d}_j)}{p(\bar{R}/\bar{d}_j)} = \frac{p(\bar{d}_j/R)\cdot p(R)}{p(\bar{d}_j/\bar{R})\cdot p(\bar{R})} \approx \frac{p(\bar{d}_j/R)}{p(\bar{d}_j/\bar{R})}$$

$p(k_i/R)$ Je pravdepodobnosť výskytu termu k_i v dokumente náhodne vybratom z R

$p(\bar{k}_i/R)$ Je pravdepodobnosť, že sa term k_i nevyskytuje v dokumente náhodne vybratom z R

$$sim(d_j,q) = \frac{(\prod_{g_i(\bar{d}_j)=1} p(k_i/R))\cdot(\prod_{g_i(\bar{d}_j)=0} p(\bar{k}_i/R))}{(\prod_{g_i(\bar{d}_j)=1} p(k_i/\bar{R}))\cdot(\prod_{g_i(\bar{d}_j)=0} p(\bar{k}_i/\bar{R}))}$$

Pravdepodobnostný model (4)

$$sim(d_j, q) \approx \sum_{i=1}^{t} w_{i,q} \cdot w_{i,j} \cdot (\log \frac{p(k_i/R)}{1-p(k_i/R)} + \log \frac{1-p(k_i/\overline{R})}{p(k_i/\overline{R})})$$

- Keďže množinu *R* na začiatku nepoznáme, je nutné nájsť spôsob inicializácie vyššie uvedených pravdepodobností. Na to existuje niekoľko spôsobov, napr.:

$p(k_i/R) = 0{,}5$ Predpokladáme, že výskyt všetkých termov k_i v dokumentoch z *R* je rovnako pravdepodobný

$p(k_i/\overline{R}) = \frac{n_i}{N}$ Distribúcia termu k_i mimo dokumentov z *R* je zhodná s jeho distribúciou v celej množine dokumentov

31

Pravdepodobnostný model (5)

- Neskôr je táto inicializačná hodnota spresňovaná nasledovne:
- Nech *V* je množina dokumentov vrátených v 1. iterácii ako odpoveď na *q* a vo V_i z nich sa vyskytuje term k_i
- Potom sú možné tieto alternatívne vyjadrenia pravdepodobností:

$$p(k_i/R) = \frac{V_i}{V} \approx \frac{V_i + 0{,}5}{V+1} \approx \frac{V_i + \frac{n_i}{N}}{V+1}$$

$$p(k_i/\overline{R}) = \frac{n_i - V_i}{N-V} \approx \frac{n_i - V_i + 0{,}5}{N-V+1} \approx \frac{n_i - V_i + \frac{n_i}{N}}{N-V+1}$$

- Tento proces môže pokračovať aj bez asistencie človeka, alebo s jeho asistenciou tak, že človek vyberie z odpovede systému množinu *V*

32

Pravdepodobnostný model (2)

- **Výhody**
  - Usporiadanie nájdených dokumentov podľa pravdepodobnosti ich relevancie k otázke
- **Nevýhody**
  - Nutnosť počiatočného odhadu niektorých pravdepodobností
  - Neberie sa do úvahy frekvencia výskytu jednotlivých termov otázky v dokumente
  - Predpoklad nezávislosti indexových termov neplatí (ale prakticky ide väčšinou iba o lokálne závislosti malých skupín termov, takže to v podstate nevadí)

33