Pretraživanje struktuiranih sadržaja

Dragan Ivanović
dragan.ivanovic@uns.ac.rs

Katedra za informatiku, Fakultet tehničkih nauka, Novi Sad

2015.

Vrste IR po sadržajima u kolekciji

- Pretraga tekstualnih sadržaja
  - nestrukturiranih sadržaja
  - strukturiranih tekstualnih sadržaja koji iako imaju strukturu u nekim poljima svoje strukture imaju velike količine tekstova
    - Pretraga po parametrima i zonama - *Parametric and zone search*
    - Pretraga složenijih struktura - može i neka druga struktura, ali je to najčešće XML
- Pretraga linkovanih tekstualnih sadržaja (pretraga veba)
- Pretraga multimedijalnih sadržaja: slika, zvuk, video
- Pretraga ostalih vrsta sadržaja: izvornih programskih kodova, 3D objekata, itd.

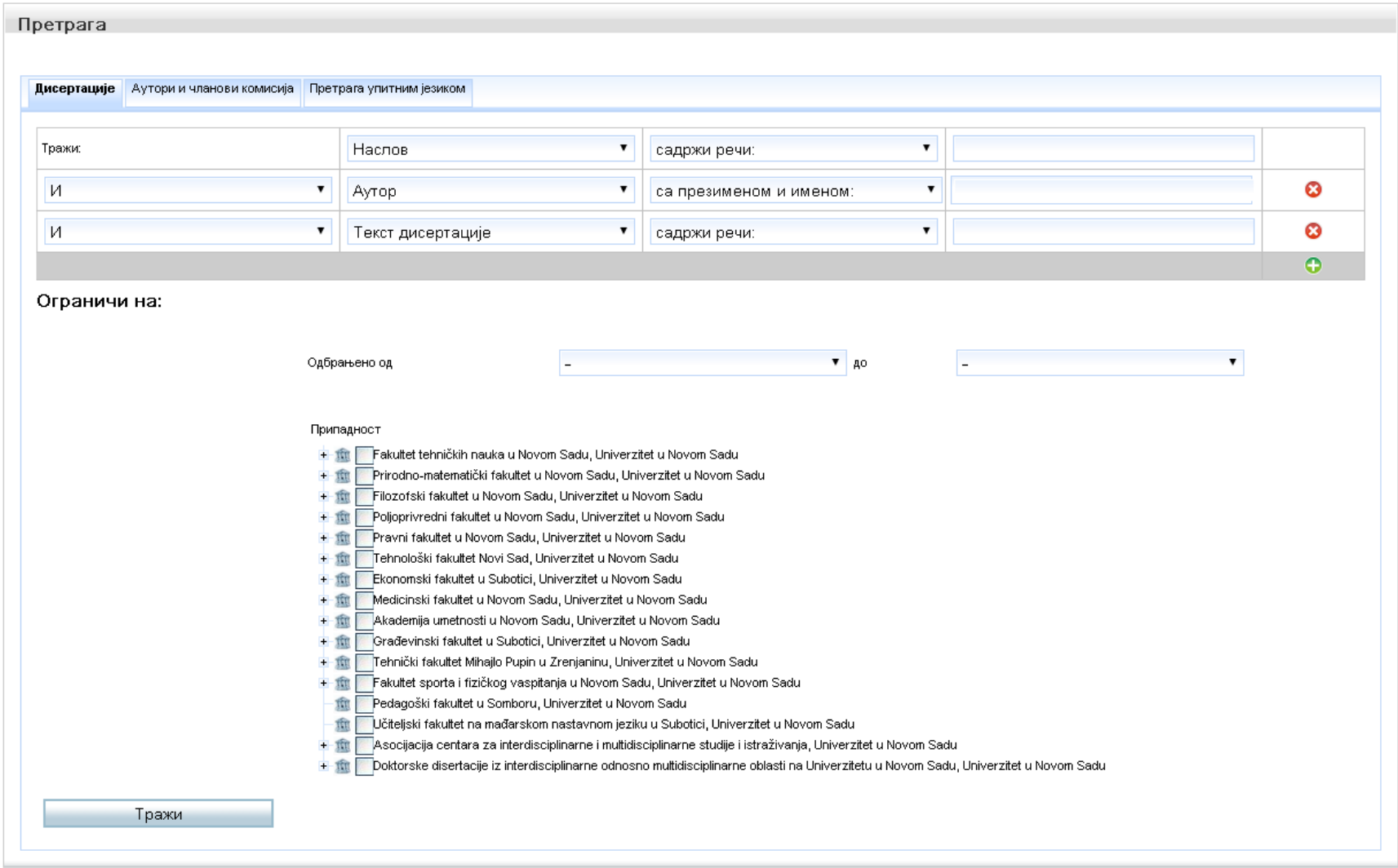
Pretraga struktuiranih tekstualnih sadržaja

- Nešto između pretraga baza podataka i IR nestruktuiranih sadržaja
- Postoji struktura, ali su elementi u strukturi bogati tekstualnim sadržajima, (dugački teksualni sadržaji) pa je zgodno imati *IR feature*-e kao što je normalizacija upita i tekstova, relevantnost u odnosu na informacionu potrebu, a ne upit, sortiranje odgovora po relevantnosti, itd.
- U pretrazi se kombinuju tekstualni kriterijumi i strukturalni kriterijumi
- U literaturi se ponekad koristi i termin *semistructured retrieval* da bi se razlikovalo od pretrage baze podataka

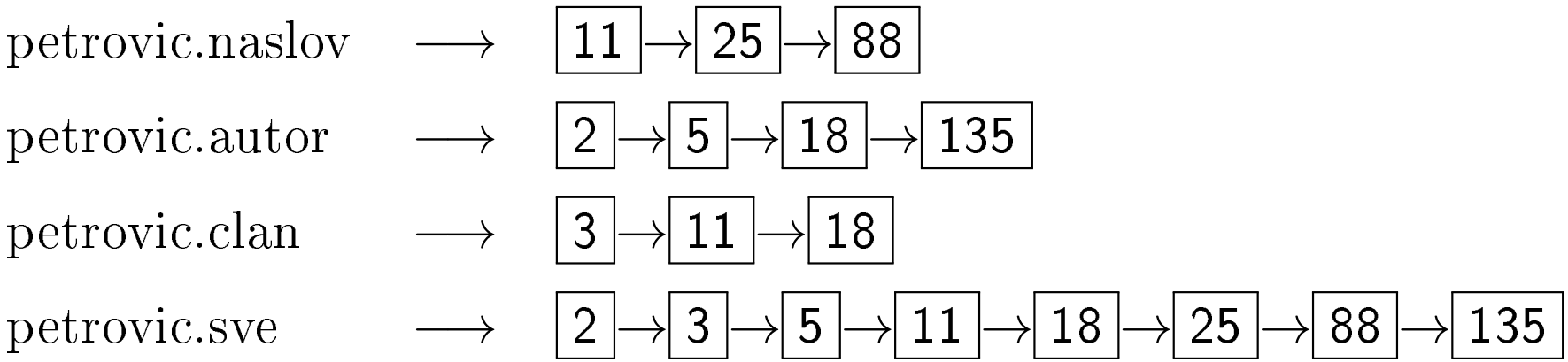
Osnove pretrage po parametrima i zonama

- Parametri su: datum izmene, pripadnost nekoj grupi, geografska pripadnost, redni broj, itd.
- Zone su: naslov, apstrakt, uvod, ključne reči, itd.
- U terminologiji Lucene-a
  - I parametri i zone su Field-ovi
  - Parametri su obično Field-ovi koji nisu analizirani (procesirani, pušteni kroz Analyzer), ali jesu indeksirani
  - Zone su uvek Field-ovi koji su i analizirani i indeksirani
- Za razliku od pretrage XML struktuiranih sadržaja
  - Zna se šta je dokument koji se indeksira i koji je rezultat pretrage
  - Nema hijerarhije
  - Manje atributa i čvorova (zona) nego kod XML struktuiranih sadržaja

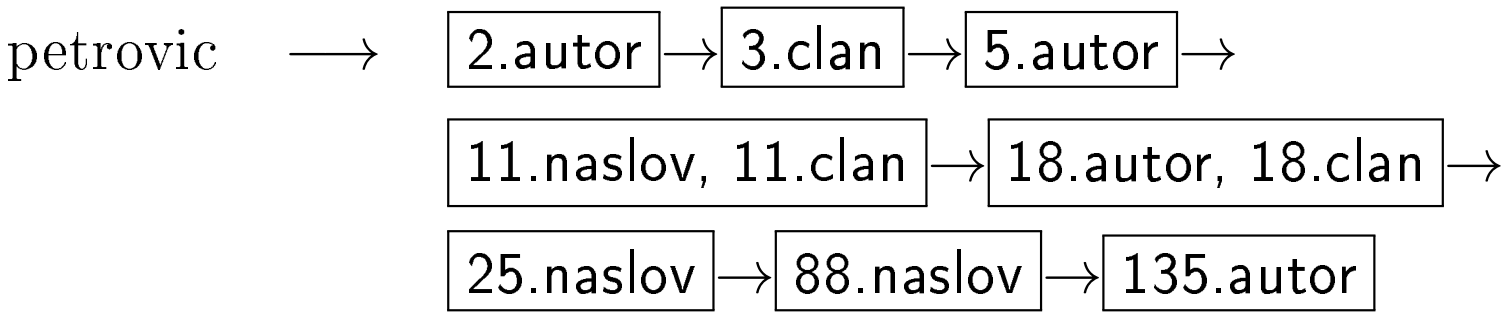
Primer forme



Indeksi - varijanta 1



Indeksi - varijanta 2



Dodela težine zonama

- Nekoj zoni možemo dati veću težinu
- Na primer: naslov ima veću težinu nego autor koji ima veću težinu nego član komisije
- Ako je upit *Petrović* relevantnija je disertacija koja u naslovu ima ovu reč od disertacije čiji je jedan član komisije sa ovim prezimenom
- Ali uzima se u obzir i tf-idf: ako jedan dokument ima *Petrović* u naslovu jedanput a u tekstu isto samo jedanput, a drugi dokument nema u naslovu ovu reč, ali je spominje 1.000 puta u tekstu šta je relevantnije?
- Težine su samo koeficijenti sa kojima se množe mere koje označavaju relevantnost (tf-idf)
- Težine određuju projektanti IR sistema, korisnici ili se koriste *machine learning* tehnike za utvrđivanje težina
- U terminologiji Lucene-a težine se zovu Boost-ovi i postoje odgovarajuće metode sa postavljanje boost-a određenim

Data-centric / document-centric XML

- Data-centric XML: uglavnom kao sredstvo za komunikaciju između aplikacija
  - najćešće sadrže podatke iz relacione baze
- Document-centric XML: "polustrukturirani sadržaj"
  - bogat tekstom
  - potreba za pretraživanjem "IR kvaliteta"
  - npr. "nađi ISBN brojeve knjiga u kojima se bar tri poglavlja bave proizvodnjom kafe, rangirane po ceni knjige"

Osnovni pojmovi

- Element
  - otvarajući i zatvarajući tag
  - proizvoljna hijerarhija
  - nema ukrštanja elemenata, odnosno tagova, zna se ko kome pripada
- Atribut
  - jedan element može (ali ne mora) imati jedan ili više atributa
  - imaju ime i vrednost
  - navode se u otvarajućem tagu
- Tekstualni *leaf nodes*
- DOM - document object model
- DTD, XML Scheme - dva standardizovana načina za opis šeme XML dokumenta
- XPath - standard za izraze kojima se vrši selekcija node-a

XML IR princip

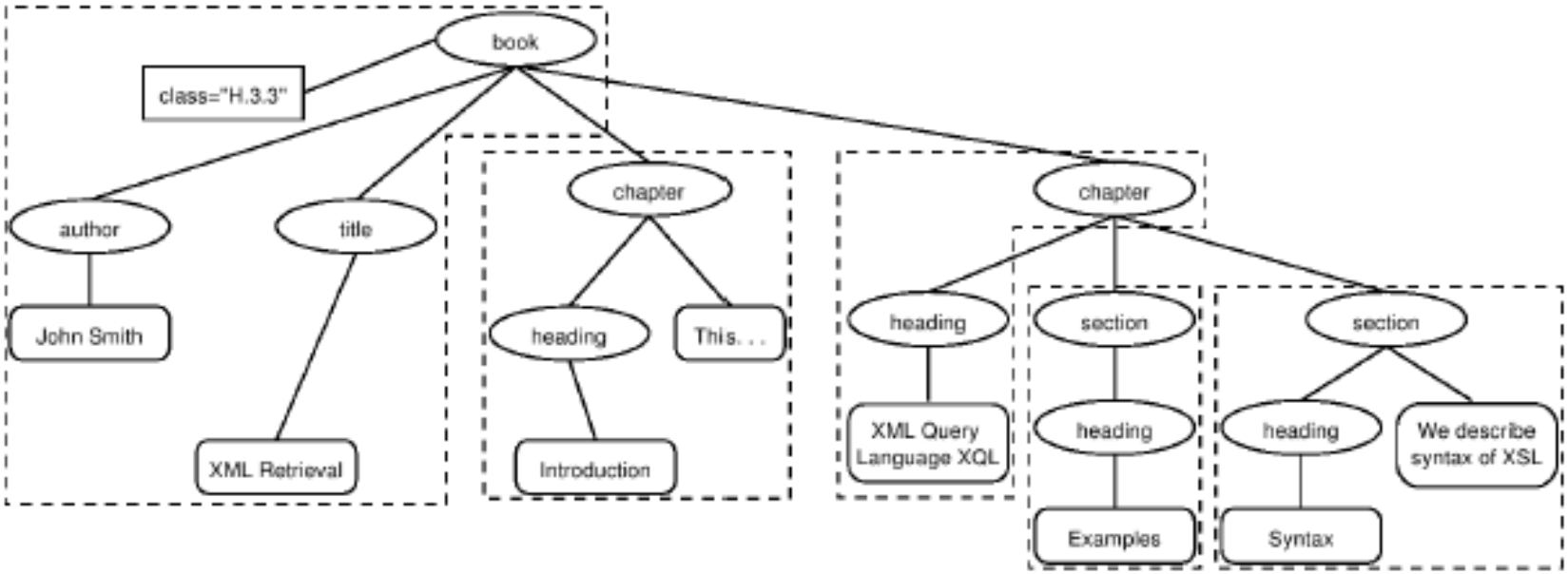
- **Structured document retrieval principle.** *A system should always retrieve the most specific part of a document answering the query.*
- Nije uvek tako prosto
  - Upit: title:Macbeth
  - Naslov cele tragedije Macbeth i naslov Act I, Scene vii, Macbeth's castle su oboje relevantni jer sadrže term Macbeth
  - Šta vratiti kao unit odgovora, celu tragediju ili samo Scenu vii?
  - U nekim situacijama ne treba vratiti manji unit nego veći

XML IR problemi - units

- Šta je *indexing unit* i *result unit*
- Najveći *node* je *indexing unit*
  - post-procesiranje rezultata, ako je jedan veliki *node* odgovor, onda ulazimo u njega i gledamo koji to njegov deo treba prikazati kao odgovor
  - najrelevantniji veliki *node* često ne sadrži najrelevantniji *subelement* (on može biti u nekom čvoru koji nije ocenjen kao mnogo relevantan)
- Najmanji *node* - *leaf* je *indexing unit*
  - često ne dovoljno informativni odgovori (trebalo bi ih proširiti)
  - multicipliranje odgovora
- Svi *node*-ovi su *indexing unit*
  - post-procesiranje rezultata, ako je jedan mali *node* odgovor, onda ga proširijumo da bi dobili nešto što treba prikazati kao odgovor
  - najrelevantniji mali *node* često ne pripada najrelevantnijem odgovoru

XML IR problemi - units

- Deljenje *node*-ova na nepreklapajuće *indexing unit*-e
  - odgovori nisu koherentni i često zbunjuju korisnike



- Od *indexing unit* nam zavisi kako računamo tf i idf

XML IR problemi - heterogenost šema

- Idealno bi bilo da
  - postoji samo jedna šema
  - korisnik razume šemu
- U praksi: jako retko
  - postoji više šema
  - šeme nisu poznate unapred
  - šeme se menjaju
  - korisnici ih ne razumeju
- Potrebno je identifikovati slične elemente u različitim šemama i proširiti upit
  - primer: podaci o zaposlenima

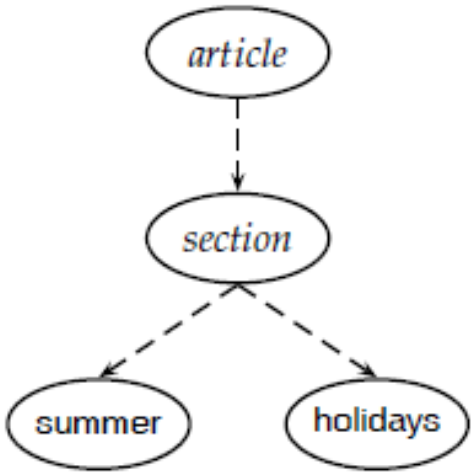
XML IR problemi - korisnički interfejs

- Omogućiti korisniku da pronađe relevantne čvorove
  - author, editor, contributor, sender
- Koji upitni jezik vidi korisnik?
  - specifičan XML upitni jezik? ne
  - forme? parametrizovani upiti?
  - textfield/textarea?
- U principu: poseban sloj između korisnika i XML-a
- Keyword-based search on structured data sources

NEXI

- *Narrowed Extended XPath I*
- Standardni format za XML upite
- *relational attribute constraints* i ***ranking constraint***

```
//article
[.//yr = 2001 or .//yr = 2002]
//section
[about(.,summer holidays)]
```



Strukture podataka za pretraživanje XML-a

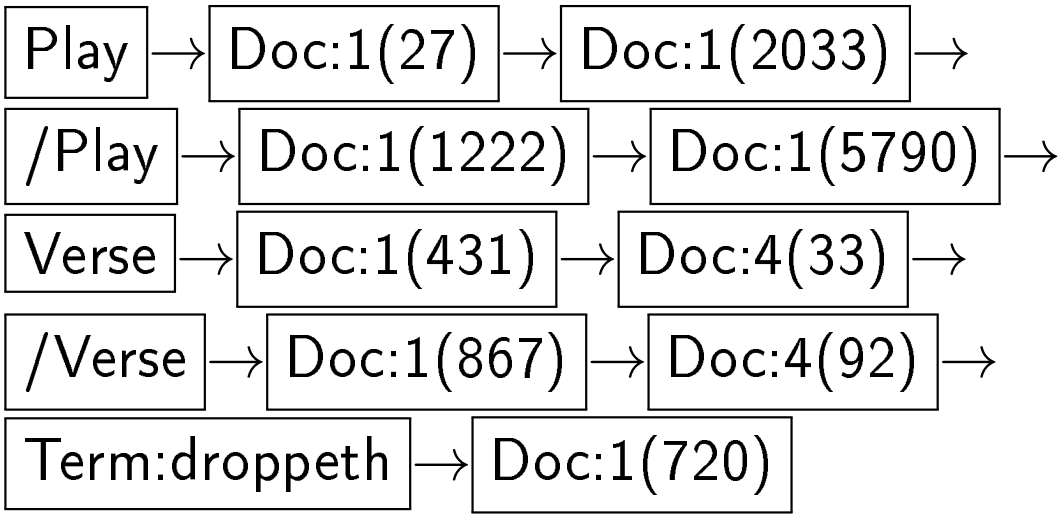
- Tekstualne pretrage: daj mi sve elemente koji zadovoljavaju tekstualni upit Q
  - nije teško: tretiraj svaki element kao poseban dokument u invertovanom indeksu
- Pretrage po strukturi: daj mi sve elemente koji su deca book elementa
- Kombinacija prethodna dva

Veze roditelj/dete

- Dodeliti broj svakom elementu
- Održavati listu veza roditelj/dete
  - npr: Chapter:21 $\leftarrow$ Book:8
  - jednostavno za neposrednog pretka
- Ali upit: „reč Hamlet ispod Scene elementa ispod Play elementa“?

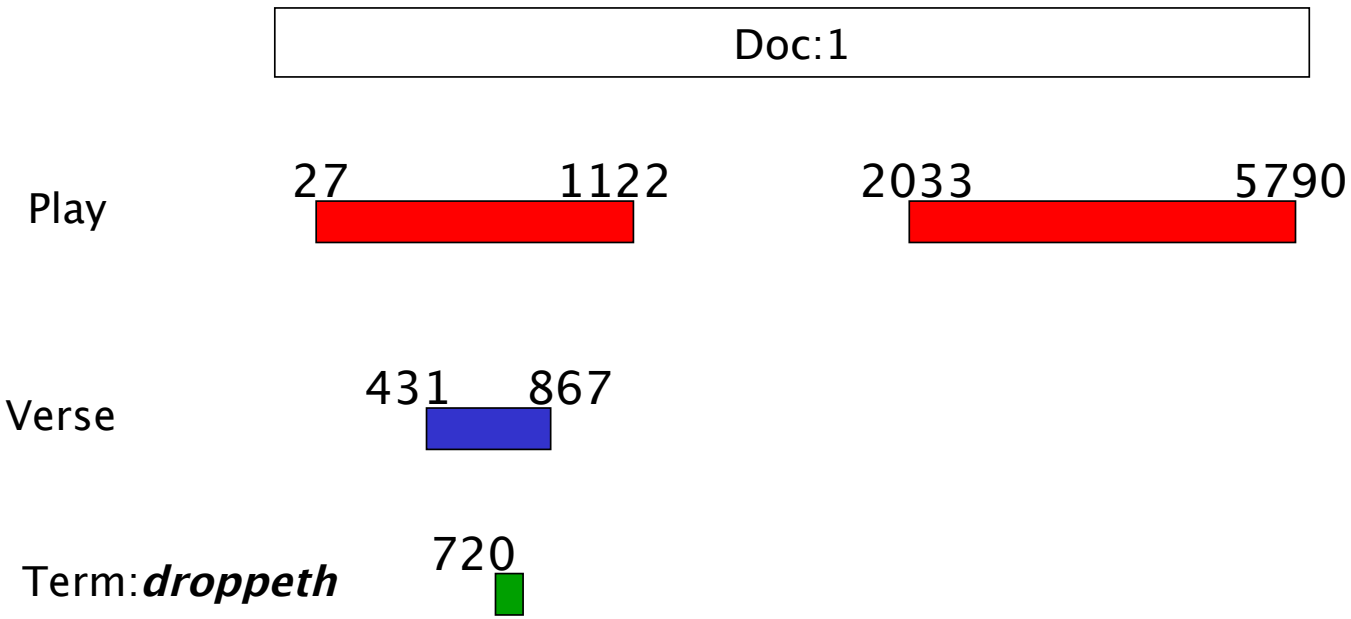
Uopšteni pozicioni indeks

- Posmatraj XML dokument kao tekstualni dokument
- Napravi pozicioni indeks za svaki element
  - označi početak i kraj svakog elementa, npr:



Sadržavanje i pozicija

- Sadržavanje se može posmatrati kao spajanje (merge) pojava



Uopšteni pozicioni indeks

- Sadržavanje podelemenata se može rešiti pozicionim invertovanim indeksom
- Pretraživanje podrazumeva „spajanje“ pojava
- Komplikacije nastaju prilikom dodavanja/uklanjanja elemenata

„Text-centric“ pretraživanje XML-a

- Duži tekstualni dokumenti tagovani XML-om
  - tehnička uputstva, časopisi, . . .
- Upiti predstavljaju informacione potrebe
  - daj mi <section> u kome se objašnjava kako se menja sijalica

Vektorski model i XML

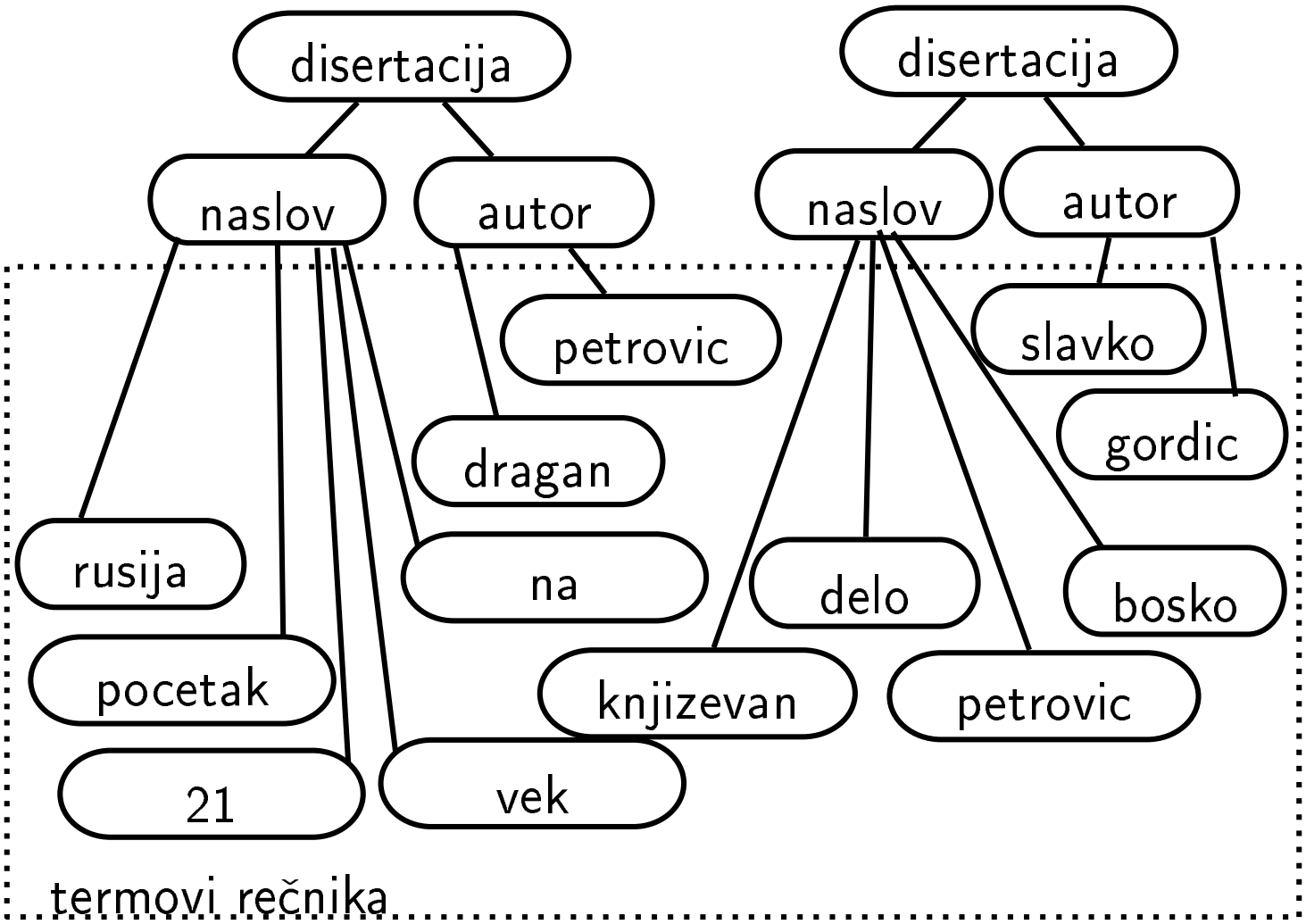
- Vektorski model: potvrđen u praksi, zasnovan na ključnim rečima
  - druge primene: klasifikacija, klasterovanje, ...
- Možemo li ga upotrebiti za „text-centric“ pretraživanje XML-a?
- Osnovni problem: prikazati strukturu XML dokumenta u vektorskom prostoru

Vektorski model i XML

- Trebalo bi napraviti razliku između ova dva slučaja

Vektorski model i XML

- Trebalo bi napraviti razliku između ova dva slučaja

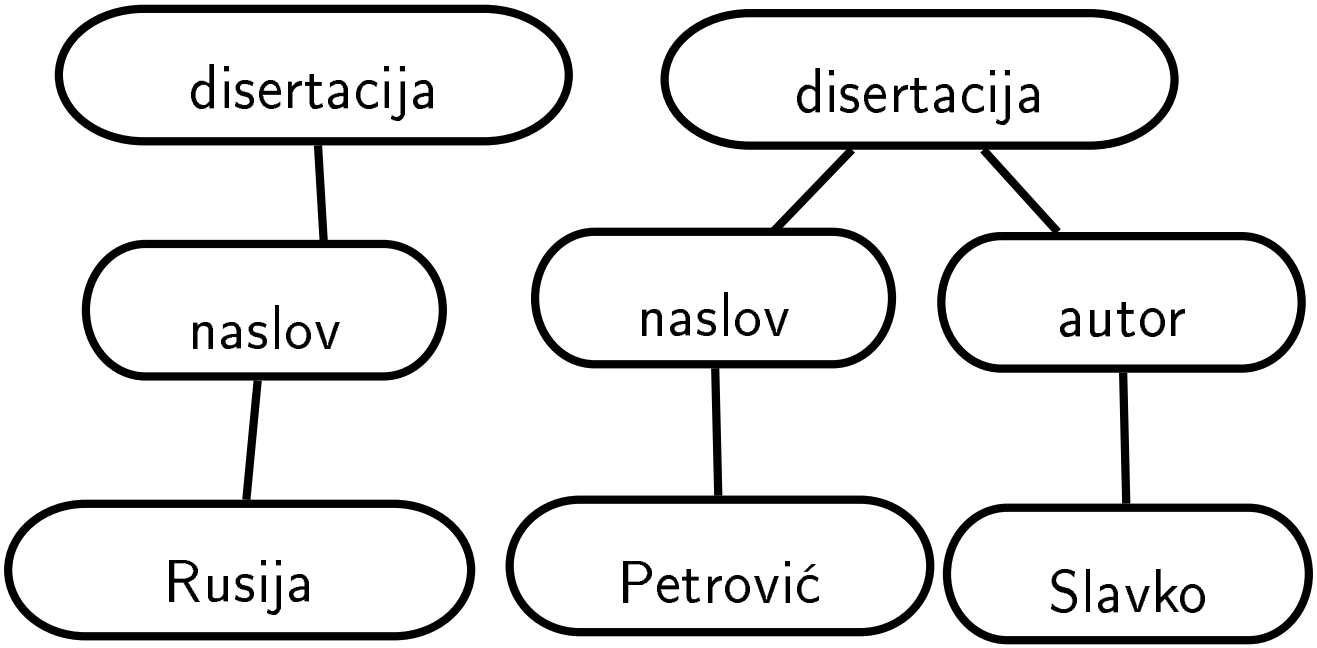


Indeksirati Gates različito

- Ose klasičnog vektorskog prostora su termovi
- Postojala bi jedna osa za term petrovic
- Sada treba da razdvojimo njeno pojavljivanje u različitim elementima, autor i naslov
- Ose moraju da opišu ne samo term nego i njegov položaj u stablu dokumenta

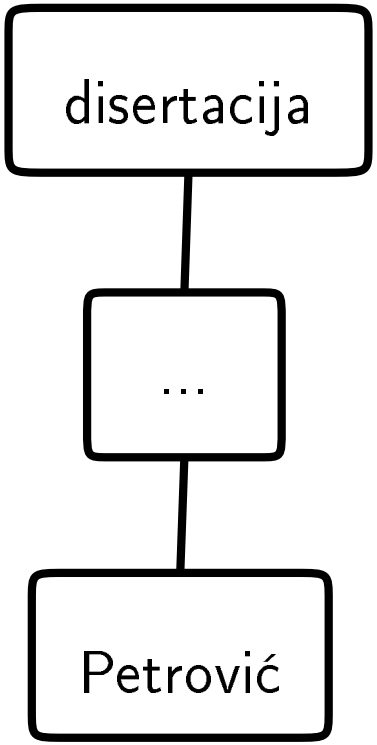
Upiti

- Da razmotrimo koje vrste upita ćemo obrađivati
- Upit kao podstablo dokumenta



Vrste upita

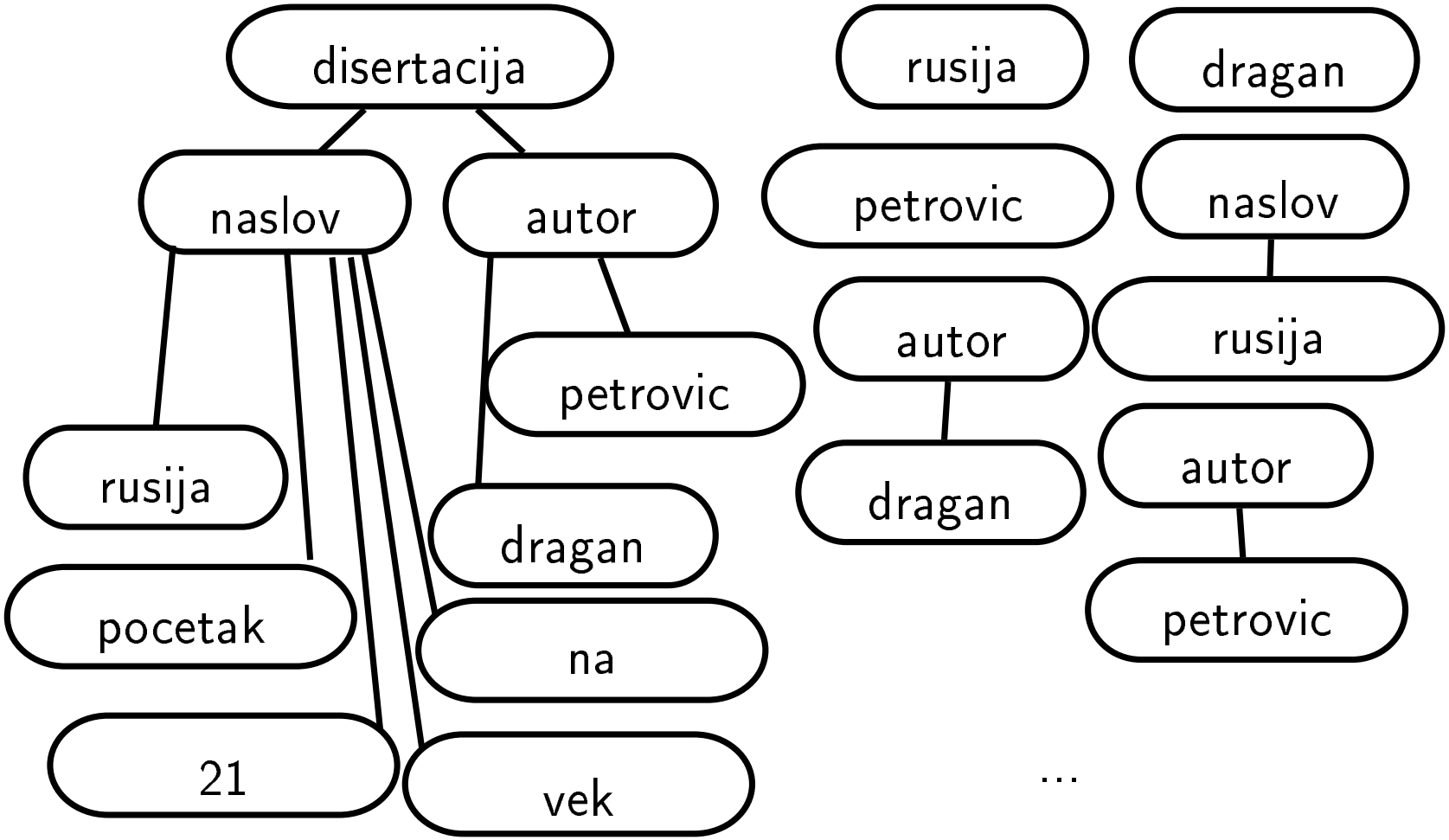
- Prethodni upit je predstavljao traženje podstabla
- A ovaj upit:



- Petrović negde ispod disertacija

Podstabla i struktura

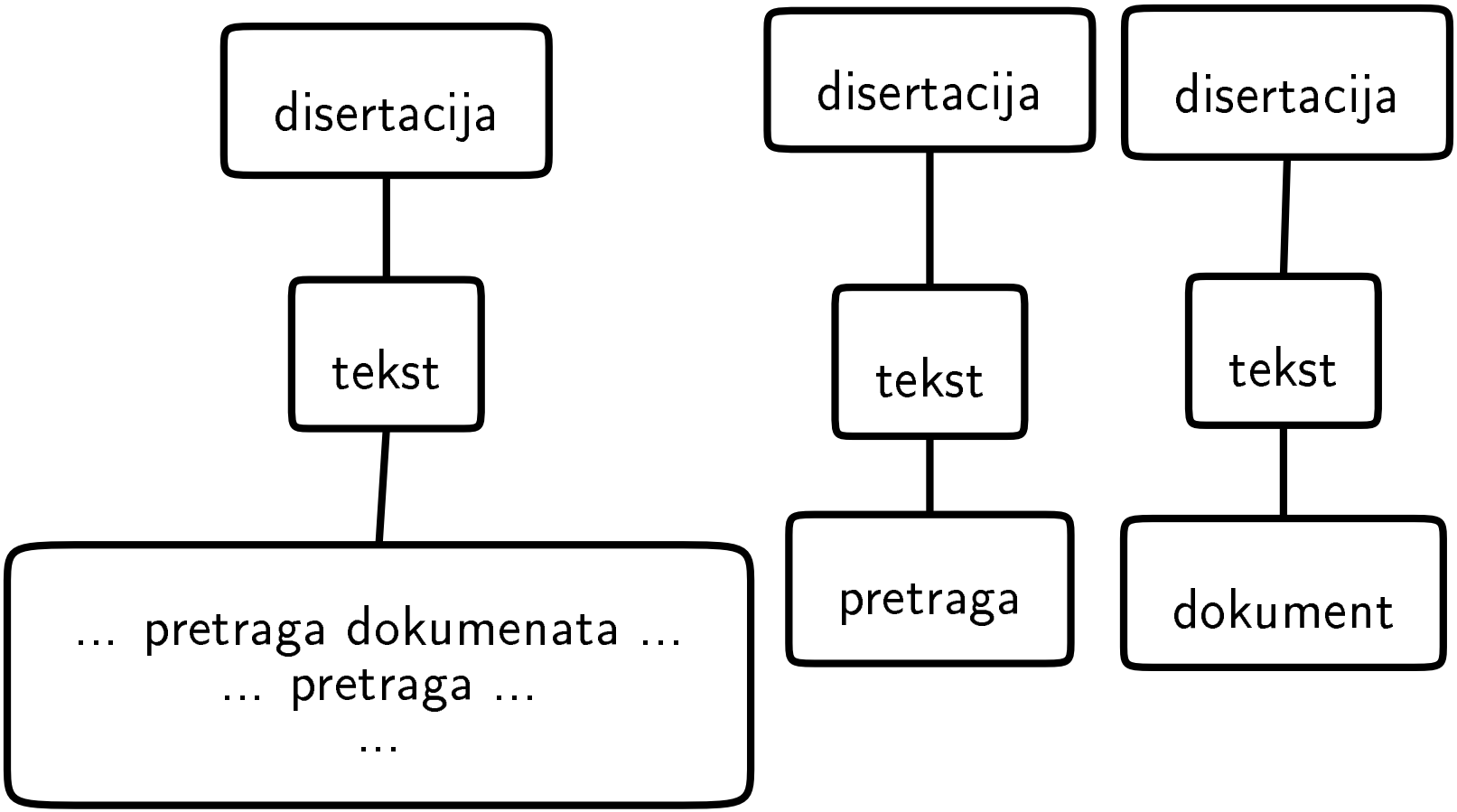
- Posmatramo sva podstabla koja sadrže bar jedan term rečnika



Strukturni termovi

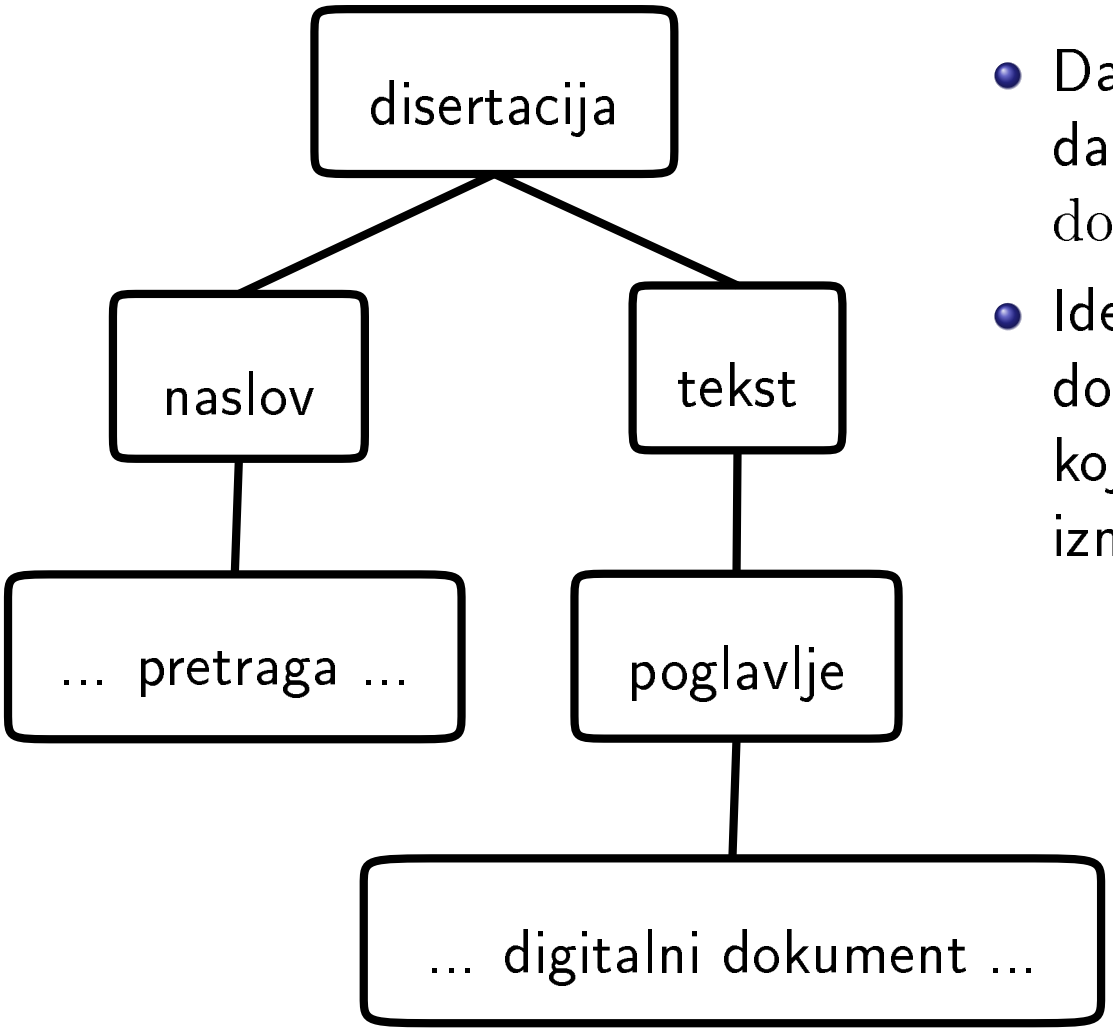
- Nazovimo sva podstabla strukturnim termovima
- Strukturni termovi se mogu pojaviti više puta u dokumentu
- Definisati po jednu osu u vektorskom prostoru za svaki različiti strukturni term
- Težine definisati prema broju pojavljivanja (slično tf)
- Sve uobičajene operacije nad termovima (lowercase, stemming, itd) ostaju

Primer tf težine



- Strukturni term koji sadrži **pretraga** ima veću težinu nego strukturni term koji sadrži **dokument**

Smanjivanje težina



- Da li bi pretraga trebalo da ima veću težinu nego dokument?
- Ideja: pomnožićemo tf doprinos terma t čvoru koji se nalazi k nivoa iznad sa γ^k za neko $\gamma < 1$

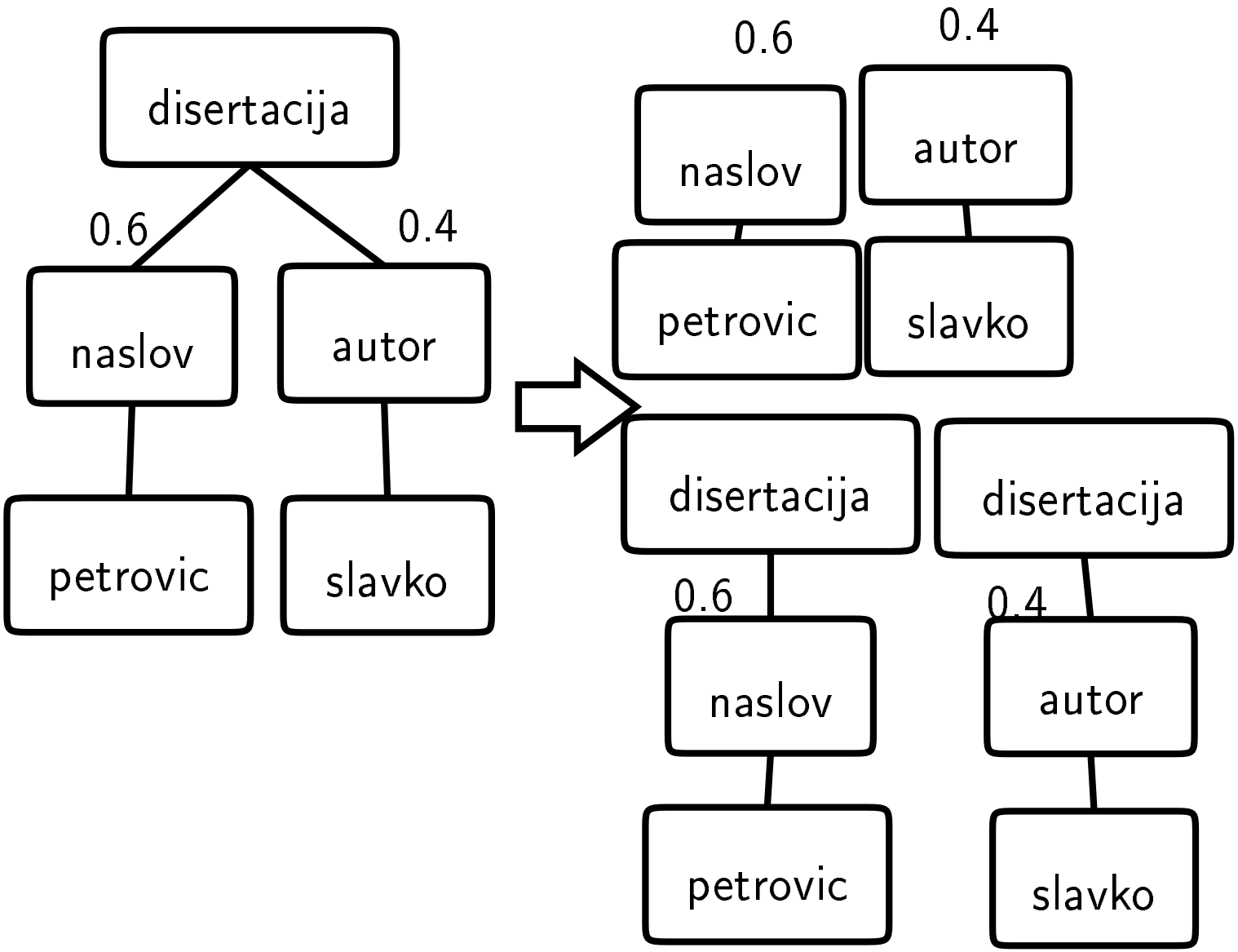
Smanjivanje težina za $\gamma = 0.8$

- Za prethodni dokument, tf težina terma
  - pretraga se množi sa 0.8
  - dokument se množi sa $0.8^2 = 0.64$
- ...za svaki strukturni term sa korenom disertacija

Strukturni termovi: dokumenti i upiti

- Pojam strukturnog terma ne zavisi od šeme dokumenta
- Zgodno za heterogene kolekcije XML dokumenata
- Dokumente predstavljamo kao vektore u prostoru strukturnih termova
- Upit se takođe može rastaviti na strukturne termove
  - i prikazati kao vektor
  - sa mogućim različitim težinama za delove upita

Primer upita



Propagacija težina

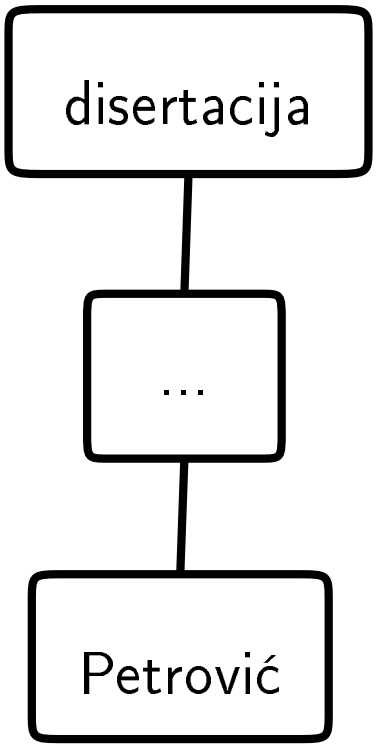
- Dodela težina 0.6 i 0.4 u prethodnom primeru je bila suviše pojednostavljena
  - može biti finije
  - verovatno ga generiše aplikacija, a ne korisnik
- Upiti i dokumenti su normalizovani vektori
- Mera sličnosti je kosinusna mera, kao i u klasičnom slučaju

Ograničenje broja strukturnih termova

- Zavisno od aplikacije može se ograničiti broj strukturnih termova
- Na primer, nikad nećemo tražiti title čvor, nego samo play i book čvorove
- Tada u indeks neće ući strukturni termovi čiji koren je title

Problem (pre)velike dimenzionalnosti prostora

- Koliko je velik ovaj prostor?
- Broj dimenzija može da raste eksponencijalno sa veličinom dokumenta
- Beznadežno je praviti indeks
- I dalje ne može da odgovori na upit kao što je

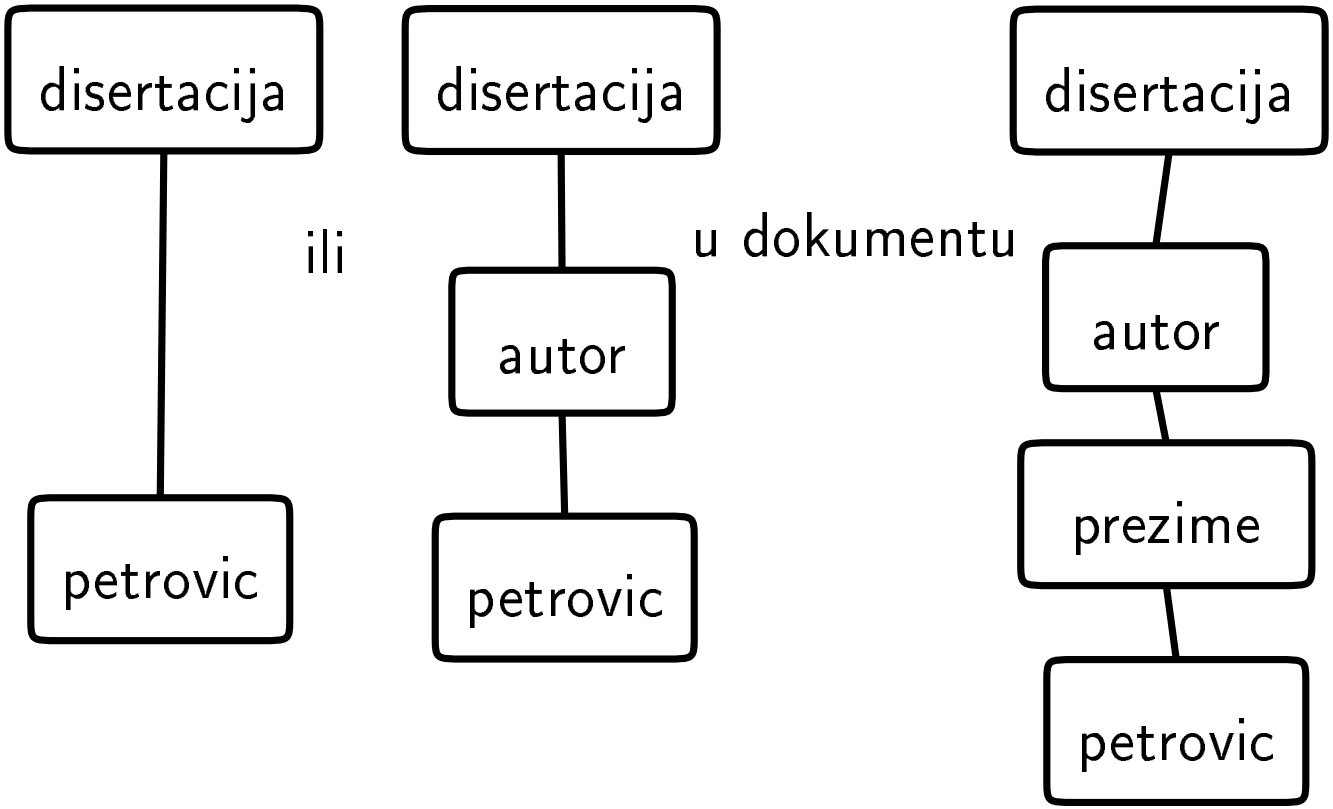


Ograničenje broja podstabala

- Indeksirati sva podstabla verovatno nije isplativo
- Većina podstabala se nikad neće koristiti u upitima
- Bilo bi idealno znati koja podstabla će se javljati u upitima

Pretraga po potomcima

- Ako imamo funkciju za poređenje podstabala koja vraća rezultat iz $[0, 1]$
- Tj. kada su strukturni termovi putanje, meri poklapanje
- Što veće poklapanje, veća ocena



Pretraga po potomcima

- Kako da koristimo ovo u pretraživanju?
- Izdvojimo sve strukturne termove u upitu
- Pretraga po rečniku strukturnih termova
  - rezultat nije binaran (term postoji/ne postoji) nego stepen poklapanja sa termom izražava brojem iz $[0, 1]$
- Dobavimo dokumente sa tim strukturnim termovima, računamo kosinusnu meru sličnosti, itd.

Primer pretraživanja

Upit ST				
0.78	ST1	→ Doc1(0.7)	→ Doc4(0.3)	→ Doc9(0.2)
...				
0.63	ST5	→ Doc3(1.0)	→ Doc6(0.8)	→ Doc9(0.6)

(ST = strukturni upit)

Nakon pretrage u indeksu rangiramo dokumente prema kosinusnoj meri

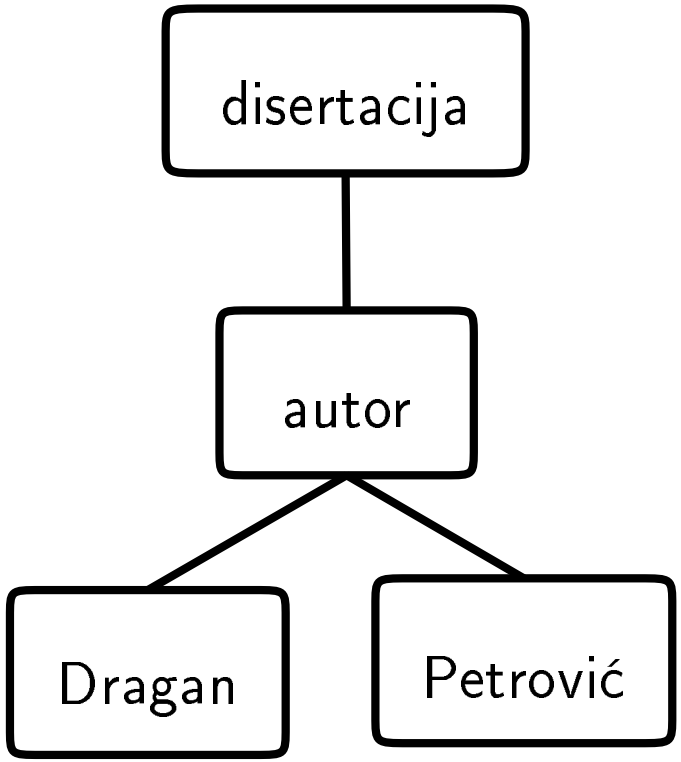
Ograničenja

- Na kakve upite ne možemo odgovoriti u vektorskom prostoru?
- „Nađi slike koji opisuju strukturu EJB komponente i pasuse koji se referenciraju na te slike“
  - treba nam nešto kao join dve tabele
- „Nađi naslove članaka iz trećeg odeljka u avgustovskom broju časopisa IEEE Trans on Software Engineering“
  - zavisi od redosleda čvorova-braće

Može li se računati idf?

- Da, ali nema smisla računati idf na nivou cele kolekcije
- Može imati smisla računati za sav tekst u okviru nekog elementa
- Dobićemo tf-idf težinu svakog terma u okviru datog elementa
- Komplikovano pitanje: kako propagirati težine u roditeljske čvorove

Primer: idf



Recimo da term petrovic ima visok idf u okviru elementa autor. Kako da izračunamo tf-idf za element disertacija? Da koristimo idf za petrovic u elementu autor ili elementu disertacija?