Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

Matevž Lapajne, Simon Klavžar in Tine Črnugelj

Tretja seminarska naloga

Domača naloga pri predmetu Iskanje in ekstrakcija podatkov s spleta

MENTOR: prof. dr. Marko Bajec, doc. dr. Slavko Žitnik

Predloženo poročilo predstavlja rezultate predobdelave in shranjevanja podatkov v indeks. Cilj naloge je pridobiti besedilni kontekst iz predloženih datotek HTML, izvesti predobdelavo teh podatkov in jih shraniti v indeksno strukturo. Indeks omogoča učinkovito iskanje in poizvedovanje po besedilnih vsebinah, ki so bile pridobljene iz HTML datotek.

Indeksiranje podatkov

Sprva se poveže v že vnaprej izdelano bazo **inverted-index.db** v datoteki **db_init.py** in se nato v skripti **data-process-index.py** s pomočjo funkcije **create_index** sprehodi čez vse datoteke v direktoriju PA3-data. Iz vsake HTML datoteke se z uporabo knjižnice BeautifulSoup prebere vsebina. Skripta nato izvede tokenizacijo na dobljenem besedilu, tako da ustvari seznam besed in ločil. Sledi izločevanje nabora pogosto uporabljenih besed (ang. stopwords), ki se pojavijo v besedilu s pomočjo funkcije **filter_tokens**. Pogosto uporabljene besede (ang. stopwords) hranimo datoteki **stopwords.py**. Druge besede se zapišejo v bazo s pomočjo funkcije **db_insert_word**. Funkcija **db_insert_posting** nam omogoča zapis posamezne besede vključno z lokacijo besede v besedilu (ang. index) in število pojavitev.

Iskanje podatkov brez obrnjenega indeksa

Na začetku preberemo besede poizvedbe iz argumentov ukazne vrstice in jih pretvorimo v male črke ter shranimo v seznam **query_words**. Zatem se zažene časovnik in začnemo s pregledovanjem datotek v imeniku PA3-data. Za vsako datoteko s končnico »html« prebere vsebino datoteke, s knjižnico BeautifulSoup iz nje izloči besedilo in ga označi (ang. tokenize). Sledi iteracija po vsaki označeni besedi v **page_tokenized** in preveri, ali obstaja v argumentih ukazne vrstice **query_words**. Če najde ujemanje, poveča število **frequency,** zabeleži indeks ujemanja **index** in ustvari odlomek **snippet** tako, da okoliške besede (4 besede pred in za ujemanjem) poveže v niz. Če je število frekvenc večje od nič, ustvari objekt Rezultat s frekvenco, potjo do dokumenta in odlomkom ter ga doda v seznam rezultatov. Po obdelavi vseh datotek se seznam rezultatov razvrsti v padajočem vrstnem redu na podlagi atributa **frequency**, pogostosti pojavitvi besede. Nato koncu še prenehamo z merjenjem časa izvajanja in ga izpišemo.

Iskanje podatkov z obrnjenim indeksom

Najprej se vzpostavi povezavo s podatkovno bazo SQLite z imenom **inverted-index.db**. Sledi shramba besed poizvedbe iz argumentov ukazne vrstice v seznam **query_words**. Za poizvedbo iz baze je potrebno ustvariti argument **placeholders,** glede na število besed poizvedbe. Klic iz baze se zapiše v spremenljivko **cursor** s pomočjo funkcije execute. V kateri prvi parameter predstavlja poizvedba SQL, ki pridobi ime dokumenta, vsoto frekvenc in združene indekse iz preglednice. Drugi parameter predstavlja besede poizvedbe **query_words**. Vrnjene vrstice se obdelajo v zanki. Indeksi se sestavijo v seznam, pridobi se ime dokumenta in prebere se ustrezna datoteka, iz katere se izloči besedilo. Ustvari se odlomek z uporabo indeksov in okoliških besed. Na koncu se ustvari objekt **Rezultat** in doda v seznam **results**.

Analiza podatkovne baze in rezultati

Število zapisov v tabeli IndexWord: 48233

Število zapisov v tabeli Posting: 381437

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)

Rezultatov zaradi dolžine besedila nismo vključili v poročilo, so pa dostopni v Github repozitoriju (./pa3/results). Poleg vnaprej določenih poizvedb smo si izbrali še naslednje poizvedbe:

- zakon
- zapiranje gospodarke družbe
- socialna varnost

Ugotavljamo, da je iskanje z obrnjenim indeksom več kot 10-krat hitrejše od naivnega iskanja.

Ugotovitve

Iskani niz »predelovalne dejavnosti«, se pojavi največrat – 1288 krat v dokumentu evem.gov.si.371.html.

Iskani niz »social services« se pojavi najmanjkrat – 12 krat skupno v štirih dokumentih.

Zaključek

V tem poročilu smo razložili implementacijo obrnjenega indeksa. Najprej smo razložili postopek indeksiranja podatkov, nato pa še razložili implementaciji iskanja po korpusu z navadnim oz. naivnim iskanjem in z uporabo obrnjenega indeks ter ugotovili, da je slednji postopek veliko hitrejši.