Poročilo 1. domače naloge - spletni pajek

I. Uvod

Cilj domače naloge je bila izgradnja spletnega pajka, ki podatke pobira s strani na domenah *.gov*. Pajek mora pridobiti vsebino HTTP strani, iz njih izvleči slike in povezave, zaznavati duplikate strani in hraniti URLje, ki so trenutno v čakalni vrsti. Vse podatke, ki jih pajek pridobi, mora shraniti v podatkovni bazi s točno določeno strukturo, kjer v tabelah hranimo podatke o obiskanih in pridobljenih straneh in domenah. Pajek pri dodajanju strani v frontier uporablja strategijo iskanja v širino (breadth-first).

II. Implementacija

Pajek je implementiran v programskem jeziku Python z uporabo knjižnic selenium, urllib, beautiful soup, psycopg2, concurrent futures in hashlib. Inicializiramo ga s podatki o skupnem številu preiskovanih strani in številu niti, s katerimi bo pajek opravljal delo.

V vsaki iteraciji pri iskanju strani najprej izberemo stran iz frontierja, katero naj pajek preišče. Pri tem upoštevamo časovno omejitev, ki je vezana na IP naslov strežnika, ki gosti preiskovano stran, pri čemer vodimo evidenco o trenutno zasedenih naslovih. V ta namen najprej naredimo poizvedbo o IP naslovu strežnika strani. Če ta naslov sovpada z trenutno že zasedenimi naslovi, nadaljujemo z iskanjem nezasedene strani znotraj frontierja. S tem spremenimo vrstni red obdelave in dodajanja novih strani v frontier, kar pomeni manjši odmik od strategije iskanja v širino, vendar tako zelo povečamo učinkovitost posameznih niti pajka. Hranimo tudi trenutno obdelovane strani, s čimer preprečimo večkratno obdelovanje posamezne strani na različnih nitih.

Če naletimo na stran, katere domene še nimamo zabeležene znotraj podatkovne baze, jo vstavimo ter za naslednjo iteracijo rezerviramo dostop na stran robots.txt domene. Če stran robots.txt obstaja, zapišemo njeno vsebino v podatkovno baza poleg morebitne mape spletišča (sitemap). Vsakič, ko obiščemo določeno domeno, pred obdelavo strani preverimo, ali imamo dovoljen dostop. Če ga nimamo, stran označimo s statusno kodo 403 in njeno vsebino pustimo prazno. Če ne dobimo odziva od strani, ki jo preiskujemo, jo označimo s statusno kodo 404, sicer pa 200.

Za vsako stran izračunamo zgoščevalno funkcijo njene vsebine, katero hranimo znotraj podatkovne baze. Pri preverjanju duplikatov strani se namesto celotne vsebine primerjajo le zgoščevalni funkciji obeh strani. V primeru ujemanja je nova stran označena kot duplikat in ustvari se povezava na izvirnik strani. Če je stran originalna, se v podatkovni bazi ustrezno posodobi.

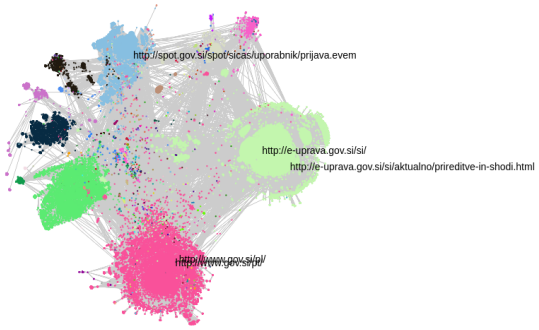
Po vstavljanju strani v podatkovno bazo se znotraj vsebine strani iščejo povezave na še neobiskane strani. Znotraj html elementov `<a>`, `<link>` in `<area>` se iščejo povezave strani znotraj polja *href*. Če je naslov relativen, ga razširimo s potjo trenutne strani, pri čemer ignoriramo relativne strani znotraj `<link>` elementov, saj ti vsebujejo resurse za nalaganje strani. Prav tako iščemo tudi povezave znotraj dogodkov *onclick*, ki se nahajajo v poljih *location.href* in *document.location*. Najdene povezave normaliziramo, s čimer poizkušamo preprečiti redundantnost strani, ter jih shranimo v tabelo *link*. Vsebino strani preiščemo tudi za slike, ki se nahajajo znotraj html elementa `<img>` in jih shranimo v tabelo *image*.

III. Rezultati

TABELA I
Statistika delovanja pajka

	gov.si	**spot**	**e-uprava**	**e-prostor**	**skupaj**
strani	6129	8185	708	8372	50195
duplikati	22	0	108	959	8440
domene	1	1	1	1	239
binarne datoteke	159	0	162	675	1737
slike	22344	7782	1440	10075	173442
slike na stran	3,64	0,95	2,03	1,20	3,46
povezave	49402	204773	2458	90962	473046

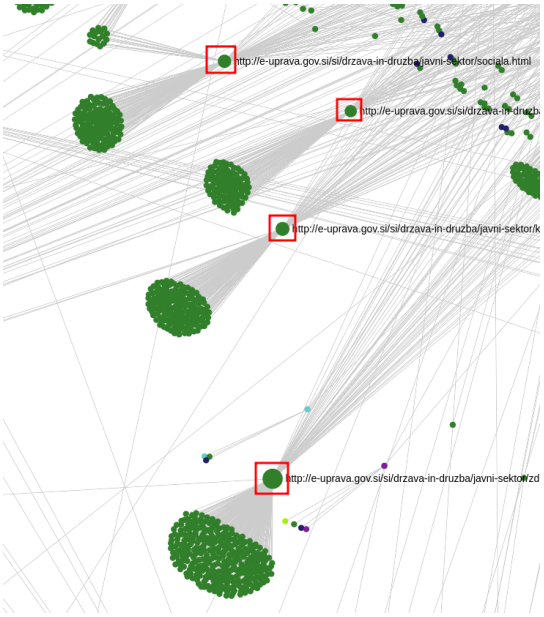
V tabeli 1 vidimo, da je pajek predvsem pri e-zdravju našel veliko povezav na druge strani, a ta domena nima veliko slik, kar ne preseneča. Nasprotno ima skupna *gov* domena mnogo več slik. E-uprava nima veliko povezav na druge strani, ker je namenjena predvsem opravilom znotraj te domene.



Slika 1. Pregled celotnega omrežja strani z označenimi clustri.

Strani znotraj domene so med seboj močno povezane, kar vodi do velikega števila duplikatov, ker pa gre za vladne strani, so po pričakovanjih tudi domene med seboj povezane s številnimi povezavami (slika 1).

Na sliki 2 je zanimivo, da najdemo primere strani, ki so očitno namenjene predvsem povezavam, da se uporabnik lahko



Slika 2. Primeri strani z veliko povezavami na druge znotraj domene.

nadaljnje informira o temah, ki jih pokriva področje državne domene. Veliko teh strani so najverjetneje razni dokumenti.

IV. Zaključek

Če bi za izdelavo pajka imeli na voljo dodaten čas, bi bilo mogoče izvesti več testiranja, da bi preprečili morebitne napake pri branju strani. Ker s Seleniumom nismo imeli prejšnjih izkušenj, bi bile izboljšave možne tudi pri boljši uporabi tega orodja. Predvsem pa bi bilo dobro, če bi lahko optimizirali velikost kočne podatkovne baze.