# Ekstrakcija podatkov iz spleta

3. domača naloga pri predmetu IEPS

Žiga Černigoj 63130028

Marko Lavrinec 63130134

Maj 2019

## 1 Uvod

Cilj naloge je razviti program, ki za podan iskani niz poišče ustrezne rezultate v bazi in datotekah. Za implementacijo bo uporabljen programski jezik Python.

## 2 Pred-procesiranje in indeksiranje podatkov

Skripta za pred-procesiranje (`./indexer/data_processor.py`) najprej preveri, če že obstaja SQLite datoteka (`./inverted-index.db`) in jo v nasprotnem primeru naredi s pravilno zasnovo tabel.

Nato se skripta sprehodi čez vse datoteke v imeniku (mapa `./indexer/PA3-data`). Iz vsake datoteke z uporabo knjižnice BeautifulSoup prebere vsebino znotraj značke `body` in pri tem odstrani značke `script`, `noscript` in `style`, katerih vsebina bi se drugače prav tako uporabila pri nadaljnjih korakih. Skripta nato izvede tokenizacijo - ustvari tabelo besed in ločil, ki so razporejeni v takem vrstnem redu, kot so v besedilu. V angleščini je za te elemente uporabljen izraz `tokens`, v tem poročilu pa bomo uporabili besedo `žetoni`. Za ustvarjanje `postingov` skripta uporabi le žetone, ki niso *stopword-i* in ločila, pred uporabo pa žetone pretvori v enak format - male črke.

Za tem skripta žetone, ki zadostujejo pogojem (t.j. so besede oziroma niso *stopword-i* ali ločila), vstavi v slovar in zabeleži vse njihove pojavitve ter število pojavitev. Za mesto pojavitve se zabeleži zaporedni indeks v izvirni tabeli žetonov.

Prvotni način vstavljanja podatkov v bazo je bil tak, da smo vsako besedo iz slovarja najprej vstavili v tabelo *IndexWord*, če ta v tabeli še ne obstaja, nato pa še v tabelo *Posting*, kamor smo shranili še ime dokumenta kjer se nahaja, frekvenco ter indekse pojavitev, ločene z vejicami. Pri tem smo vsak vnos posebej tudi potrdili (*commit-ali*).

Nato smo se odločili za uporabo transakcij z več vnosi naenkrat. Tako najprej v bazo v tabelo *IndexWord* vstavimo vse besede iz vseh dokumentov in nato naredimo potrditev (*commit*). Nato za vsak dokument vstavimo vse postinge v tabelo *Posting* in šele nato naredimo potrditev (*commit*). Tak način se je

izkazal za veliko izboljšavo, saj je namreč pohitril shranjevanje podatkov v bazo za okrog 8x.

# 3 Iskanje po bazi in Inverted index

Podani iskani niz najprej tokeniziramo in nato v bazi poiščemo vse pojavitve teh besed z naslednjim SQL stavkom:

```
SELECT word, documentName, frequency, indexes
FROM Posting
WHERE word IN ({seq})
```

Rezultate poizvedbe nato združimo po dokumentih, izračunamo skupno frekvenco pojavitev in generiramo izseke, ki vsebujejo besede iz iskanega niza ter največ 3 besede pred in 3 besede po besedah iz iskanega niza.

Dobljeni rezultati se nato v zaporednem vrstnem redu izpišejo uporabniku skladno z navodili naloge, ki veleva, naj sortiramo zadetke skladno od največje do najmanjše skupne frekvence. Rezultate bi lahko sortirali tudi drugače, saj v programu glede na indekse zaznavamo sosednost, kar pomeni, da bi lahko pri večbesednem iskanju najprej prikazali zadetke, ki imajo iskane besede navedene v enakem vrstnem redu kot so v rezultatu in šele nato sortirali glede na frekvenco pojavitev.

Koda iskanja z invertnim indeksom je v datoteki `./indexer/index_search.py`.

# 4 Zaporedno branje datotek

Tudi pri tem načinu podani iskani niz najprej tokeniziramo. Skripta se nato sprehodi čez vse datoteke, jih tokenizira. Za olajšanje razvoja takega iskanja smo tudi tu generirali postinge, vendar jih nismo nikamor shranili. Postinge smo nato uporabili, da smo preverili, ali se besede iz iskanega niza pojavijo v datoteki ter nato dobili frekvenco in indekse pojavitev besed iz iskanega niza. Potem smo tudi tu enostavno dobili največ 3 besede pred in 3 besede po besedah iz iskanega niza.

Delovanje takega iskanja je počasno, saj se ob vsakem iskanju na novo procesira vsebina vseh datotek in po njih išče ustrezne besede.

Koda iskanja z zaporednim branjem datotek je v datoteki `./indexer/naive_search.py`.

# 5 Rezultati in analiza

## 5.1 Rezultati

Spodaj so našteti iskani nizi in datoteke, kjer se nahajajo celotni izpisi iskanja. Rezultatov iskanj zaradi njihove dolžine nismo umestili neposredno v poročilo. Pod seznamom iskanih nizov je zapisanih nekaj ugotovitev.

- predelovalne dejavnosti

  - invertni indeks: `./results/predelovalne_dejavnosti.txt`
  - zaporedno branje datotek: `./results/predelovalne_dejavnosti_naivesearch.txt`
- trgovina
  - invertni indeks: `./results/trgovina.txt`
  - zaporedno branje datotek: `./results/trgovina_naivesearch.txt`
- social services
  - invertni indeks: `./results/social_services.txt`
  - zaporedno branje datotek: `./results/social_services_naivesearch.txt`
- okoljevarstveno dovoljenje
  - invertni indeks: `./results/okoljevarstveno_dovoljenje.txt`
  - zaporedno branje datotek: `./results/okoljevarstveno_dovoljenje_naivesearch.txt`
- stanovanjski objekt
  - invertni indeks: `./results/stanovanjski_objekt.txt`
  - zaporedno branje datotek: `./results/stanovanjski_objekt_naivesearch.txt`
- registracija samostojnega podjetnika
  - invertni indeks: `./results/registracija_samostojnega_podjetnika.txt`
  - zaporedno branje datotek: `./results/registracija_samostojnega_podjetnika_naivesearch.txt`

## 5.2 Ugotovitve

Dolžino snippetov smo zaradi velikega števila pojavitev nekaterih iskanih nizov v nekaterih dokumentih omejili na 200 znakov. Značilen primer je iskani niz "predelovalne dejavnosti", saj se besede iskanega niza v dokumentu `evem.gov.si.371.html` pojavijo kar 1291-krat. Besede tega iskanega niza se pojavijo v največ dokumentih. Iskanje je vrnilo pojavitve v kar 754 dokumentih.

Najmanj rezultatov smo dobili za iskani niz "social services", saj je večina strani v slovenščini. Z iskanjem smo našli vsega skupaj 10 pojavitev besed iz iskanega niza v 4 dokumentih.

Za iskani niz po lastni izbiri smo hoteli izbrati besedo "reforma", za katero pa nismo dobili rezultatov.

### 5.3 Analiza baze

- Število zapisov v tabeli `IndexWord`: 47862
- Število zapisov v tabeli `Posting`: 392218
- Beseda z največjo frekvenco pojavitve: "proizvodnja" (v datoteki `evem.gov.si.371.html` se pojavi 2266-krat)
- Dokument z največjim številom zapisov v tabeli `Posting` (t.j. posredno - z največ raznolikimi besedami): `evem.gov.si.371.html`, pojavi se v 13301 zapisih v tabeli Posting
- Nasprotje tega: `evem.gov.si.55.html`, pojavi se v 31 zapisih v tabeli Posting
- Besede v največ dokumentih: "pogoji" (v 1398 zapisih - dokumentih), "uporabe" (v 1399 zapisih - dokumentih), "domov" (v 1384 zapisih - dokumentih)

## 6 Zagon algoritma

Zagon programa za indeksiranje se lahko izvede preko ukazne vrstice z vpisom ukaza `python data_processor.py`, za iskanje rezultatov z uporabo inverznega indeksa je potrebno pognati `python index_search.py`, za pridobitev rezultatov z zaporednim branjem datotek pa `python naive_search.py`.

## 7 Zaključek

V tej seminarski nalogi smo razvili delujoči program za indeksiranje vsebin spletnih strani.

## References