Spletni pajek

1. domača naloga pri predmetu IEPS

Žiga Černigoj 63130028

Marko Lavrinec 63130134

Marec 2019

1 Uvod

Cilj naloge je razviti spletnega pajka, ki bo obiskal in pridobival podatke le s spletnih strani na domenah `*.gov.si`. Začetne domene so `evem.gov.si`, `e-uprava.gov.si`, `podatki.gov.si` in `e-prostor.gov.si`. Pajek mora biti implementiran tako, da se lahko izvaja večnitno ali večprocesno. Upoštevati mora pravila v datotekah `robots.txt`, če le-te obstajajo. Če obstaja dokument `sitemap`, mora v njem navedene povezave shraniti v `frontier`. Pajek mora znati zaznati duplicirano vsebino ter URL naslove. Obiskati mora URL naslove, ki so navedeni v značkah `a` in značkah z atributom `onclick`. Shraniti si mora slike v značkah `img` in dokumente formatov `pdf`, `doc`, `docx`, `ppt`, `pptx`.

2 Implementacija

2.1 Podatkovna baza

Podatkovna baza je postavljena na postgresu po danem vzorcu iz naloge. Narejenih je le nekaj sprememb na strukturi, kjer se je dodala vmesna tabela in nekaj stolpcev.

Zaradi 3. normalne oblike je bila tabela `image` predelana v tabelo `page_image` in `image`. Tako image ne vsebuje več atributa `page_id`, vsebuje pa unikaten ključ `url`. Tabela `page_image` je samo vmesna tabela za relacijo mnogo proti mnogo med `page` in `image`.

Za lažje delo z več nitmi, procesi in napravami sta bila dodana tudi stolpca `reservation_id (int)` in `reserved (datetime)` v tabeli `site` in `page`. Ob želji obiska nove strani ali domene tako z ukazom rezerviramo še neobiskano stran oziroma domeno, ki jo nato iz baze tudi izberemo. Torej se najprej izvede `UPDATE` na bazo, ki postavi rezervacijo, nato pa še `SELECT`, ki iz baze pridobi pravkar rezervirano stran. Ko pa z obiskom strani končamo rezervacijo počistimo. V primeru padca našega pajka, se rezervacije sproščajo po eni uri od postavitve.

2.2 Jezik in knjižnice

Spletni pajek je spisan v programskem jeziku Python. Za svoje delovanje uporablja več knjižnic, ki olajšajo njegovo izvajanje. Te so Selenium (upravljanje brskalnika), sqlalchemy (dostop do baze), BeautifulSoup4 (razčlenjevalnik vsebine), requests (za dostop do dokumentov in vsebin), urllib (za delo z robots.txt), multiprocessing (za večprocesno izvajanje).

2.3 Osnove algoritma

Algoritem ob zagonu iz baze vzame prvo še ne obiskano stran tipa `FRONTIER`. Obstajata dve možnosti:

- V bazi ni nobenega nerezerviranega `fronteirja`.
- Dobi se `frontier`, ki še ni rezerviran.

V primeru prve točke se iz baze vzame prvo stran, ki še ni bila obdelana in ni rezervirana. Neobdelano stran najdemo tako, da ima v poljih `sitemap` in `robots` nastavljeni vrednosti `null`. Tudi tu obstajata dve možnosti.

- V bazi ni nobenega nerezervirane in neobiskane strani -¿ pajek je zaključil svoje delo
- Dobi se domeno, ki se ji obišče `robots.txt` iz nje pa nato še morebiten `sitmap`, s tem pa smiselno napolni tabelo page z novimi stranmi. Ob koncu tega koraka se rekurzivno pokliče začetek tega algoritma.

Če smo prišli do sem, potem zagotovo imamo stran, ki jo še nismo obiskali. S knjižnico Selenium naredimo zahtevek na stran in iz dobljene strani nato pridobimo slike in povezave. Te obdelamo in če ustrezajo zahtevam `robots.txt` datoteke jih shranimo v bazo.

Na koncu v bazi obiskani strani spremenimo tip strani in jo napolnimo s html vsebino in statusom prejetega odgovora. Algoritem nato rekurzivno ponavljamo, dokler nam ne zmanjka neobiskanih strani in domen.

2.4 Večprocesno izvajanje

Večprocesno izvajanje je realizirano z uporabo knjižnice multiprocessing. Ob zagonu algoritma se zažene toliko procesov, kot je podano ob izvedbi ukaza za zagon algoritma.

Vsak proces začne z neko stranjo iz baze, ki še ni bila rezervirana s strani ostalih procesov, ter jo rezervira ter začne obdelovati. Stran je tu mišljena kot zapis v tabeli page v bazi, ki je tipa `FRONTIER`. Po koncu obdelovanja trenutne strani vzame naslednjo še ne rezervirano stran iz baze in postopek se ponovi.

Zaradi uporabe pristopa rezervacije strani lahko algoritem poganjamo tudi na več računalnikih.

2.5 Pridobivanje in razširjanje povezav

Pridobivanje povezav znotraj vsebine spletnih strani je izvedeno s poizvedovanjem, ki ga omogoča knjižnica BeautifulSoup.

Že pred razširjanjem preverimo, ali ima URL naslov dovoljeno končnico (ignoriramo datoteke, ki niso spletne strani ali datoteke zahtevanih formatov).

Z razširjanjem poskrbimo, da se relativne povezave razširijo v absolutne URL naslove. Pri tem preverimo sledeče:

- ali se naslov začne s `http://`, `https://` ali `ftp://`: naslov je že absoluten
- ali se naslov začne s `/` ali `../`: naslov je relativen, vendar je potrebno bazni naslov prilagoditi, da kaže v pravilno mapo (ta je lahko določen z značko `<base>` ali pa je naslov trenutne strani brez imena dokumenta)
- ali se naslov začne s `mailto:`, `file:` ali `javascript:`: v tem primeru URL naslov ignoriramo

Če naslov ne ustreza nobenemu od zgornjih pogojev, je relativen in brez posebnega začetka. Potrebno ga je samo dodati baznemu URL naslovu (ta je lahko določen z značko `<base>` ali pa je naslov trenutne strani brez imena dokumenta).

2.6 Pridobivanje vsebine strani, dokumentov in slik

Pred pošiljanjem zahtevka za spletno stran ali dokument, preverimo, ali trenutni URL naslov kaže na dokument v enem izmed zahtevanih formatov, ali pa kaže spletno stran. Glede na to se potem izvede postopek pridobivanja vsebine strani (2.6.1) oziroma postopek pridobivanja dokumentov (2.6.2).

2.6.1 Pridobivanje vsebine strani

Pridobivanje vsebine spletne strani je implementirano z uporabo ogrodja Selenium, ki je namenjeno upravljanju brskalnikov brez grafičnega vmesnika. Ogrodje Selenium upravlja z brskalnikom PhantomJS.

Za razčlenjevanje vsebine in poizvedovanje po njej je uporabljena knjižnica BeautifulSoup.

2.6.2 Pridobivanje dokumentov

Za pridobivanje dokumentov je uporabljena knjižnica requests. Dokumente v bazo shranimo v binarni obliki.

2.6.3 Pridobivanje slik

Podobno kot za pridobivanje dokumentov je tudi za pridobivanje slik uporabljena knjižnica requests. Zahtevki za slike se izvedejo po razčlenitvi vsebine spletne strani. Slike so v bazo shranjene v binarni obliki.

2.7 Zaznavanje duplikatov

Za osnovno zaznavanje duplikatov nam skrbi že baza sama. Ta namreč ne dovoli vnosa podvojenih url linkov. Kar posledično pomeni, da istih povezav ne moremo večkrat shraniti.

Seveda pa to ni dovolj. V izogib pretiranemu obiskovanju strani, smo se odkrivanja duplikatov lotili s primernim čiščenjem url povezav (iz katerih smo odstranili nepotrebne dele), prav tako pa smo tudi GET parametre razvrstili po abecedi. Slednje v praksi pomeni, da če imamo dve podobni povezavi, da ti dve peljeta na isto stran:

- `stran.gov.si/novica?kategorija=5&stran=1`
- `stran.gov.si/novica?stran=1&kategorija=5`

V tem primeru naš algoritem parametre razvrsti po abecedi in vedno dobimo povezavo iz zgornje prve opcije, podvojenost pa bo potem zaznala baza sama, saj bo v obeh primerih shranjen isti url.

Dodaen je tudi sistem za zaznavanje duplikatov v povezavi s html vsebino. Če je tako zaznana vsebina na trenutni strani enaka že shranjeni vsebini na neki drugi strani z drugim urljem, se nam tu vsebina ne bo shranila, se bo pa napolnil stolpec `duplicated_with` z idjem zapisa, kjer je ta vsebina že shranjena, tip pa bo na koncu `DUPLICATE`.

3 Zagon algoritma

Zagon spletnega pajka se lahko izvede preko ukazne vrstice z vpisom ukaza `python crawler.py <n>`, kjer je n opcijski parameter, ki določa število procesov, ki jih pajek zažene. Če parameter ni podan, se zažene zgolj en proces.

4 Težave in posebnosti

Imeli smo željo, da nam ne bi bilo treba podvajati zahtevkov, da bi dobili statusno kodo odgovora in nato še vsebino strani. Zato smo iskali tak brskalnik, ki bi omogočal pridobivanje statusa odgovora (status code) skupaj z vsebino v istem zahtevku. Preizkusili smo več brskalnikov brez grafičnega vmesnika, vendar smo le s PhantomJS uspeli narediti željeno.

Težave smo imeli tudi s procesiranjem `robots.txt` datotek, na koncu pa smo uporabili kar RobotFileParser iz popularne Python knjižnice urllib.

5 Analiza

5.1 Na domenah `evem.gov.si` in `e-prostor.gov.si`

Na domenah `evem.gov.si` in `e-prostor.gov.si` se nahaja približno 24265 različnih strani (zapisi v tabeli `pages`), ki so med seboj povezane s 921386

page_type	število
BINARY	3439
HTML	6270
DUPLICATE	14586

Table 1: Število zapisov v tabeli page glede na `page_type` za `evem.gov.si` in `e-prostor.gov.si`

data_type	število
pdf	306
ppt	1
doc	73
docx	37
pptx	0

Table 2: Število zapisov v tabeli page glede na `data_type` (format) za `evem.gov.si` in `e-prostor.gov.si`

povezavami (zapisi v tabeli `link`). Na jih se nahajajo 203 slike (zapisi v tabeli `image`).

Na sliki 1 je prikazan majhen del spletnih strani, ki se nahajajo na domeni `evem.gov.si` in povezav med njimi. Podrobnejša vizualizacija 314 strani in 6480 povezav med njimi je na voljo na tej povezavi oziroma med literaturo [1].

V tabeli 1 je število zapisov v tabeli page glede na `page_type`, v tabeli 2 pa število binarnih datotek glede na `data_type` (format datotek). Tabela 3 ponuja podatke o številu zapisov v tabeli page glede na pripadnost neki domeni (zapisu v tabeli `site`).

domena	število
`e-prostor.gov.si`	17747
`evem.gov.si`	6548

Table 3: Število zapisov v tabeli page glede na pripadnost domeni (zapisu v tabeli `site`) za `evem.gov.si` in `e-prostor.gov.si`

page_type	število
BINARY	1043
FRONTIER	113734
HTML	14100

Table 4: Število zapisov v tabeli page glede na `page_type` za vse `*.gov.si` domene

data_type	število
DOC	176
PPT	3
PDF	778
PPTX	7
DOCX	79

Table 5: Število zapisov v tabeli page glede na `data_type` (format) za vse `*.gov.si` domene

5.2 Na vseh `*.gov.si` domenah

V naš spletni pajek smo na začetku poleg štirih podanih domen dodali še 5 naključno pridobljenih s spleta. Po dodajanju teh v bazo smo zagnali naš pajek na dveh napravah z večjim številom procesov. Že po nekaj minutah je bila naša baza napolnjena z več kot 30 podomenami družine `*.gov.si`, ki so cilj povezav z naših začetnih strani. Po 20 minutah izvajanja pa je pajek našel 150 strani, do prekinitve izvajanja pa kar 171 poddomen.

V tabeli 4 so predstavljeni podatki o številu zapisov v tabeli `page` glede na `page_type` za vse `*.gov.si` domene, v tabeli 5 pa število binarnih datotek glede na `data_type` (format datotek).

Povprečno število enoličnih povezav na strani je $49,865197988$, povprečno število slik na strani pa je $1,5$.

6 Zaključek

V tej seminarski nalogi smo razvili delujoči spletni pajek za pobiranje vsebin strani iz domen `*.gov.si`. Pajek je prilagojen tako, da bi ga enostavno lahko preusmerili tudi na druge ali dodatne domene, prav tako pa je tudi pripravljen na delovanje na več napravah hkrati. Pajek poleg vsebin strani shranjuje tudi dokumente zahtevanih formatov, slike in povezave, ki jih strani vsebujejo.

References

[1] Gov crawler visualizer. https://zigacernigoj.github.io/GOV-crawler-visualizer/index.html. [Dostopano dne 02.04.2019].

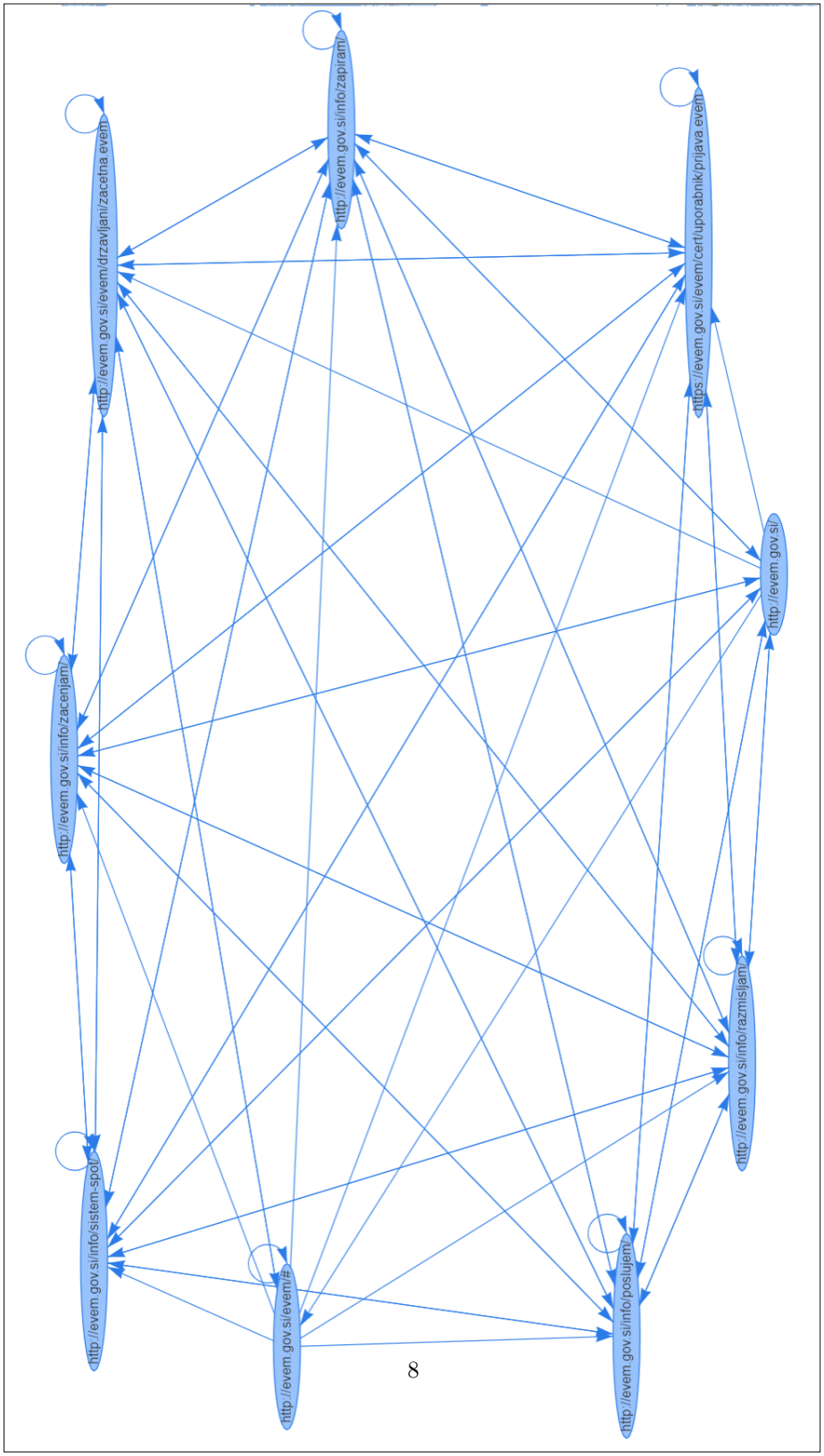

Figure 1: Primer povezav na spletnih straneh `evem.gov.si`