UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

ISKANJE IN EKSTRAKCIJA PODATKOV S SPLETA

SEMINARSKA 3

Dokumentni sistem: indeksiranje in poizvedovanje

Končno projektno poročilo

Avtorja:
BERNIK Matic
TOVORNIK Robert

Profesor:
dr. Marko BAJEC
Asistent:
doc. dr. Slavko ŽITNIK

May 28, 2019

1 Uvod

Cilj naloge je bil razviti program, ki omogoča poizvedovanje po večji količini dokumentov. Uporabljen testni nabor podatkov sestavlja 1416 HTML dokumentov oz. spletnih strani iz štirih različnih domen:

- e-prostor.gov.si
- e-uprava.gov.si
- evem.gov.si
- podatki.gov.si

Za namene pohitritve poizvedovanja nad dokumenti je bila potrebna tudi gradnja preprostega indeksa dokumentov. Osnovo najine implementacije predstavljata dve Python skripti build_index.py (predprocesiranje dokumentov in gradnja indeksa) in retrieve_queries.py (poizvedovanje po dokumentih).

2 Indeksiranje

2.1 Predprocesiranje

Ker se pri poizvedovanju po spletnih straneh tipično sklicujemo le na (tekstovno) vsebino, ki se uporabniku prikaze, je bilo iz HTML dokumenta najprej potrebno izluščiti vsebinsko besedilo, brez samih strukturnih elementov HTML kot so npr. značke. V ta namen sva uporabila knjižnico BeautifulSoup in njeno get_text() funkcionalnost. Pred tem sva iz dokumentov tudi odstranila vsebino znotraj značk <script> in <style>, ki samo po sebi za uporabnika ravno tako ne nosi vsebinske informacije in bi tako zgolj po nepotrebnem prispevala k času procesiranja dokumentov in končni velikosti indeksa.

Zaradi neprilagojenosti slovenskega lematizatorja LemmaGen za rabo iz jezika Python, se za vključitev postopka lematizacije tokrat nisva odločila. Z lematizacijo bi sicer lahko se dodatno zmanjšala velikost besedišča in posledično indeksa, ter omogočila robustnejše poizvedovanje po dokumentih z večjim številom zadetkov.

Izluščeno besedilo sva nato tokenizirala z uporabo knjižnice nltk in iz nadaljnje obravnave odstranila stopworde na podlagi že predpripravljenega seznama. Odstranila

sva tudi tokene dolžine 1 in take, ki bi bili sestavljeni iz samih ločil oz. posebnih znakov (primer: "//", ki je sicer del spletnega naslova).

Testirala sva tudi nadomestitev vseh števil enotno oznako "$NUMBER" (regularni izrazi) z namenom dodatnega zmanjšanja besedišča in splošnejsega poizvedovanja (ni potrebno poznati natančnega števila, ki ga iščemo). Velik stranski učinek taksne implementacije je sicer, da v poizvedbi ne moremo uporabiti konkretnih primerkov števil. Izboljšava implementacije bi omogočala rabo obeh.

2.2 Indeks

Indeks je implementiran kot SQLite podatkovna baza z dvema tabelama. Tabela IndexWord hrani množico vseh unikatnih tokenov, ki se nahajajo v indeksiranih dokumentih. Tabela Posting za vsak par dokumenta in besede oz. tokena, ki se v njem nahaja, hrani mesta pojavitve te besede znotraj dotičnega dokumenta.

V najinem primeru čas gradnje indeksa (uporaba 1 niti procesorja i7-7700HQ) znaša približno 47 sekund, indeks, zgrajen nad testnim naborom dokumentov, pa obsega:

- 38648 besed znotraj tabele IndexWord,
- 373338 vnosov v tabeli Posting.

Tokeni oz. besede, ki se znotraj testnega nabora dokumentov pojavljajo najpogosteje so:

- "$number" (34400 pojavitev)
- "podatkov" (10502 pojavitev)
- "slovenije" (9856 pojavitev)
- "republike" (8583 pojavitev)
- "podatki" (5562 pojavitev)
- "dejavnosti" (5560 pojavitev)
- "navigation" (4474 pojavitev)
- "krepko" (4277 pojavitev)

- "navadno" (4242 pojavitev)
- "povezava" (4140 pojavitev)

Dokumenti z najširšim besediščem so:

- "evem.gov.si.371.html" (12474 razlicnih tokenov)
- "podatki.gov.si.340.html" (6606 razlicnih tokenov)
- "e-prostor.gov.si.57.html" (1784 razlicnih tokenov)
- "evem.gov.si.398.html" (1636 razlicnih tokenov)
- "evem.gov.si.651.html" (1282 razlicnih tokenov)

2.3 Dodatna pohitritev

Kot že omenjeno v koraku predprocesrianja sva največjo pohitritev pri priklicu izvorne vsebine dokumentov in pri predprocesiranju dosegla s "pametno" obdelavo strukture HTML dokumentov. Ker se v značkah <script> in <style> skriva zgolj izvedbena in prikazovalna logika spletnega dokumenta in ne vsebinska, poleg tega pa je vsebina teh značk dostikrat polna kompleksnih nizov, sva omenjene značke eliminirala. S tem korakom sva dosegla znatno pohitritev. Za primer datoteke "evem.gov.si.55.html", ki izstopa že po sami velikosti (večina datotek je velikostnega reda nekaj 10 kB, omenjena pa je velika 3MB) sva dosegla časovno pohitritev obdelave dokumenta **iz predhodnih 142 sekund na zgolj na 0.049 sekunde.**

3 Poizvedovanje

Poizvedovanja nizov sva se lotila na dva načina: naivnega in indeksiranega. Pri tem sva se omejila na dolžino poizvedovalnih nizov med 1 in 5 besed. Primarno uporabljen način poizvedovanja je indeksiran, saj je znatno hitrejši. Sekundarni ali naivni način sva implementirali zgolj za namen časovne primerjave učinkovitosti primarnega poizvedovanja. Tekom preizkušanja sva potrdila superiornost indeksiranega načina poizvedovanja.
Pri obeh načinih poizvedovanja sva uporabila enak postopkovni cevovod:

- Predprocesiranje poizvedbenega niza

- Iskanje pojavitvenih indeksov poizvedbenega niza
- Urejanje in priklic pojavitev v originalnem dokumentu
- Izpis priklica poizvedbe v tekstovno datoteko

Glavna razlika med pristopoma se nahaja v drugem koraku - način iskanja indeksov poizvedbenega niza.

Predprocesiranje poizvedbenega niza sva opravila na enak način, kot predprocesiranje dokumentov, kar je opisano v prejšnjem poglavju. Nato sva glede na poizvedbeni niz poiskala indekse pojavitev v korpusu. Sledilo je razvrščanje pomembnosti dokumentov glede na število dokumentov oziroma frekvenco pojavitev (zadetkov) poizvedbenega niza v posameznih dokumentih. Razvrstila sva jih po pomembnosti padajoče (najpomembnejši dokument je prvi v seznamu). Nato je sledil posamezen priklic izsekov izvornih dokumentov, kjer se nahajajo elementi iskalnega niza. V tem delu sva uporabila nekoliko drugačen način predprocesiranja, ki pa je v večinskem delu enak tistemu iz prvega dela. Glavna razlika je bila v ohranitvi izvornega besedila (torej preskočitev postopka zamenjav, eliminacij itd..) s poudarkom zgolj na postopku razbitja besedila na pojavnice, pri čemer se ohranjajo pravilni indeksi posameznih besed. Na koncu se za vsak posamezen dokument, ki vsebuje pojavitve poizvedbenih nizov v tekstovno datoteko zapiše ena vrstica v obliki: POMEMBNOST_DOKUMENTA(frekvenca) - POJAVITVENI_DOKUMENT - IZSEKI_IZVORNEGA_BESEDILA. Tekstovne datoteke imajo dve možni predponi : **DB** ali **RAW**. **DB** predstavlja izhode indeksiranega poizvedovanja, **RAW** pa izhode naivnega poizvedovanja. Vsebinsko so enaki, razlikujejo se zgolj v vrstnem redu pojavitev izvornega besedila, zaradi razlike v načinu indeksiranja.

3.1 Iskanje pojavitvenih indeksov - podatkovna baza

V postopku poizvedovanja s pomočjo indeksa (podatkovne baze), prikličeva vse ujemanjoče dokumente s pomočjo SQL poizvedbe:

```
"""SELECT documentName,word,frequency,indexes FROM Posting WHERE word = ?;"""
```

pri čemer iščeva ujemanje besede word z vrednostjo posameznih poizvedbenih besed znotraj poizvedenega niza. S tem dobiva vse dokumente v katerih se posamezna beseda pojavi (vsaj enkrat) in pripadajoče indekse besed. Za pravilno

pripadajoče indekse poskrbiva že v postopku predprocesiranja, kjer posameznih pojavnicam pripiševa indeks pojavitve. Ker nas zanima unija poizvedbenega niza nato za vsak priklican dokument seštejeva pomembnost glede na poizvedbeni niz (seštejeva posamezne frekvence pojavitvenih besed) s čimer določiva končno pomembnost posameznih dokumentov glede na niz. Rezultate poizvedovanja pridobiva v končni obliki: pomembnost/frekvenca - pojavitveni_dokument - unija_pojavitvenih_indeksov.

3.2 Iskanje pojavitvenih indeksov - naivno

Pri naivnem načinu poizvedovanja pa je postopek nekoliko drugačen - preprostejši. Sprehodiva se skozi seznam vseh dokumentov korpusa, vsako stran predprocesirava na zgoraj omenjen način (zgolj razbitje - z ohranitvijo izvornega besedila), nato pa za vsako pojavnico preverjamo ali je hkrati tudi del poizvedbenega niza. V primeru, ko pride do ujemanja se zabeleži ujemanje za specifičen dokument v katerem je bila pojavnica zabeležena, ustrezno se poveča pomembnost dokumenta in zabeleži indeks pojavnice v izvornem besedilu. Ob zaključu preiskovanju vrneva enako urejen seznam kot pri postopku indeksiranja: pomembnost/frekvenca - pojavitveni_dokument - unija_pojavitvenih_indeksov.

3.3 Poizvedbeni nizi

Poizvedovanje sva izvedla na sedmih nizih. Prvih šest je skladnih z omejitvijo, pri čemer so bili prvi trije podani, naslednji trije pa so bili poljubno izbrani. Sedmi poizvedbeni niz pa je služil namenu preverjanja časovne učinkovitosti.

- "predelovalne dejavnosti"
- "trgovina"
- "social services"
- "finance borza trg"
- "podatki analiza znanost"
- "zakon"
- "kataster zemljišče davek meja ocena vlada stavba model register javni"

3.4 Časovna analiza pristopov

Da bi ustrezno preverili časovno učinkovitost indeksiranega pristopa sva se odločila za primerjalno analizo s pristopom naivnega poizvedovanja. Pri tem sva potrdila, da je indeksiran način veliko učinkovitejši, ocenjujeva da nekje za faktor 1000-10000 krat hitrejši. Razlog je očiten: medtem, ko pri indeksiranem načinu posamezen dokument v korpusu preberemo zgolj enkrat, ga moramo pri indeksiranem načinu prebrati ob vsakokratni poizvedbi. Pravzaprav moramo ob vsakokratni poizvedbi prebrati celoten korpus dokumentov in jih primerno pred procesirati, kar je glavni razlog za časovno neučinkovitost pristopa. Takšen pristop je sicer časovno konstanten, vendar zato tudi počasen. Pri indeksiranem pristopu sva zaznala manjši trend naraščanja časovne zahtevnosti s povečevanjem dolžine poizvedovalnega niza, vendar zahtevnost ostaja v rangu nekaj milisekund.

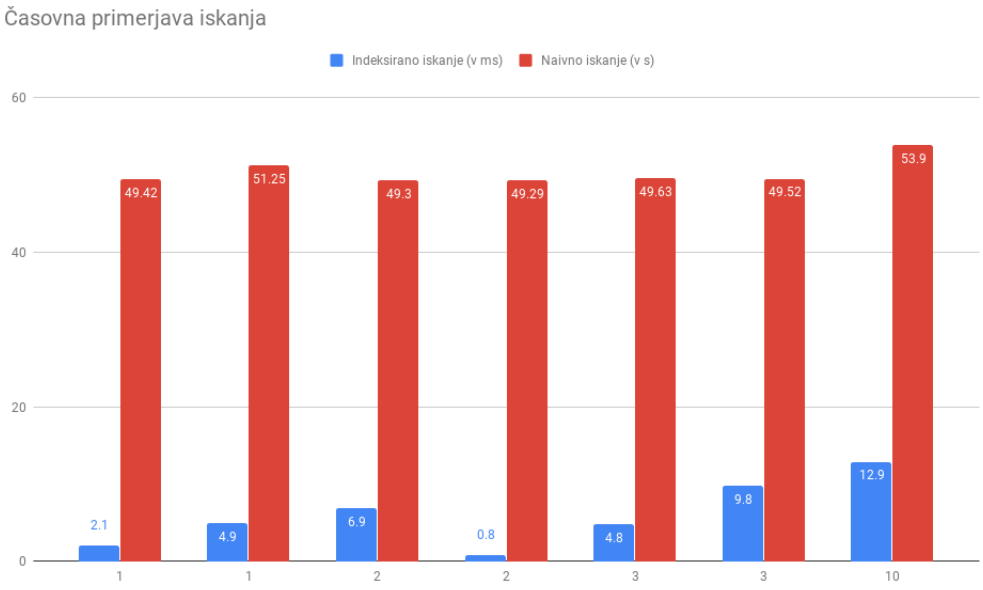

Figure 1: Graf prikazuje časovno razliko med pristopoma indeksiranega in naivnega poizvedovanja. Kot je razvidno z grafa je naivno poizvedovanje veliko počasnejše faktor 1000-10000. Pozor! Čas indeksiranega iskanja je podan v milisekundah, čas naivnega iskanja pa v sekundah.

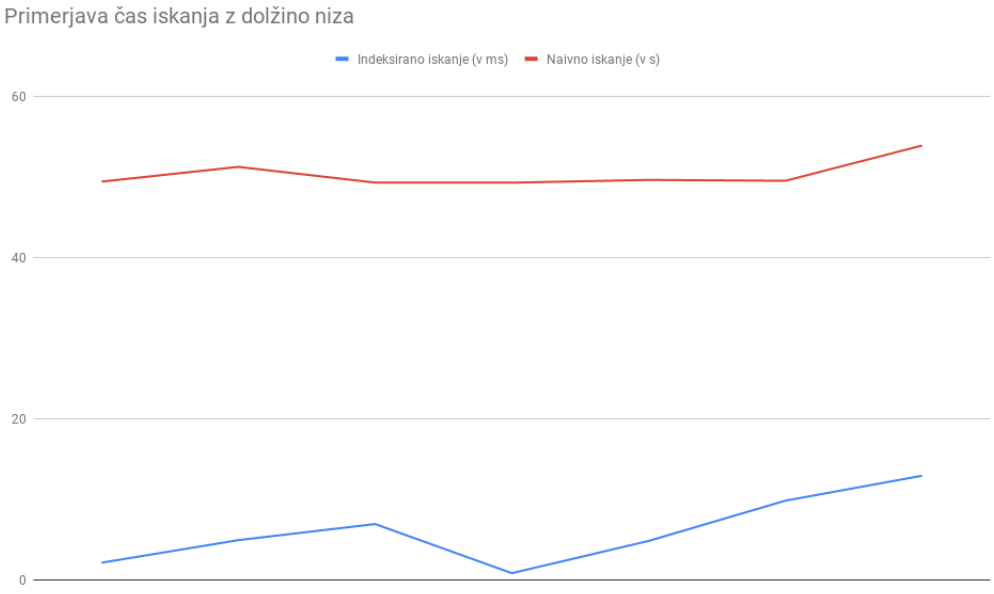

Figure 2: Graf prikazuje niz naraščanja poizvedbe v odvisnosti z dolžino niza poizvedbe. Kot je razvidno z grafa lahko zaznamo, da je poizvedovanje pri naivnem iskanju dokaj enakomerno, medtem, ko je pri indeksiranem poizvedovanju mogoče zaznati trend naraščanja.

4 Rezultati

Zaradi visoke pomembnosti/frekvence pojavitve določeni poizvedbenih nizov, sva rezultate izpisovala v tekstovne datoteke, po eno vrstico za vsak pomemben dokument. Izhodi so na voljo preko povezave na datoteke repozitorija. Izhodne datoteke imajo dve predponi: **DB** - indeksirano poizvedovanje in **RAW** - naivno poizvedovanje. Vsebinsko so datoteke preverjeno enake, vrstni red in pomembnost/frekvence so enaki za pare datotek, razlikujejo se zgolj v vrstnem redu prikaza izsekov izvornega besedila, zaradi razlike v načinu indeksiranja.

5 Povezave

Github repozitorij projekta se nahaja na naslovu: https://github.com/MaticBernik/webpage_indexing_retrieval