

# Uvod v informacijsko povpraševanje

Iztok Savnik

# Viri

- Predavanje temelji na naslednjih virih:
  - Knjiga: Christopher Manning, An Introduction to Information Retrieval, Draft 2009, Cambridge
  - Predavanja pri predmetu: CS276A, Information Retrieval, Uni Stanford, 2009
  - Knjiga: Raghu Ramakrishnan, Database Management Systems, McGraw Hill, 2006
  - Predavanje: Introduction to IR Systems, Uni Wisconsin

# Vsebina

- Uvod v IR
- Invertirani indeksi
- Boolove poizvedbe
- Pretvorba teksta v žetone
- Ocenjevanje rezultatov
- Vektorji
- Google

# Uvod v IR

# Informacijsko poizvedovanje

- Raziskovalno področje, ki je tradicionalno ločeno od podatkovnih baz
  - Začetki v IBM, Rand in Lockheed v 50-tih
  - G. Salton na Cornell v 60-tih
  - Od takrat je bilo precej raziskav na področju
- Produkti so tradicionalno ločeni od SUPB
  - Originalno so to sistemi za upravljanje dokumentov za knjižnice, vlade, pravo, itd.
  - V zadnjih letih je področje pridobilo na polularnosti zaradi svetovnega spleta

# IP vs. SUPB

- Zelo različna sistema:

IP	SUPB
Nenatančen pomen	Natančen pomen
Iskanje s klj. besedami	SQL
Nestrukt. oblika podatkov	Strukturirani podatki
Večinoma branje; dodajanje dok. občasno	Veliko sprememb podatkov
Stran s <b>“top k”</b> rezultati	Generiranje kompletnega odgovora

- Oba podpirata poizvedbe nad velikimi količinami podatkov z uporabo indeksiranja
  - V praksi je potrebno izbrati med dvema

# Model “vreče besed”

- Tipični IP podatkovni model
  - Vsak dokument je samo vreča (multiset) besed
- Detail 1: “bele besede”
  - Nekateri besede in se ne vstavljajo v vrečo
  - Primer: “*the*”, <H1>, *itd.*
- Detail 2: “Krnjenje” in druge pretvorbe vsebine
  - Uporaba jezikovno-odvisnih pravil za pretvorbo v osnovno obliko besed
  - Primer: “*surfing*”, “*surf*ed” --> “*surf*”

# Boolovo iskanje

- Poišči dokumente, ki se ujemajo z Boolovim izrazom vsebovanosti:  
    “Windows”  
    AND (“Glass” OR “Door”)  
    AND NOT “Microsoft”
- Opazka: povpraševalni izrazi se tudi filtrirajo z uporabo krnjenja...
- Ko pravi spletni iskalnik “10,000 documents found“ je to velikost rezultata Boolovega iskanja



# Indeksi za tekst

- Ko IR skupnost pravi “tekstovni indeks”...
  - Običajno pomeni precej več kot DB skupnost misli
- V DB izrazoslovju, vsebuje tekstovni indeks datoteko in indeks
  - Pravzaprav je to logična shema (t.j. tabele)
  - Skupaj s fizično shemo (t.j. indeksi)
  - Praviloma se ne shranjuje v SUPB
    - Tabele so implementirane kot datoteke datotečnega sistema

# Invertirana datoteka

- Iskanje
  - “databases”
  - “microsoft”

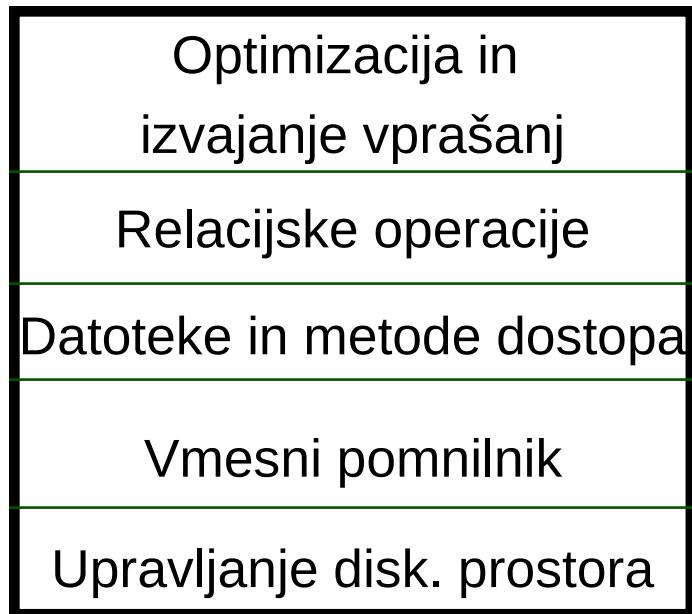
term	docURL
data	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
database	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
date	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
day	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
dbms	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
decision	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
demonstrate	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
description	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
design	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
desire	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
developer	<a href="http://www.microsoft.com">http://www.microsoft.com</a>
differ	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
disability	<a href="http://www.microsoft.com">http://www.microsoft.com</a>
discussion	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
division	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
do	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>
document	<a href="http://www-inst.eecs.berkeley.edu/~cs186">http://www-inst.eecs.berkeley.edu/~cs186</a>

# Uporaba boolove logike

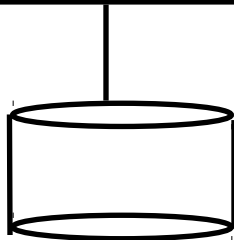
- Kako izračunati boolove izraze?
  - “term1” OR “term2”?
    - Unija dveh množic DocID-jev !
  - “term1” AND “term2”?
    - Presek dveh množic DocID-jev!
    - Storitve + zlivanje seznamov
  - “term1” AND NOT “term2”?
    - Razlika množic z uporabo sortiranja
  - “term1” OR NOT “term2”
    - Unija “term1” in “NOT term2”
    - “NOT term2” = vsi dokumenti, ki ne vsebujejo term2
    - Velika množica!! se običajno ne dovoli !

# SUPB vs. IR arhitektura

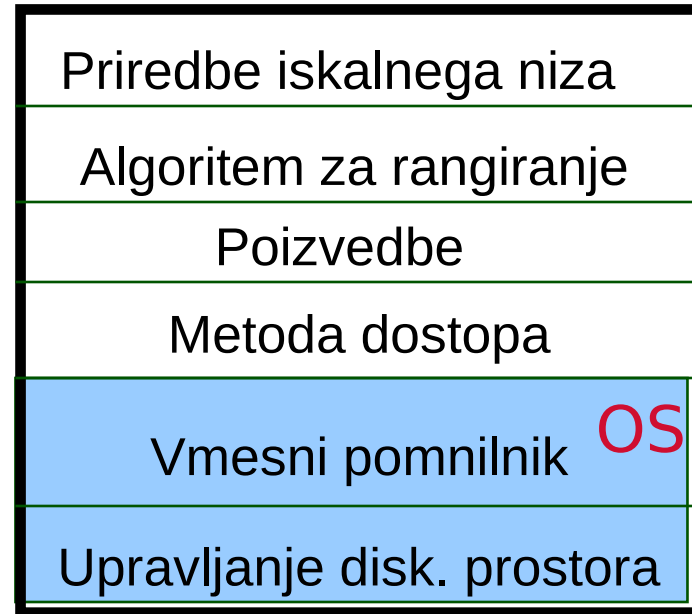
## SUPB



Vzporednost  
in  
Obnavljanje

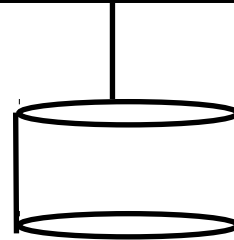


## Iskalnik



Enost.  
SUPB

OS



# Veliko več v IR ...

- Kako rangirati izhod? Kako izračunati „pomembnost“ vsakega zapisa rezultata?
  - To narediti v redu in učinkovito je težko !
- Drugi načini za pomoč uporabniku pri takšnem številu izhodov?
  - Grupiranje dokumentov, vizualizacija dokumentov, ...
- Si lahko pomagamo s hiper-povezavami?
  - Lepi triki !
- Kako uporabiti kompresijo za boljše I/O performanse?
  - Zmanjšanje sznamov DocID-jev; poskusimo dati stvari v RAM
- Kako delati s sinonimi, napačnim črkovanjem, kraticami?
- Kako napisati dober spletni crawler?

# Invertirani indeksi

# Vprašanja

- Katera dela avtorja Shakespeare vsebujejo besede **Brutus** IN **Caesar** vendar NE **Calpurnia**?
- Lahko bi uporabil grep nad vsemi Shakespeare-jevimi deli, ki vsebujejo **Brutus** in **Caesar**, ter potem izločil vrstice (datoteke), ki vsebujejo besedo **Calpurnia**?
  - Počasno (za veliko število del)
  - NOT **Calpurnia** ni trivialno
  - Ostale operacije (npr. Poišči besedo **Romans** blizu **countrymen**) niso učinkovite

# Relacija beseda-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

***Brutus AND Caesar but NOT Calpurnia***

1 če igra  
vsebuje besedo  
in 0 sicer



# Vektor ujemanja

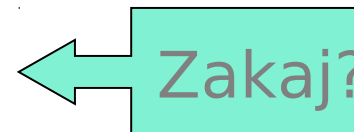
- Vektor, ki vsebuje 0/1 za vsako besedo
- Odgovor na poizvedbo: vzami vektorje za ***Brutus, Caesar*** in ***Calpurnia*** (komplementiran) □ naredi *AND po bitih*.
- $110100 \text{ AND } 110111 \text{ AND } 101111 = 100100$ .

# Večja zbirka

- Recimo, da je  $n = 1\text{M}$  dokumentov, ki vsak vsebuje 1K besed.
- Povprečno imamo 6 zlogov/besedo vključno s presledki/ločili.
  - 6GB podatkov v dokumentih.
- Recimo, da imamo  $m = 500\text{K}$  različnih besed.

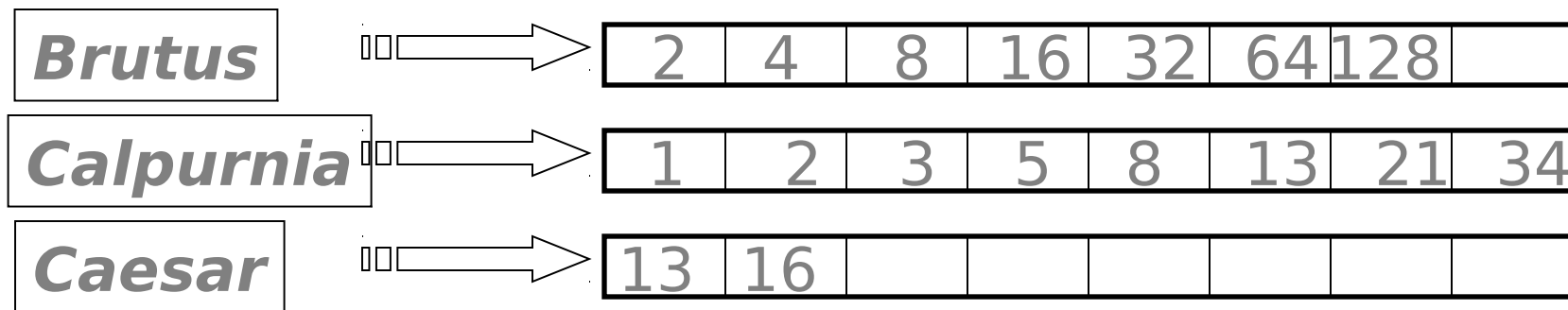
# Ne moremo zgraditi matrike

- 500K x 1M matrika ima polovico triljona ničel in enic.
- Nimamo pa več kot en biljon enic!
  - Matrika je zelo redka.
- Kaj je boljša predstavitev?
  - Zapomnimo si samo enice.



# Invertiran indeks

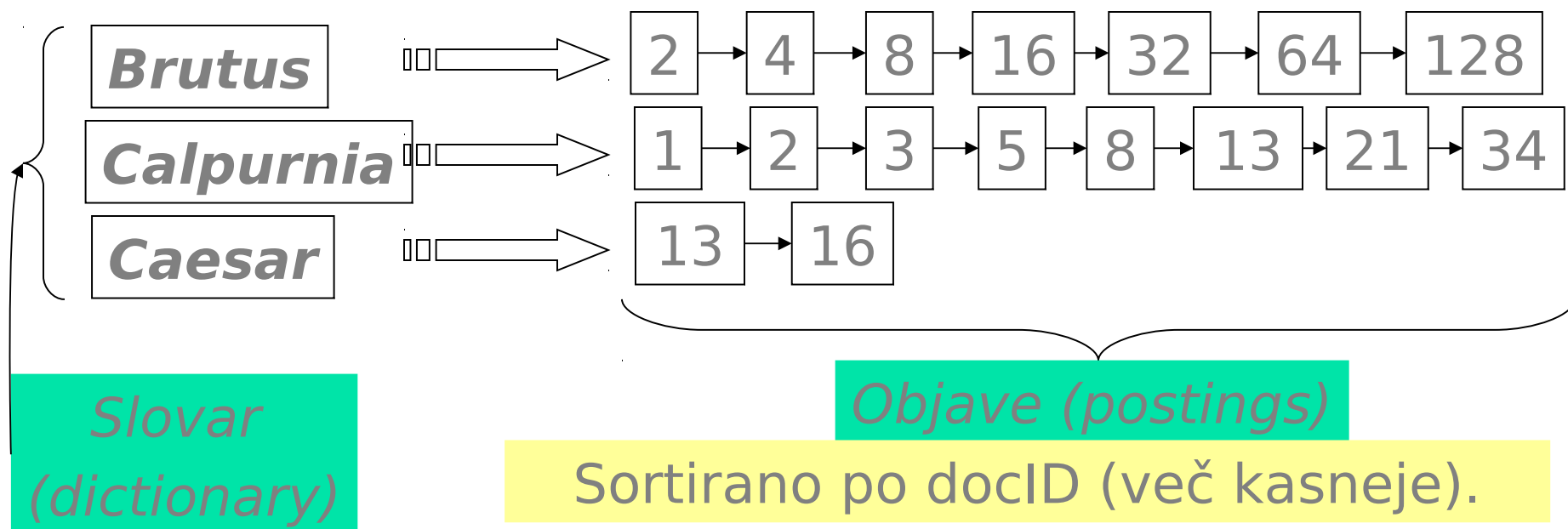
- Za vsako besedo *T* shranimo seznam vseh dokumentov, ki vsebujejo *T*.
- Naj uporabimo polje ali seznam za predstavitev?



Kaj se zgodi, če dodamo besedo **Caesar** k dokumentu 14?

# Invertiran indeks (2)

- Seznammi so v splošnem bolj zaželeni kot polja
  - Dinamična alokacija pom.prostora
  - Vstavljanje besed je enostavno
  - Poraba prostora zaradi kazalcev



# Konstrukcija invertiranega indeksa

Dokumenti  
za indeksiranje



Friends, Romans, countrymen.

⋮

Tokenize

Tok žetonov

Friends

Romans

Countrymen

*Več o tem  
kasneje.*

Linguistični  
moduli

Spremenjeni žetoni

friend

roman

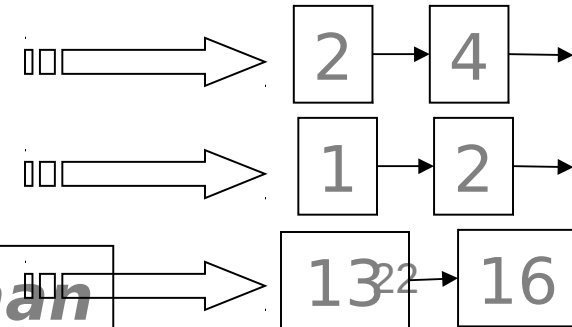
countryman

Indeksi

**friend**

**roman**

**countryman**



Invertiran indeks.

# Koraki indeksiranja

- Sekvenca (Obdelana beseda, ID dokumenta)  
parov

Doc 1

Doc 2



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

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

Term	Doc #
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
	23

- Sortiraj po besedah

**Glavni korak indeksiranja**

Term	Doc #
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	Doc #
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



- 
- Zakaj frekvenca?

25

Rezultat je razcepljen v datoteko slovarja in datoteko objav.

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



Term	N docs	Tot Freq
ambitious	1	1
be	1	1
brutus	2	2
capitol	1	1
caesar	2	3
did	1	1
enact	1	1
hath	1	1
I	1	2
i'	1	1
it	1	1
julius	1	1
killed	1	2
let	1	1
me	1	1
noble	1	1
so	1	1
the	2	2
told	1	1
you	1	1
was	2	2
with	1	1

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
1	1
1	1
2	1
1	1
1	2
2	1
1	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1
2	1

# Kje plačamo pomnilnik?

Slovar

Term	N docs	Tot Freq
ambitious	1	1
be	1	1
brutus	2	2
capitol	1	1
caesar	2	3
did	1	1
enact	1	1
hath	1	1
I	1	2
i'	1	1
it	1	1
julius	1	1
killed	1	2
let	1	1
me	1	1
noble	1	1
so	1	1
the	2	2
told	1	1
you	1	1
was	2	2
with	1	1

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
1	1
1	1
2	1
1	2
1	1
2	1
1	1
1	2
2	1
1	1
2	1
2	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1

Objave

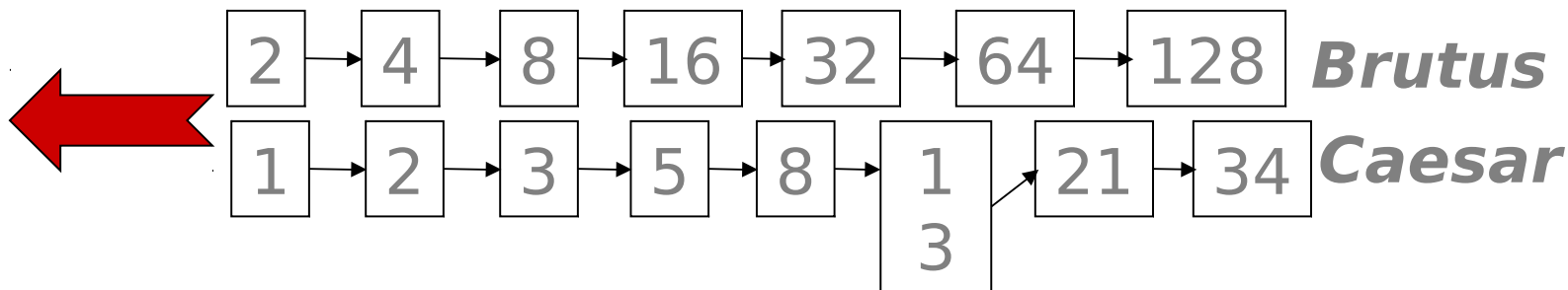
# Boolove poizvedbe

# Procesiranje poizvedb

- Poglejmo si procesiranje poizvedbe:

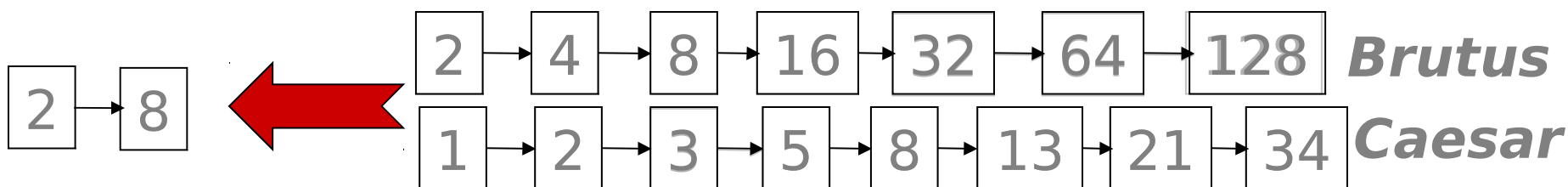
## ***Brutus AND Caesar***

- Poišči ***Brutus*** v slovarju;
  - Vrni postings.
- Poišči ***Caesar*** v slovarju;
  - Vrni postings.
- “Združi” dvojce seznamov:



# Zlivanje seznamov

- Sprehod po seznamu v linearnem času glede na oba seznama



Dolžini seznamov  $x$  in  $y \implies O(x+y)$

Ključno: objave sortirane po docID.

# Boolove poizvedbe: natančno ujemanje

- Boolove poizvedbe so vprašanja z *AND*, *OR* in *NOT* skupaj z izrazi (besedami)
  - Vsak dokument je vreča besed
  - Natančno: dokument ustreza pogoju ali ne.
- Osnovno komercialno orodje v zadnjih 3 desetletjih
- Profesionalni uporabniki (npr. pravniki) imajo radi boolove poizvedbe:
  - Veš natančno kaj dobiš

# Primer: WestLaw <http://www.westlaw.com/>

- Največji komercialni naročniki za iskanje pravnih dokumentov (začetek 1975; rank dodan 1992)
- Okoli 7 tera-zlogov podatkov; 700,000 uporabnikov
- Večina uporabnikov še vedno uporablja boolova vprašanja
- Dolge, natančne poizvedbe; inkrementalno razvita vprašanja; ni podobno spletnemu iskanju



# Bolj splošni boolovi izrazi

- Primer: Določi algoritem za procesiranje naslednjih poizvedb:

***Brutus AND NOT Caesar***

***Brutus OR NOT Caesar***

Kakšen je čas? Še vedno  $O(x+y)$ ?

# Poljubni boolovi izrazi

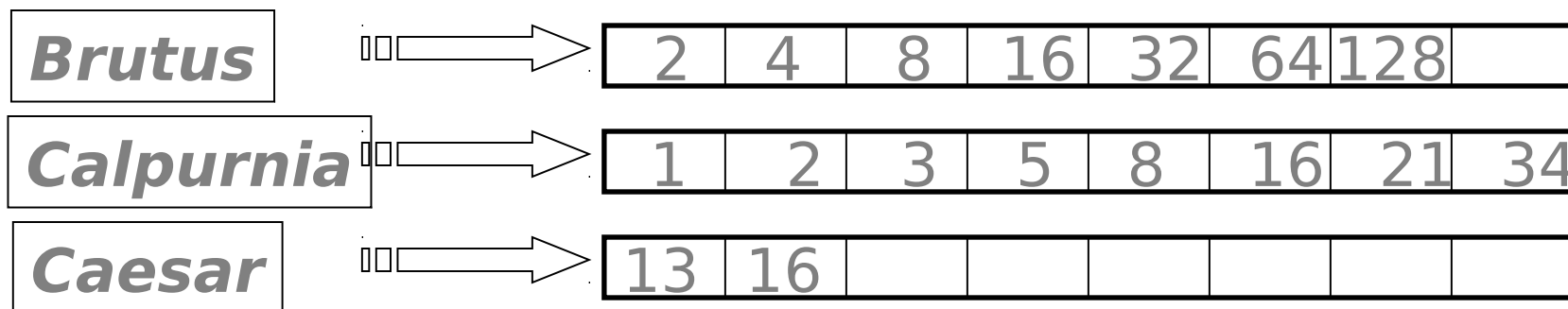
Kaj pa poljuben boolov izraz?

***(Brutus OR Caesar) AND NOT  
(Antony OR Cleopatra)***

- Združjemo v linearnem času?
  - Linearno na kaj?
- Kako narediti boljše?

# Optimizacija vprašanj

- Kakšen je najboljši vrstni red za procesiranje poizvedbe?
  - Naj bo poizvedba AND t besed
- Za vsak od t besed poišči objave in združi sezname



Poizvedba: **Brutus AND Calpurnia AND Caesar**

# Primer optimizacije vprašanj

- Procesiraj v vrstnem redu naraščajočih frekvenc:
  - *Začni z najmanjšo množico, jo poveži z drugo najmanjšo, itd.*

Zato smo shranili  
frekvence v slovarju

<b>Brutus</b>	→	2	4	8	16	32	64	128	
<b>Calpurnia</b>	→	1	2	3	5	8	13	21	34
<b>Caesar</b>	→	13	16						

Izvajanje: (**Caesar AND Brutus**) AND **Calpurnia**.

# Bolj splošna optimizacija

- Primer: (*madding OR crowd*) AND (*ignoble OR strife*)
- Preberi frekvence vseh izrazov.
- Oцени velikost vsakega OR z vsoto frekvenc (konzervativno).
- Procesiraj po naraščajočem vrstnem redu velikosti OR izrazov.

# Več kot boolovo iskanje

- Kaj pa fraze?
- Lokalnost: Poišči ***Gates NEAR Microsoft.***
  - Potreben je indeks, ki hrani pozicijo besed v dokumentih.
- Zone v dokumentih: Poišči dokumente (*author = **Ullman***) *AND* (text contains ***automata***).

# Akumulacija dokaza

- Število pojavitev 1 vs. 0 za iskane dokumente
  - 2 vs. 1 pojavitve
  - 3 vs. 2 pojavitve, itd.
- Potrebujemo frekvenco besed v dokumentih

# Pretvorba teksta v žetone



# Osnovni cevodod indeksiranja

Documents to  
be indexed.



Friends, Romans, countrymen.

⋮

Tokenizer

Token stream.

Friends

Romans

Countrymen

Linguistic  
modules

Modified tokens.

friend

roman

countryman

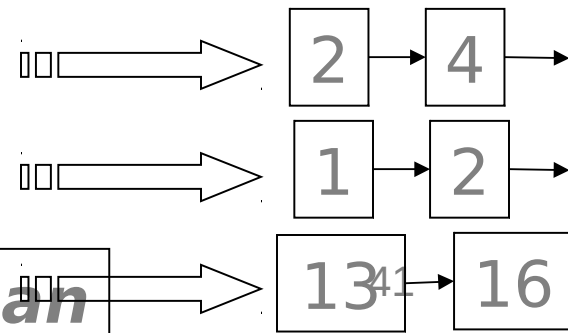
Indexer

Inverted index.

**friend**

**roman**

**countryman**



# Pretvorba v žetone

- Vhod: “*Friends, Romans and Countrymen*”
- Izhod: Žetoni
  - *Friends*
  - *Romans*
  - *Countrymen*
- Vsak žeton je zdaj kandidat za vpis v indeks oz. slovar
- Kaj so veljavni žetoni?

# Razčlemba dokumenta

- V kakšni obliki je dokument?
  - pdf/word/excel/html?
- V katerem jeziku je dokument?
- Kateri nabor znakov uporablja?

Vse zgoraj naštetu je klasifikacijski problem...

# Določitev formata in jezika

- Dokumenti, ki jih indeksiramo so lahko zapisani v različnih jezikih
  - En indeks lahko vsebuje besede večih različnih jezikov
- Včasih lahko dokument in njegove komponente uporablja več različnih jezikov in formatov
  - Francoski mail, ki vsebuje pdf v portugalščini
- Kaj je potem enota dokumenta?
  - Mail? Pripeti dokumenti? Zip?

# Pretvorba v žetone

- Problemi pri pretvorbi:
  - ***Finland's capital*** → ***Finland? Finlands?***  
***Finland's?***
  - ***Hewlett-Packard*** → ***Hewlett*** in ***Packard***  
kot dva žetona?
  - ***San Francisco***: en žeton ali dva? Kako se  
odločiti kaj je žeton?

# Jezikovni problemi

- Poudarki: ***résumé*** vs. ***resume***.
- ***L'ensemble***: eden ali dva žetona? L ? L' ? Le?
- Kitajščina in japonščina nima presledkov med besedami
- Japonščina ima več različnih abeced
- Hebrejščina se bere od desne proti levi; lahko so vrinjeni stavki v drugem jeziku, ki se bere od leve proti desni
- Pomenski problemi: Morgen will ich in MIT ...

# Drugi problemi

- Sinonimi in homonimi
  - *automobil* = *car*
  - Indeksiranje? Razširitev poizvedb?
- Soundex
  - Razred hevristik za pretvorbo besede v fonetske ekvivalente
- Lematizacija
  - Pretvorba besed v osnovno obliko
  - *are, is, am* → *be*; *car, cars, car's, cars'* → *car*

# Krnjenje

- Krnjenje (angl. stemming) je postopek prevoda besed v osnovno obliko primerno za indeksiranje
- Odvisno od jezika !
- Primer:
  - *automate(s), automatic, automation → automat*
- Porterjev algoritem
  - konvencije + 5 faz redukcije (sekvenčno)
  - Vsako fazo sestavlja množica ukazov
    - Primer: *sses → ss, ies → i, ational → ate, tional → tion*
  - Konvencija
    - Primer: *izmed pravil v neki fazi izberi tisto, ki se aplicira na najdaljšo pripono*



# Ocenjevanje rezultatov

# Ocenjevanje

- Do zdaj smo gledali samo Boolova vprašanja:
  - Vsebovanost besede v dokumentih
  - Dobro za eksperte, ki natančno vedo kaj delajo in dobro poznajo izrazoslovje
  - Aplikacije lahko konzumirajo tisoče rezultatov
  - Ni v redu za večino uporabnikov, ki ne znajo dobro formulirati Boolovo vprašanje
  - Večina uporabnikov ne želi pregledovati 1000 rezultatov

# Ocenjevanje

- *Želimo vrniti dokumente v vrstnem redu, ki bo najverjetneje koristen uporabniku*
- Potrebno je izmeriti ujemanje med vprašanjem in dokumenti.
- Kako lahko rangiramo dokumente v korpusu glede na dano poizvedbo?
- Dodelimo točke  $[0,1]$ 
  - Za vsak dokument za vsako vprašanje

# Linearni kombinatorji zon

- Prva generacija metod za ocenjevanje: uporabi linearno kombinacijo Boolovih vprašanj

- Npr.

Score =  $0.6 * \langle \textit{sorting} \text{ in } \underline{\text{Title}} \rangle + 0.3 * \langle \textit{sorting} \text{ in } \underline{\text{Abstract}} \rangle + 0.05 * \langle \textit{sorting} \text{ in } \underline{\text{Body}} \rangle + 0.05 * \langle \textit{sorting} \text{ in } \text{Boldface} \rangle$

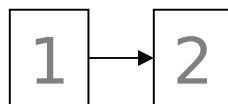
- Vsak izraz kot npr.  $\langle \textit{sorting} \text{ in } \underline{\text{Title}} \rangle$  ima vrednost  $\{0,1\}$ .
    - Skupna ocena je v območju  $[0,1]$ .

# Primer

- Poizvedba: ***bill OR rights***
- Poizvedba vrne naslednje zonske indekse:

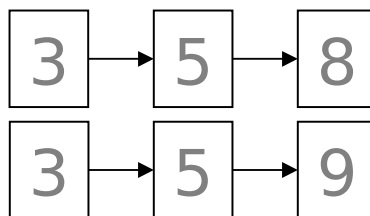
Author

***bill  
rights***



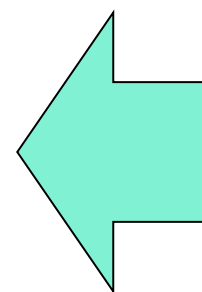
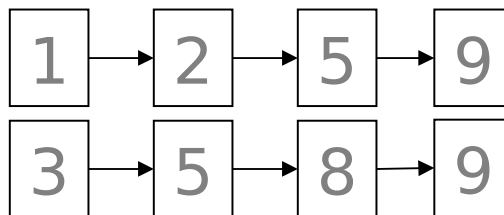
Title

***bill  
rights***



Body

***bill  
rights***



Izračunaj  
oceno  
za vsak  
dokument.  
Uteži:  
0.6,0.3,0.1

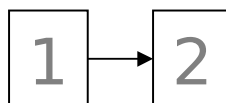
# Splošna ideja

- Imamo vektor uteži s vsoto komponent 1
  - Imamo utež za vsako zono
- Z danim Boolovim vprašanjem dodelimo oceno vsakemu dokumentu tako, da seštejemo prispevke vsake zone
- Tipično -- uporabnik bi želel videti K najbolj ocenjenih dokumentov

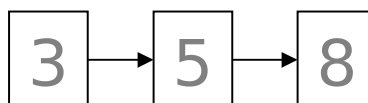
# Indeksi za zone

- Najenostavnejša verzija: imamo ločene indekse za različne zone
- Varianta: en indeks z ločenimi vnosi za vsako besedo in zono
- Primer:

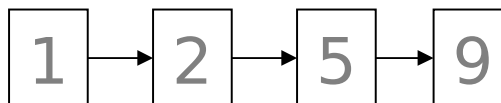
***bill.author***



***bill.title***

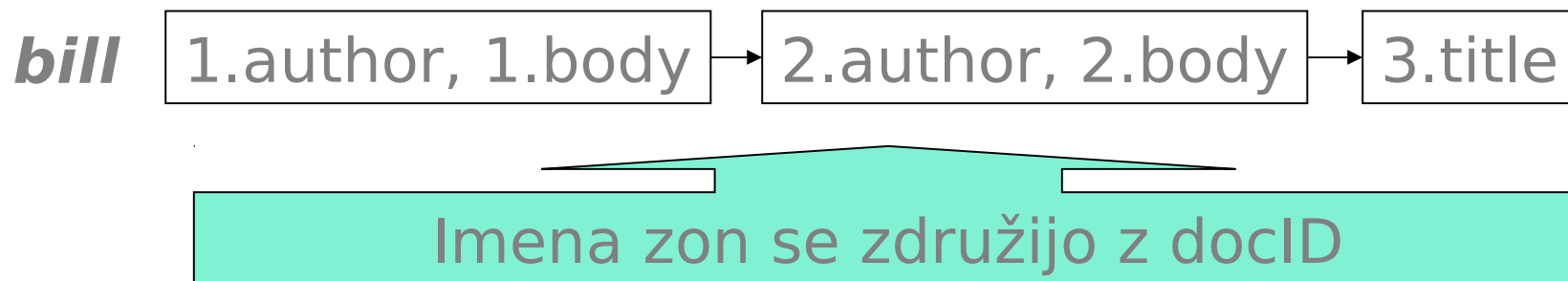


***bill.body***



# Indeks za kombiniranje zon

- Prejšnja rešitev je potratna: vsaka beseda je potencialno replicirana za vsako zono
- Boljša rešitev: zone shranimo med objave



- V času izvajanja se akumulirajo prispevki k zonam in objav



# Vektorji

# Dokumenti kot vektorji

- Dokumente in vprašanja vidimo kot vektorje.
- Vektorski model dokumentov.
  - Model: množica besed
  - Model: vreča besed
- Računali bomo različnost vektorjev
  - Dokument in vprašanje
  - Incidenčna matrika
  - Pogostost besed v dokumentih
  - Razdalje med vektorji
- Naravna mera za ocenjevanje dokumentov
- Ni več boolovih vprašanj !

# Incidenčna matrika

- Dokument (ali zona v dokumentu) je binarni vektor  $X$  v  $\{0,1\}^v$ 
  - Poizvedba je vektor
- Ocena: mera prekrivanja  $|X \cap Y|$

	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

# Primer

- Za vprašanje ***ides of march*** ima Shakespeare-ov *Julius Caesar* oceno 3
- Vse Shakespeare-ove igra imajo oceno 2 (ker vsebujejo ***march***) ali 1
- Vrstni red ocen da igro *Julius Caesar* na vrh

# Mera prekrivanja

- Kaj je narobe z mero prekrivanja?
- Ne upošteva:
  - frekvenco besed v dokumentu
  - Razpršenost izraza v kolekciji
    - ***of*** je bolj pogost kot ***ides*** ali ***march***
  - Dolžina dokumenta
    - (and vprašanja: ocena ni normalizirana)

# Mera prekrivanja

- Normalizacijo lahko naredimo na več načinov:
  - Jaccardov koeficient:

$$|X \cap Y| / |X \cup Y|$$

- Kosinusna mera:

$$|X \cap Y| / \sqrt{|X| \times |Y|}$$

# Ocenjevanje: pogostost besed

- Zaenkrat smo upoštevali pozicijo in prekrivanje besed v dokumentu.
- Očitna je naslednja ideja:
  - če dokument govori več o iskanem izrazu potem se boljše ujema z iskanim izrazom
  - To velja celo v primeru, da imamo samo eno besedo v iskanem izrazu
- Dokument je relevanten, če ima veliko pojavitev komponent iskanega izraza

# Matrike Izraz-Dokument

- Poglejmo si število pojavitev izraza v dokumentu:
  - Model „vreča besed“
  - Dokument je vektor v  $\mathbb{N}^v$ : stolpec spodaj

	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



# Vsota in frekvenca

- V IR literaturi izraz **frekvenca** pomeni „vsota“
  - *tf* pomeni število pojavitev besede v dokumentu
- Dolgi dokumenti imajo predenost, ker je bolj verjetno, da vsebujejo iskani izraz
- Do neke mere lahko to izboljšamo z normalizacijo glede na dolžino dokumenta
  - *tf* deljeno z dolžino dokumenta
- *tf* bo torej „vsota“ !

# Vsota in frekvenca

- Poglejmo si še enkrat poizvedbo ***ides of march***.
  - *Julius Caesar* vsebuje 5 pojavitev ***ides***
  - Nobena druga igra ne vsebuje ***ides***
  - ***march*** se pojavi v več kot 10 primerkih
  - Vse igre vsebujejo ***of***
- Z dano oceno bi bila najbolje ocenjena igra verjetno tista, ki bi imela največ izrazov ***of***.

# Uteževanje frekvence izrazov: $tf$

- Kaj je relativna pomembnost?
  - 0 vs. 1 pojava izraza v dokumentu
  - 1 vs. 2 pojavitve
  - 2 vs. 3 pojavitve ...
- Ni jasno: medtem ko se zdi, da je več bolje, veliko več ni toliko bolj od nekaj.
  - Lahko preprosto uporabljamo osnovno  $tf$
  - Druga opcija, ki se pogosto uporablja v praksi:

$$wf_{t,d} = 0 \text{ if } tf_{t,d} = 0, 1 + \log tf_{t,d} \text{ otherwise}$$

# Izračun ocene

- Ocena za poizvedbo  $q$  = vsota po izrazih  $t$  v  $q$ :
  - $wf$  namesto  $tf$

$$= \sum_{t \in q} tf_{t,d}$$

- [opazka: 0 v primeru  $q=\text{prazen}$ ]
- Predstavljena ocena se da kombinirati z zonami
- Še vedno se ne upošteva porazdelitev zadetkov po posameznih izrazih (**ides** je redkejši od **of**)

# Uteževanje je odvisno od tipa izraza

- Katera izjava pove več o dokumentu?
  - 10 primerkov *hernia*?
  - 10 primerkov *the*?
- Radi bi poudarili težo “splošnih” izrazov
  - Kaj je “splošen izraz”?
- Simselno je uporabiti frekvenco kolekcije (*cf*)
  - Skupno število pojavitev izraza v celotni kolekciji

# Frekvenca dokumenta

- Frekvenca dokumenta (*df* )
- *df* = število dokumentov, ki vsebujejo izraz

izraz	<i>cf</i>	<i>df</i>
<i>try</i>	10422	8760
<i>insurance</i>	10440	3997

- Frekvenca dokumentov/kolekcije so možne samo za znane statične kolekcije
- Kako uporabimo *df* ?

# Mera izrazov $tf \times idf$

- $tf \times idf$  mera kombinira:
  - frekvenca izrazov ( $tf$ )
    - ali  $wf$ , mera za pogostost izraza v dokumentu
  - Inverzna frekvenca dokumenta ( $idf$ )
    - Mera informativnosti izraza; “redkost” izraza po celotnem korpusu
    - Lahko bi uporabili osnovno obliko števila dokumentov v katerih se izraz pojavi ( $idf_i = 1/df_i$ )
    - Vendar se veliko bolj pogosto uporablja:

$$idf_i = \log\left(\frac{n}{df_i}\right)$$

# Povzetek: tf x idf (ali tf.idf)

- Priredi oceno tf.idf vsakemu izrazu  $i$  v vsakem dokumentu  $d$

$$w_{i,d} = tf_{i,d} \times \log(n / df_i)$$

$tf_{i,d}$  = frequency of term  $i$  in document  $d$

$n$  = total number of documents

$df_i$  = the number of documents that contain term  $i$

- Ocena se povečuje s številom pojavitev izraza v dokumentu
- Povečuje se z redkostjo izraza po celotnem korpusu



# Matrika izraz-dokument v $\mathbb{R}$

- Funkcija števila pojavitev besede v dokumentu
  - Vreča besed model
  - Vsak vektor je v  $\mathbb{R}^v$
  - Tukaj uporabljamo *tf.idf*

Lahko  $>1$ !

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	13.1	11.4	0.0	0.0	0.0	0.0
Brutus	3.0	8.3	0.0	1.0	0.0	0.0
Caesar	2.3	2.3	0.0	0.5	0.3	0.3
Calpurnia	0.0	11.2	0.0	0.0	0.0	0.0
Cleopatra	17.7	0.0	0.0	0.0	0.0	0.0
mercy	0.5	0.0	0.7	0.9	0.9	0.3
worser	1.2	0.0	0.6	0.6	0.6	0.0

# Dokumenti kot vektorji

- Vsak dokument  $j$  lahko vidimo kot vektor  $wf \times idf$  vrednosti; ena komponenta za vsak izraz
- Imamo vektorski prostor
  - Izrazi so osi
  - Dokumenti so točke (vektorji)
  - Tudi s krnenjem imamo 20,000+ dimenzij
- Korpus dokumentov nam da matriko, ki jo lahko vidimo kot naš vektorski prostor

# Zakaj bi dokumente spremenili v vektorje?

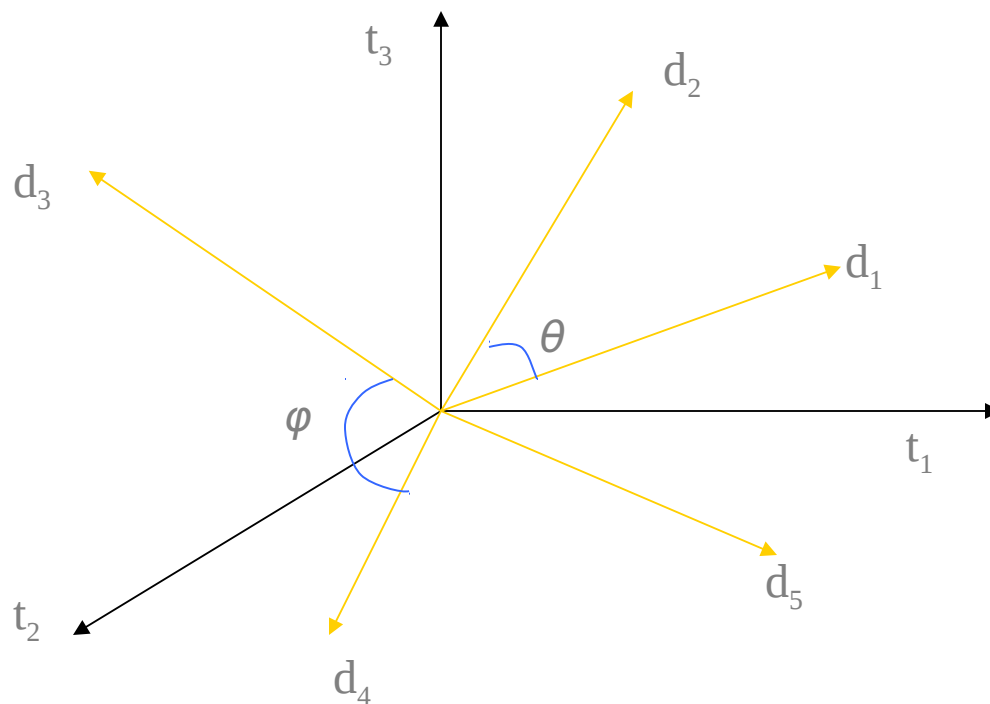
## 1) Query-by-example:

- za dan dokument D poišči vse podobne

## 2) Skupine:

- Poišči vektorje, ki so blizu danega vektorja

# Intuicija



Pravilo: dokumenti, ki so blizu v vektorskem prostoru  
govorijo o podobnih stvareh

# Vektorski prostor

## Vektorji kot poizvedbe:

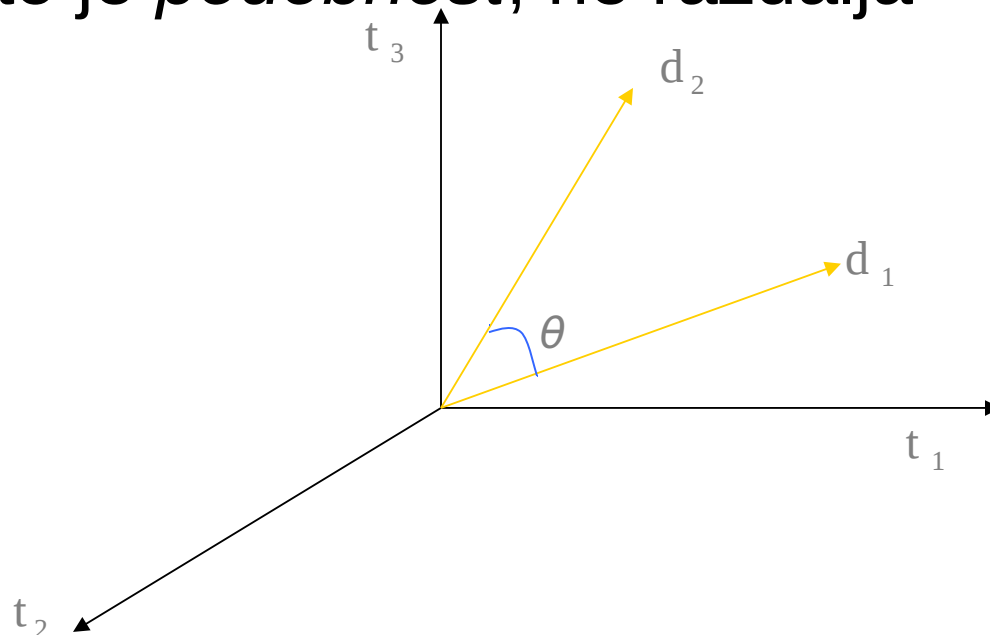
- Poizvedbe gledamo kot kratke dokumente
- Vrnemo dokumente, ki so rangirani po oddaljenosti (bližini) njihovih vektorjev od poizvedbe, ki je tudi predstavljena kot vektor

# Prvi približek

- Razdalja med  $d_1$  in  $d_2$  je dolžina vektorja  $|d_1 - d_2|$ .
  - Evklidska razdalja
- Zakaj to ni dobra ideja?
- Še vedno nismo obdelali normalizacije glede na dolžino dokumenta
  - Dolgi dokumenti so bližje drugim ker imajo več besed
- Lahko pa normaliziramo razdaljo tako, da izračunamo kot !

# Kosinusna mera

- Razdalja med dvem vektorjema se določi s kotom med vektorjema  $d_1$  in  $d_2$ : kosinus kota med vektorjema
- Pozor – to je *podobnost*, ne razdalja



# Kosinusna podobnost

- Vektor je najprej *normaliziran* z deljenjem vsake komponente z dolžino vektorja – pogledjmo si najprej to v dveh dimenzijah

$$\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$$

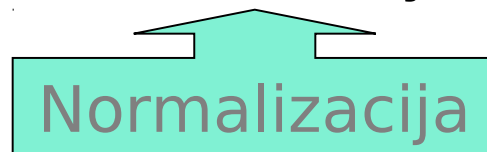
- Preslikava v  $\vec{\text{kroglo}}$  z radijem ena
- Velja,  $|d_j| = \sqrt{\sum_{i=1}^n w_{i,j}} = 1$
- Dolgi dokumenti nimajo večje teže



# Kosinusna podobnost

$$\text{sim}(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{|\vec{d}_j| |\vec{d}_k|} = \frac{\sum_{i=1}^n w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^n w_{i,j}^2} \sqrt{\sum_{i=1}^n w_{i,k}^2}}$$

- Kosinus kota med vektorjema
- Imenovalec vsebuje dolžine vektorjev



Normalizacija

# Normalizirani vektorji

- Za normalizirane vektorje je podobnost preprosto:

$$\cos(d_j, d_k) = d_j \cdot d_k$$

# Google

# Karakteristike iskalnika

- Vsak lahko dodaja vsebino
  - Vprašanja kvalitete; Spam
- Različni tipi podatkov
  - Telefonski imenik, brošure, katalogi, disertacije, novice, vreme, vse na enem mestu!
- Različni uporabniki
  - Lexis-Nexis: profesionalni iskalci
  - Online katalogi: študenti iščejo šolsko literaturo
  - Splet: vsi tipi ljudi in vse vrste ciljev
- Obseg
  - Stotine milijonov poizvedb na dan; bilijoni dokumentov

# Spletne poizvedbe

- Spletne poizvedbe so **kratke**:
  - ~2.4 besede v povprečju
  - Je bilo manj 1.7 (~1997)
- **Pričakovanja uporabnikov**:
  - Mnogi pravijo “**prvi zadetek bi moral biti tisto kar iščem!**”
  - To deluje, če ima uporabnik najbolj popularno/skupno idejo in ne sicer

# Kaj pa ocenjevanje?

- Tu imamo več variacij
  - Velikokrat zmešnjava; podrobnosti zaščitene/varabilne
- Kombinacija podmnožice:
  - **IR-stil ocene**: osnovano na frekvenci izraza, podobnosti, pozicijo v tekstu (npr. v naslovu), font, itd.
  - **Informacija o popularnosti**
  - **Informacije o analizi povezav**
- Večinoma se uporablja verzijo ocenjevanja na osnovi vektorskega prostora + kombinacije:
  - Izračunaj IR oceno
  - Množenje ocene z ocenami za vsako posebno lastnost

# Posebne ocene

- “Popularnost” strani (npr. direkten zadetek)
  - Pogosto obiskane strani
  - Pogosto obiskane strani preko poizvedbe
- “Citiranje” povezav (Google)
  - Katere naslove citirajo drugi?
  - Raziskave v sociologiji o bibliografskih citatih za identifikacijo “avtorativnih virov”

# Indeksiranje v Google

- *Indekser* pretvori vsak dokument v kolekcijo “**zadetkov**”, ki jih uvrsti v sode (barrel) sortirane po docID (**objave**). Kreira tudi zbirko povezav (links).
  - **Zadetek**: <wordID, position in doc, font info, hit type>
  - **Tip zadetka**: **plain** ali **fancy**.
  - **Fancy zadetek**: URL, naslov, tekst povezave, meta-oznaka.
  - Optimizirana predstavitev zadetkov (2 zloga)
- *Sortirnik* uredi vsak sod glede na wordID za kreacijo **invertiranega indeksa**. Kreira tudi datoteko lexicon (**slovar**).
  - **Lexicon**: <wordID, offset into inverted index>
  - Lexicon je običajno v dinamičnem spominu

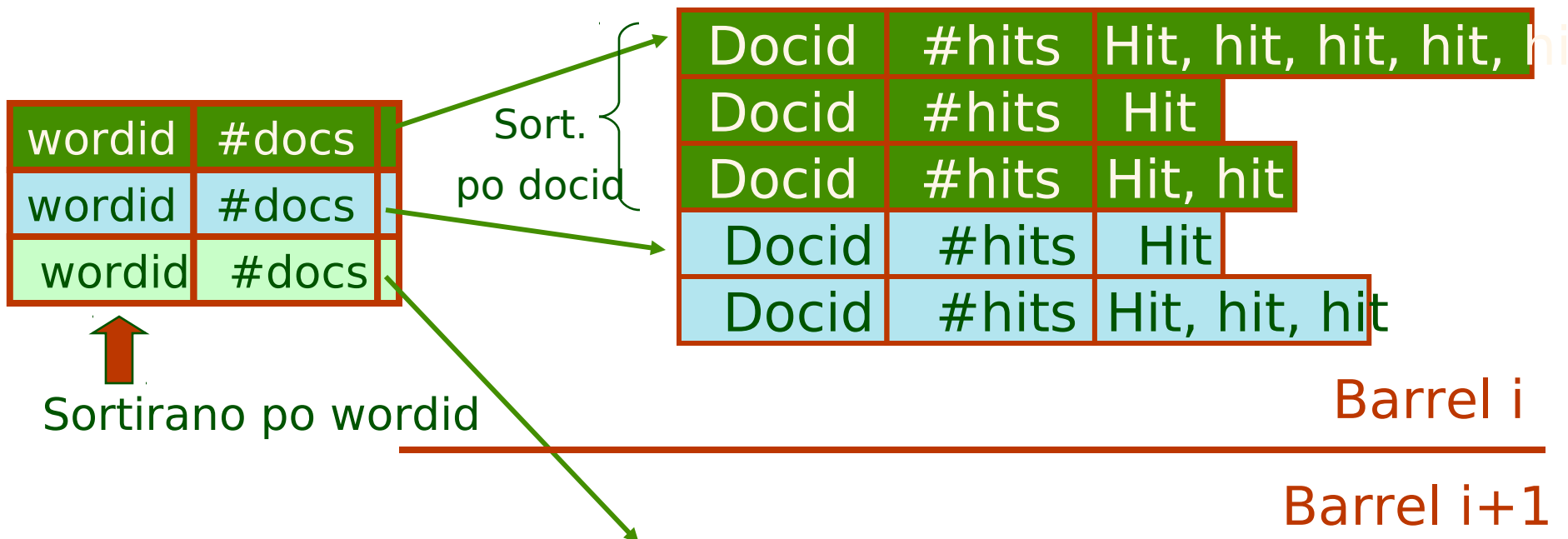


# Google-ov invertirani indeks

Vsak "sod" vsebuje objave zaporedja besed.

Lexicon, v spominu

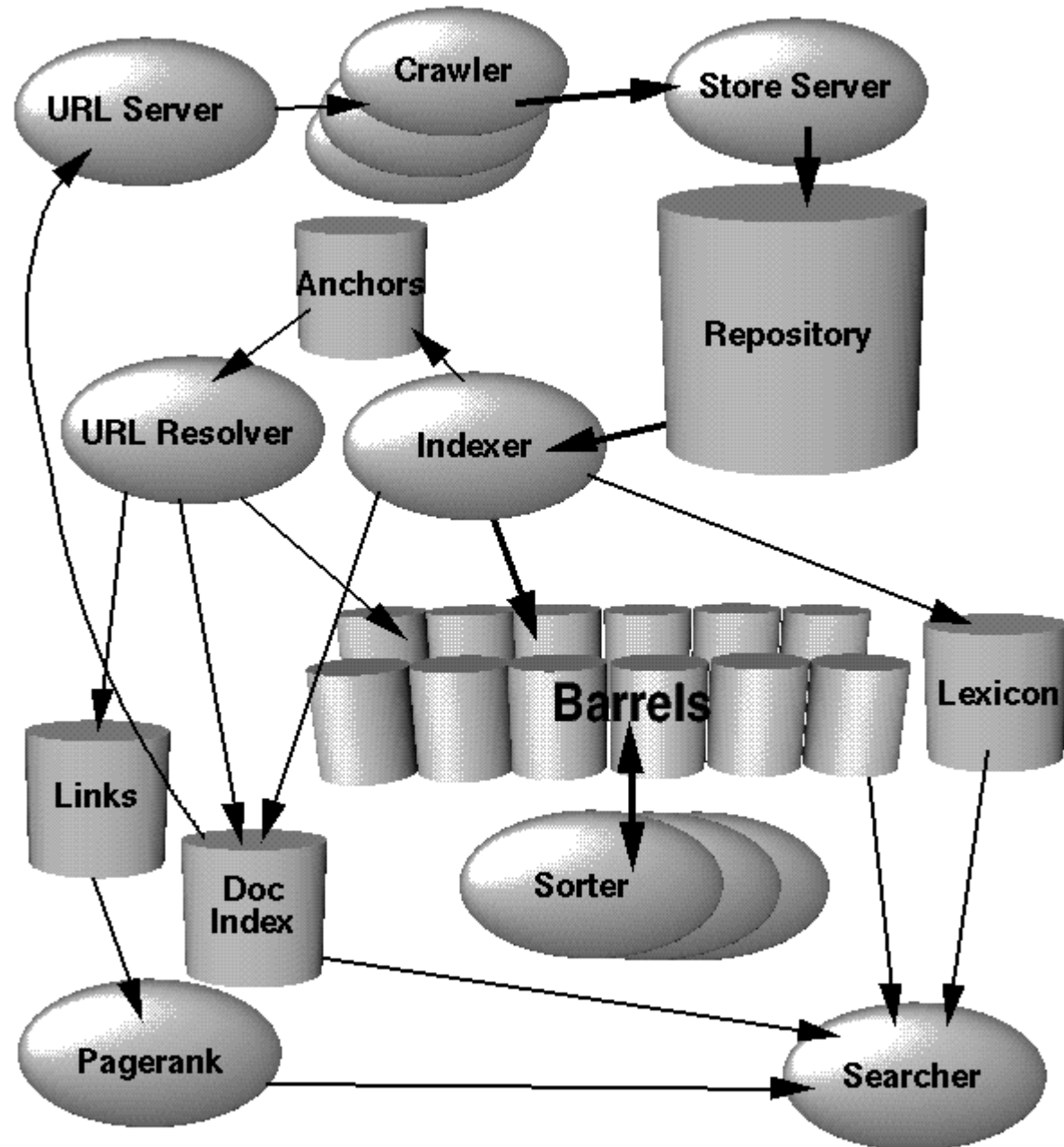
"invertirani sodi", na disku



# Google

- Sortirani sodi = invertiran indeks
- Ocena strani je izračunana iz strukture povezav; kombinirano z IR oceno
- IR ocena je odvisna od TF, tipa zadetka, bližine, itd.
- Bilijon dokumentov
- 100 milijonov poizvedb/dan
- AND poizvedbe

npb 12/13



# Analiza povezav za ocenjevanje

- **Predpostavka:** če so strani, ki kažejo na dano stran dobre je dobra tudi dana stran
  - Reference: Kleinberg 98, Page et al. 98
- Raziskave v sociologiji in biblio-metriki
  - Kleinberg-ov model vsebuje “avtoritete” (zelo referencirane strani) in “hub-e” (strani, ki vsebujejo kvalitetne reference)
  - Googlov model ne vsebuje hubov in je zelo blizu delu na vplivih uteži: Pinski-Narin (1976).

# Ocena strani

- Naj bodo  $A_1, A_2, \dots, A_n$  strani, ki kažejo na stran  $A$ . Naj bo  $C(P)$  # povezav iz strani  $P$ . Ocena strani (pagerank=PR)  $A$  je definirana:

$$PR(A) = (1-d) + d ( PR(A_1)/C(A_1) + \dots + PR(A_n)/C(A_n) )$$

- Fiksna točka zgornje enačbe

# Uporabniški model

- Ocena strani je porazdelitev verjetnosti po straneh spleta: vsota vseh verjetnosti je ena
- **Uporabniški model:**
  - “naključni surfer” izbere stran in sledi povezavam dokler se ne naveliča: potem naključno izbere drugo stran in nadaljuje
  - $\text{PageRank}(A)$  je verjetnost, da uporabnik obišče stran  $A$
  - $d$  je verjetnost, da se naveliča na dani strani
- Google izračuna oceno strani:
  - Prvo izračuna IR oceno strani
  - Popravi IR oceno z uporabo PageRank za izračun “top” strani