SVEUČILIŠTE U ZAGREBU
GEODETSKI FAKULTET

Nikolina Vidonis

SEMANTIČKO OBOGAĆIVANJE TEKSTA PROSTORNIM INFORMACIJAMA METODAMA PROCESIRANJA PRIRODNOG JEZIKA

Diplomski rad

Zagreb, 2014

I. Autor

Ime i prezime:	Nikolina Vidonis
Datum i mjesto rođenja:	13. travnja 1990., Kopar, Slovenija

II. Diplomski rad

Predmet:	Analiza prostornih podataka
Naslov:	Semantičko obogaćivanje teksta prostornim informacijama metodama procesiranja prirodnog jezika
Mentor:	prof. dr. sc. Damir Medak
Voditelj:	Dražen Odobašić, dipl. ing.

III. Ocjena i obrana

Datum zadavanja zadatka:	14. siječnja 2014.
Datum obrane:	29. rujna 2014.

Sastav povjerenstva pred kojim je branjen diplomski rad:

prof. dr. sc. Damir Medak
prof. dr. sc. Drago Špoljarić
dr. sc. Mario Miler

Zahvale

Ovom prilikom htjela bih zahvaliti mentoru prof. dr. sc. Damiru Medaku i asistentu Draženu Odobašiću, dipl. ing. geod., koji su me vodili kroz izradu ovog diplomskog rada. Svojim strpljenjem i savjetima doprinijeli su većoj kvaliteti ovog rada. Tijekom izrade rada susrela sam se s mnogim problemima koje sam uz njihovu pomoć uspješno riješila. Hvala im što su me svojim znanjem i otvorenim pristupom inspirirali da naučim više.

Zatim bih htjela zahvaliti svojim roditeljima na svoj podršci koju su mi pružili kako tokom studiranja, tako i za vrijeme izrade i pisanja diplomskog rada. Hvala vam što ste svo ovo vrijeme vjerovali u mene.

Na kraju zahvaljujem kolegi Filipu Todiću koji mi je bio velika tehnička i moralna podrška.

Sažetak

Semantičko obogaćivanje je dodavanje dodatnih informacija već postojećem skupu podataka. Prirodni jezici su jezici koje koriste ljudi (npr. hrvatski, engleski, španjolski, itd.) te se prirodno razvijaju. S druge strane, formalni jezici, poput programskih jezika, su jezici koji su dizajnirani za specifične svrhe. Cilj ovog rada je odrediti prostornu dimenziju internetskih članaka korištenjem metoda procesiranja prirodnog jezika. Kao izvori podataka korištena su četiri web portala, dok je za referentnu bazu prostornih podataka korištena baza geografskih imena GeoNames. Algoritam je razvijen korištenjem programskog jezika Python te brojnih dodatnih modula. Nakon testiranja, odabran je Jaro Distance algoritam u svrhu obrade prirodnog teksta. Na kraju su prikupljeni podaci vizualizirani pomoću karata žarišta na tjednoj osnovi te je pripremljen videozapis.

Ključne riječi: semantičko obogaćivanje, prirodni jezici, prepoznavanje naziva, Python, Jaro Distance algoritam

Abstract

Semantic enrichment is an enrichment of the existing data set with additional information. Natural languages are languages used by people (eg. Croatian, English, Spanish, etc.). They are developed naturally. On the other hand, formal languages, such as programming languages, are languages that are designed for specific purposes. The aim of this study is to determine whether or not spatial references mentioned within Internet articles can be located using methods for processing natural languages. The information sources are four web portals while the reference spatial database is the geographical names database GeoNames. An algorithm was developed using the Python programming language and its additional modules. After testing, the Jaro Distance algorithm was selected for processing natural text. In the end the collected data was visualized using heat maps generated on a weekly basis as shown in the prepared video.

Key words: semantic enrichment, natural languages, named-entity recognition, Python, Jaro Distance algorithm

Sadržaj

1. Uvod

Skoro sve što se događa, događa se negdje. Znati gdje se nešto događa može biti od vitalne važnosti.

Longley, P. A., Goodchild, M. F., Maguire, D. J., Rhind D. W.

U današnje vrijeme Internet je bogat i vrijedan izvor podataka, pa tako i prostornih. Puno internetskih stranica sadrži tekstualni opis lokacije poput adrese, naziva grada i dr. Ovi opisi podataka su vrijedne geografske informacije u slučaju da se pronađe pravi način kako ih iskoristiti. Primjerice, web portali, Facebook i Twitter su mjesta na kojima se gotovo u realnom vremenu može doznati što se događa i gdje. Korištenje mobilnih tehnologija za pristup Internetu znatno povećavaju količinu dostupnih podataka.

Primjerice, za vrijeme neke elementarne nepogode (potresa, požara, poplave) na društvenim mrežama poput Facebooka i Twittera se u realnom vremenu i s mjesta događaja postavljaju statusi, slike i filmovi. To olakšava i ubrzava otkrivanje štete koju je nepogoda prouzročila, mjesta na kojima je možda potrebno tragati za ljudima i dr.

Jedan od primjera prikupljanja geografskih informacija iz tekstualnih podataka je primjer pružanja humanitarne pomoći nakon što je potres pogodio Haiti 2010. godine. Budući da su postojeće karte Haitija u trenutku potresa bile zastarjele, pomoć nije mogla biti pružena učinkovito. Kako je većina telekomunikacijske mreže ostala netaknuta nakon potresa, stanovnici su mogli tim putem tražiti pomoć. U tom su se trenutku uključili Ushahidi i OpenStreetMap. Ushahidi je platforma otvorenog koda za krizne situacije koja prikuplja slobodno dostupne informacije od građana (eng. *crowd-sourced*) putem društvenih mreža, SMS-a, itd. Uspostavljena je Ushahidi platforma za Haiti i broj na koji su stanovnici mogli slati SMS poruke. Tako prikupljeni podaci su agregirani i kartirani i dobivena je slika stvarne situacije na terenu. U međuvremenu su dobrovoljci na osnovu doniranih satelitskih snimaka (prije i nakon potresa) kartirali zahvaćeno područje u OpenStreetMap projektu. SMS poruke koje su pristizale na Ushahidi su bile na kreolskom jeziku tako da su okupljani volonteri koji ga znaju kako bi se informacije uspješno prevele i kartirale. Zahvaljujući Ushahidi platformi sposobnost kartiranja je drastično povećana što je omogućilo bolje akcije spašavanja. U tom je pothvatu sudjelovalo preko 2000 volontera i 20 organizacija iz 49 zemalja (URL1).

Slično je i s web portalima. Nove vijesti objavljuju se u svako doba dana. Ponekad su to vijesti koje svakih nekoliko minuta izvještavaju o novostima na nekom događaju (npr. utakmice i koncerti). Ovaj rad obrađivat će samo web portale kao izvore prostornih podataka.

1.1. Obogaćivanje teksta geografskim informacijama

Geoportali su napravljeni kako bi funkcionirali kao glavna stanica u pronalaženju, procjeni i početku korištenja geografskih podataka. Ipak, trenutni geoportali se suočavaju s problemima u optimizaciji procesa zbog semantičke heterogenosti. To dovodi do niskog odziva i niske preciznosti u tekstualnim pretraživanjima. Stoga su Bernhard Vockner i Manfred Mittlböck predložili alternativni pristup semantičkom pronalaženju koji podržava višejezičnost i informacije iz domene konteksta (Vockner & Mittlböck, 2014).

Autori Vockner i Mittlböck (Vockner & Mittlböck, 2014) izradili su prototip geoportala gdje su lokacije određene i osvježavane na dnevnoj bazi koristeći skriptu napisanu u programskom jeziku Python koja koristi NUTS (eng. *Nomenclature of territorial units for statistics*) i LAU2 (eng. *Local Administrative Unit*) regije Europske Unije. Nazive lokacija proširuju prostornim podacima dok su metapodaci preuzeti iz kataloga njihove implementacije geoportala. Cilj je na osnovu imena lokacije što točnije geolocirati lokaciju pomoću programskog jezika Python i biblioteke geopy. Primjerice, ako korisnik traži *Hallein* (mali grad u Austriji), također će dobiti rezultat *Salzburg* jer se Hallein nalazi u saveznoj državi *Salzburg*. Izazovi nastaju kada je korišten isti naziv grada i savezne države (npr. grad Salzburg u saveznoj državi Salzburg). U tim slučajevima može se koristiti kontekst kako bi se pronašlo točno rješenje. Koristi li se kombinacija riječi *grad* i *Salzburg*, sustav može zaključiti da se radi o gradu Salzburgu.

Autori Vockner i Mittlböck su za cijele tekstove koristili metode prepoznavanja naziva (eng. *Named Entity Recognition* - NER), kao što je Natural Language Toolkit (NLTK) i prilagođenu verziju Geodict biblioteke kako bi se pronašle i izdvojile ključne riječi za određivanje lokacije. Prepoznavanje naziva je podvrsta zadatka izvlačenja informacija.

2. Materijali i metode

U sljedećem poglavlju slijedi detaljan opis pojmova i tehnologije primijenjenih u izradi diplomskog rada. Rad je najvećim dijelom izrađen pomoću programskog jezika Python, a vizualizacija podataka je izvedena pomoću QGIS i OpenShot Video Editor aplikacija. Također je detaljno opisan način prikupljanja i obrade ulaznih tekstualnih podataka. Riječ je o novinskim člancima preuzetim s četiri hrvatska web portala.

2.1. Semantičko obogaćivanje

Semantičko obogaćivanje (eng. *semantic enrichment*) (URL2) je dodavanje dodatnih informacija već postojećem skupu podataka. Može biti obogaćivanje teksta dodatnim podacima, označavanjem, kategoriziranjem ili klasificiranjem podataka povezivanjem s drugim podacima, rječnicima ili drugim izvorima podataka. Primjerice, dodavanje podataka o prometu (stanje na cestama, gužve, radovi na cesti, itd.) i prometnoj mreži.

Napredniji način razmišljanja o semantičkom obogaćivanju je korištenje relativno primitivnog oblika računalnog učenja (eng. *machine learning*). Sustav automatski poboljšava svoje razumijevanje sadržaja i konteksta podataka na osnovu prethodnih znanja. Svakoga dana sustav obogaćuje svoje razumijevanje okoline, što vodi do dubljeg uvida i potencijalno do ispravnijih projekcija i analiza.

2.2. Prirodni i formalni jezici

Prirodni jezici su jezici koje koriste ljudi (npr. hrvatski, engleski, španjolski, itd.). Prirodno se razvijaju, a ljudi pokušavaju nametnuti pravila (Downey, 2013).

Formalni jezici su jezici koje su dizajnirali ljudi za specifične svrhe. Primjerice, zapis koji koriste matematičari je formalni jezik koji je posebno dobar za obilježavanje odnosa među brojevima i simbolima. Kemičari koriste formalni jezik za prikaz kemijske strukture molekula. Programski jezik je formalni jezik dizajniran za izraz proračuna (eng. *computations*).

Formalni jezici imaju stroga pravila za sintaksu. Na primjer, matematički izraz *3+3=6* je sintaktički točan izraz, dok *3+=3$6* nije. Isto tako, H_2O je sintaktički korektan kemijski izraz, dok $_2Zz$ nije.

Pravila sintakse dolaze u dva oblika, vezana za tokene i vezana za strukturu. Tokeni su osnovni elementi svakog jezika poput riječi, brojeva i kemijskih elemenata. Jedan od problema s

izrazom *3+=3$6* je što *$* nije dozvoljen matematički znak. Slično, izraz $_2Zz$ nije pravilan jer ne postoji kemijski element čiji je simbol *Zz*.

Druga vrsta pravila sintakse se odnosi na strukture izjava, odnosno način na koji su znakovi složeni. Izraz *3+=3* nije sintaktički točan izraz jer usprkos činjenici da su svi znakovi dozvoljeni, + i = ne smiju biti jedan do drugoga. Slično, indeks $_2$ u kemijskoj formuli mora se nalaziti iza simbola elementa, nikako ispred.

Dok se čita izjava na nekom prirodnom jeziku, ili na nekom formalnom jeziku, treba razmišljati o strukturi rečenice. Kod prirodnih jezika riječ je o nesvjesnom procesu, a postupak se zove raščlanjivanje (eng. *parsing*).

Primjerice, kada čujemo rečenicu *Upalila se žarulja* jasno nam je da je *žarulja* subjekt, a *se upalila* predikat. Nakon raščlanjivanja rečenice može se shvatiti njeno značenje, odnosno semantika rečenice. Uz pretpostavku poznavanja značenja imenice *žarulja* i značenja glagola *upaliti se*, može se razumjeti i opće značenje rečenice.

Iako prirodni i formalni jezici imaju puno zajedničkih značajki (npr. tokeni, struktura, sintaksa, semantika itd.), u nekim se stvarima ipak razlikuju:

- *dvosmislenost*: prirodni jezici su puni dvosmislenosti, koje ljudi rješavaju koristeći kontekstualne informacije. Formalni jezici su dizajnirani da budu skoro ili uvijek jednoznačni, što znači da svaka izjava ima točno jedno značenje, bez obzira na kontekst;
- *redundantnost*: da bi se nadoknadila neodređenost i smanjili nesporazumi, prirodni jezici obiluju s redundantnosti. Rezultat toga je što su oni često opširni. Formalni jezici su manje redundantni i smisleniji;
- *doslovnost*: Prirodni jezici su puni metafora i izraza. Kad se kaže *Upalila se žarulja*, postoji mogućnost da se nije upalila žarulja, već da je netko smislio dobru ideju. Formalni jezici znače točno ono što piše, bez prenesenog značenja.

Formalni jezici su zbijeniji od prirodnih jezika, tako da treba dulje vrijeme da ih se pročita. Također, struktura formalnog jezika je vrlo važna. Stoga treba naučiti raščlaniti program u glavi, identificirati tokene i interpretirati strukturu. Kod formalnih jezika je bitan svaki detalj. Male pogreške u pisanju i interpunkcijskim znakovima, preko kojih se može preći u prirodnim jezicima, mogu jako puno značiti u formalnim jezicima.

2.3. Prepoznavanje naziva

Cilj izdvajanja informacija (eng. *Information Extraction*) je izdvajanje strukturiranih informacija o aktivnostima kompanija i sličnim aktivnostima iz nestrukturiranih podataka poput novinskih članaka. Primijećena je potreba za prepoznavanjem klasa (eng. *units*) poput:

- imena (osoba, organizacija i lokacija);
- brojčanih vrijednosti (datum, vrijeme, valute, postotne vrijednosti).

Prepoznavanje tih klasa je jedan od bitnijih podzadataka ekstrakcije informacija i naziva se *prepoznavanje naziva* (Nadeau & Sekine, 2007).

Prepoznavanje naziva (eng. *Named Entity Recognition – NER*) ima zadatak u tekstu prepoznati riječi koje se odnose na određene objekte (npr. osobe, organizacije i lokacije). Pronalaženje naziva dijeli se na tri metode (Liu, Wei, Zhang, & Zhou, 2013):

- temeljeno na pravilima,
- temeljeno na strojnom učenju (eng. *machine learning*) i
- hibridna metoda.

Trenutno se NER najviše fokusira na formalne tekstove kao što su novinski članci, no postoje i studije koje se bave neformalnim tekstovima poput blogova, emailova, kliničkih bilježaka, twitter objava itd. (Liu, Wei, Zhang, & Zhou, 2013)

Većina pristupa zahtijeva potpuno poklapanje barem jedne riječi u nazivu. Ipak, dobro je omogućiti fleksibilnost u uvjetima koje trebaju zadovoljiti riječi kako bi se proglasile jednakima. U NER području koriste se bar tri takva pristupa (Nadeau & Sekine, 2007):

- riječima se prije uspoređivanja mogu maknuti svi sufiksi. Tako bi primjerice riječ *tehnologije* bila pozitivno poklapanje s riječi *tehnologija* koja se nalazi u rječniku;
- riječi se mogu uspoređivati tako što se zada granična razlika i onda se usporede koristeći Levenshteinov ili Jaro-Winklerov algoritam. To omogućava uključivanje malih leksičkih razlika koje nisu nužno sufiksi. Jaro-Winkler algoritam je posebno dizajniran za traženje poklapanja među imenima, prateći opažanje da su početna slova točna, dok se pri kraju riječi događaju promjene;
- treći pristup je pomoću Soundex algoritma. Soundex algoritam je fonetički algoritam koji indeksira riječi prema tome kako se izgovaraju na engleskom jeziku. Taj kod je

kombinacija prvog slova riječi i troznamenkastog broja. Primjerice, slična imena kao *Lewinskey* i *Lewinsky* imaju jednak Soundex kod, *l520*.

2.3.1. Algoritmi za obradu prirodnih jezika

U sljedećim poglavljima će biti detaljnije objašnjeni različiti algoritmi koji se koriste za obradu prirodnog jezika. Riječ je o algoritmima dostupnima u Jellyfish i FuzzyWuzzy bibliotekama za programski jezik Python. Teorija iz tih algoritama je u nastavku detaljno obrađena, ali zajednička karakteristika svih algoritama je usporedba riječi po sličnosti. Pritom je potrebno imati testni uzorak koji se uspoređuje se referentnom osnovom koja se uzima kao bespogrešna.

Levenshtein Distance i Damerau-Levenshtein Distance

Levenshtein algoritam, također poznat i kao eng. *Edit-Distance* računa najmanji broj promjena koje je potrebno napraviti kako bi se jedan niz znakova pretvorio u drugi. Najčešći način računanja Levenshteinovog algoritma je korištenjem dinamičkog programiranja. Dinamičko programiranje je pristup rješavanju složenih problema njihovim razlaganjem na više jednostavnijih problema (URL3).

Stvara se matrica dimenzija [*m*, *n*], gdje je *m* broj slova u prvoj riječi, a *n* broj slova u drugoj riječi. Matrica se popunjava od gornjeg lijevog kuta prema donjem desnom kutu. Svaki skok horizontalno ili vertikalno odgovara umetanju slova ili njegovom brisanju. Svaka promjena ima vrijednost (eng. *cost*) 1. Dijagonalni skok ima vrijednost 1 ako se znakovi u retku i stupcu razlikuju ili 0 ako su jednaki. U svaku ćeliju se upisuje najmanja moguća vrijednost. Tako je broj u donjem desnom kutu vrijednost Levenshteinove udaljenosti između dvije riječi. Na slici 1 prikazan je primjer pretvorbe riječi "*meilenstein*" u "*levenshtein*":

		m	e	i	l	e	n	s	t	e	i	n
	0	1	2	3	4	5	6	7	8	9	10	11
l	1	1	2	3	3	4	5	6	7	8	9	10
e	2	2	1	2	3	3	4	5	6	7	8	9
v	3	3	2	2	3	4	4	5	6	7	8	9
e	4	4	3	3	3	3	4	5	6	6	7	8
n	5	5	4	4	4	4	3	4	5	6	7	7
s	6	6	5	5	5	5	4	3	4	5	6	7
h	7	7	6	6	6	6	5	4	4	5	6	7
t	8	8	7	7	7	7	6	5	4	5	6	7
e	9	9	8	8	8	7	7	6	5	4	5	6
i	10	10	9	8	9	8	8	7	6	5	4	5
n	11	11	10	9	9	9	8	8	7	6	5	4

Slika 1. *Primjer pretvorbe riječi meilenstein u levenshtein*

Postoje dva moguća redoslijeda promjena koje donose najmanju vrijednost Levenshteinove udaljenosti i prikazana su u tablici 1.

Tablica 1. *Prikaz dva različita redoslijeda pretvorbe riječi levenshtein u meilenshtein*

m	**→**	**l**	**m**	**→**	**l**
e	=	e	e	=	e
i	**-**		**i**	**→**	**v**
l	**→**	**v**	**l**	**-**	
e	=	e	e	=	e
n	=	n	n	=	n
s	=	s	s	=	s
	+	**h**		**+**	**h**
t	=	t	t	=	t
e	=	e	e	=	e
i	=	i	i	=	i
n	=	n	n	=	n

Gdje je „→“ promjena, „+“ umetanje, „-“ izbacivanje.

Levenshteinova udaljenost je metrička udaljenost, tj. zadovoljava uvjete nejednakosti trokuta (zbroj duljina dviju stranica veći je od duljine treće stranice). Za većinu drugih algoritama za uspoređivanje nizova znakova to ne vrijedi (URL4).

Damerau-Levenshtein udaljenost (eng. *Damerau-Levenstein distance*) je varijacija Levenshteinove udaljenosti, kod koje je osim umetanja, izbacivanja i promjene dozvoljena i zamjena dva susjedna znaka.

Jaro Distance i Jaro-Winkler Distance

Jaro Distance algoritam je posebno dobar izbor kada se uspoređuju kratki nizovi znakova, kao što su primjerice imena i prezimena, za potrebe povezivanja zapisa. To je stoga što je relativno otporan na zamjenu mjesta slovima. Također, riječi su sličnije što je sličniji početak riječi (URL5).

Jaro-Winklerov algoritam sastoji se od dva dijela, originalnog Jarovog algoritma i Winklerovog dodatka. Izraz kojim se računa Jaro udaljenost je:

$$d_j = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right).$$

Izraz se sastoji od aritmetičke sredine triju izračuna:

- omjer podudarajućih slova *m* i duljine prvog niza znakova s_1
- omjer podudarajućih slova *m* i duljine drugog niza znakova s_2
- omjer broja slova ne-mijenjajućih slova (eng. *non-transpozition*s) i broja podudarajućih slova *m*

Kako zapise unose ljudi, očito je da se dogode i pogreške u pisanju. William Winkler je zaključio da je vjerojatnije da će se pogreška dogoditi pri kraju niza znakova. Njegov dodatak Jarovoj formuli oprašta neke pogreške pri kraju niza znakova u skladu sa sličnostima prvih nekoliko znakova. Tako je Winkler dobio značajno bolje rezultate uz malo dodatno opterećenje. Winklerova formula glasi (URL6):

$$d_w = d_j + (l_p(1 - d_j))$$

gdje je:

- d_w – Jaro-Winkler udaljenost
- d_j – Jaro udaljenost
- l_p – težinska vrijednost

Rezultat Jaro-Winklerove udaljenosti će uvijek biti veći od rezultata Jaro udaljenosti, no i dalje će biti u rasponu od 0 do 1. Težinska vrijednost l_p sastoji se od dvije varijable. Vrijednost *l* je broj početnih slova koja se poklapaju, a maksimalna vrijednost je četiri. Vrijednost *p* je težinska vrijednost koja može iznositi maksimalno ¼. Kako se *p* povećava, sve će se više propusta popuniti. Kada *p* iznosi ¼ nizovi znakova kojima su prva četiri znaka jednaka će uvijek rezultirati idealnim poklapanjem. Kako ne želimo da JOHN i JOHNSON budu označeni kao savršeno poklapanje, tj. kao identični nizovi znakova, nije dobro koristiti ¼ kao vrijednost *p*. Nakon mnogo istraživanja, Winkler preporuča korištenje vrijednosti 0.1 za *p*.

Soundex

Soundex algoritam generira kod od četiri znaka. Kod je baziran na izgovoru riječi na engleskom jeziku. Taj kod se može koristiti za usporedbu dvije riječi te kako bi se utvrdilo zvuče li one podjednako. Ovaj algoritam je koristan u pretraživanju podataka u bazi podataka ili tekstualnim datotekama, naročito u slučajevima kada se pretražuju imena koja su često pogrešno napisana.

Soundex algoritam koristi niz pravila kako bi niz znakova pretvorio u kod od četiri znaka.

Koraci za pretvorbu su sljedeći:

- zanemariti sve znakove u nizu znakova osim onih koji se nalaze u engleskoj abecedi
- prvi znak Soundex koda je prvo slovo niza znakova koji se pretvaraju u kod
- nakon prvog slova u nizu znakova, ne kodiraju se samoglasnici ni suglasnici h, w i y. Ti znakovi mogu utjecati na kod svojom prisutnošću no ne kodiraju se direktno
- dodijeliti broj između 1 i 6 svim slovima nakon prvog na sljedeći način:
    - 1: B, F, P, V
    - 2: C, G, J, K, Q, S, X, Z
    - 3: D, T
    - 4: L
    - 5: M, N
    - 6: R
- u slučaju da su susjedna slova ista, uklanjaju se sva osim jednog, osim ako su samoglasnici, h, w ili y između njih u početnom nizu znakova
- prisiliti kod da se sastoji od četiri znaka dodavanjem nula ili skraćivanjem

Postoje varijacije u Soundex algoritmu tako da može biti teško uspoređivati Soundex kodove generirane od strane različitih sustava. Jedna od tih varijacija je identificirati kada su prvih nekoliko suglasnika u riječi kodirana kao isti broj i zatim obrisati taj broj. U drugoj varijaciji se slova h, w i y tretiraju drugačije od samoglasnika. Potpuno ih se zanemaruje tako da se brišu dupli znakovi koda koji su razdvojeni jednim od ta tri znaka (URL7).

2.4. Python

Python je programski jezik visoke razine koji nudi jednostavan, ali učinkovit pristup objektno-orijentiranom programiranju. Nastao je u ranim 1990-im godinama kao nasljednik ABC programskog jezika. Razvio ga je Guido van Rossum, koji ostaje njegov glavni autor, uz doprinose mnogih drugih autora. Razvijen je pod licencom otvorenog koda (eng. *open source*) odobrene od strane OSI-ja (eng. *The Open Source Initiative*), što znači da se može slobodno koristiti i distribuirati, čak i komercijalne svrhe. Dostupan je za sve veće operativne sustave - Windows, Mac OS X, Linux/Unix (URL8).

Python ima odlične funkcije za procesiranje jezičnih podataka. Primjerice:

```
for line in open("file.txt"):
  for word in line.split():
  if word.endswith('ing'):
        print word
```

Ovaj kratak program prikazuje neke glavne značajke Python programskog jezika. Kao što je već spomenuto, Python je objektno-orijentiran, što znači da je svaka varijabla subjekt (eng. *entity*) koji ima definirane atribute i metode. Primjerice, vrijednost varijable *line* je više nego slijed znakova. To je tekstualni objekt koji ima metodu ili operaciju *.split()* koja omogućava podjelu reda na riječi. Metode imaju argumente koji se pišu unutar zagrada, npr. *word.endswith('ing')* ima argument 'ing' koji označava željeni kraj riječi. Python je jednostavan za čitanje, tako da nije teško pretpostaviti što program gradi, čak i nekome tko nije nikad programirao. Na kraju, ovaj program čita datoteku file.txt i ispisuje sve riječi koje završavaju na „ing" (Bird, Loper, & Klein, 2009).

2.4.1. Requests

Requests je jednostavna i elegantna Python biblioteka za obradu HTTP-a. Omogućava jednostavan integraciju aplikacija napisanih u Python programskom jeziku s web servisima. Nema potrebe za ručnim dodavanjem parametara upita URL-u ili kodiranjem formata poslanih podataka.

Koristi ga Vlada Ujedinjenog Kraljevstva, Amazon, Google, Mozilla, Twitter. Preuzet je preko 12 milijuna puta. (URL9)

Nekoliko primjera funkcija koje omogućava Requests modul:

```
import requests
r = requests.post("http://httpbin.org/post")
r = requests.put("http://httpbin.org/put")
r = requests.delete("http://httpbin.org/delete")
```

2.4.2. BeautifulSoup

BeautifulSoup je Python biblioteka dizajnirana za brzu izradu projekata kao što je transformacija prikazanih podataka iz jednog sučelja u drugo (modernije sučelje) (eng. *screen-scraping*) (URL10).

Tri su značajke koje ju čine tako moćnom:

- BeautifulSoup nudi nekoliko jednostavnih metoda i Python idiome (eng. *Pythonic idioms*) za navigaciju, pretraživanje i modifikaciju raščlanjivog stabla(eng. *parse tree*). Riječ je o alatu za seciranje dokumenta i izvlačenje traženih informacija.
- BeautifulSoup automatski pretvara ulazne dokumente u Unicode i izlazne u UTF-8. Korisnik se ne treba zamarati enkripcijom, osim ako u dokumentu nije naznačena enkripcija i BeautifulSoup ne može raspoznati koja je. Tada korisnik samo treba specificirati o kojoj je vrsti enkripcije riječ.
- BeautifulSoup je na vrhu popularnih Python raščlanjivača (eng. *parser*) kao što su *lxml* i *html5lib*, što omogućava isprobavanje različitih strategija raščlanjivanja i izbor između brzine i fleksibilnosti.

BeautifulSoup će raščlaniti sve ulazne podatke i obuhvatiti cijelo stablo. Primjeri zadataka koje ova biblioteka može riješiti:

- pronalazak svih linkova
- pronalazak svih linkova koji imaju određenu klasu
- pronalazak naslova tablice koji je napisan podebljanim slovima
- i sl.

2.4.3. NLTK

Natural Language Toolkit (NLTK) je platforma za izgradnju Python programa koji će obrađivati prirodne jezike. Pruža sučelje koje je jednostavno za korištenje s preko 50 korpusa i leksičkih izvora kao što je WordNet. Uz to nudi i paket biblioteka za obradu teksta koje nude mogućnosti klasifikacije, tokenizaciju, označavanje, raščlanjivanje i semantičko razmišljanje (eng. *semantic reasoning*). NLTK je dostupan za Windows, Mac OS X i Linux operativne sustave. Slobodan je i otvorenog koda (Bird, Loper, & Klein, 2009).

2.4.4. Biblioteke za obradu teksta

U nastavku će biti detaljnije objašnjeni FuzzyWuzzy i Jellyfish biblioteke za programski jezik Python koje su korištene u izradi ovog diplomskog rada. Svrha obje biblioteke je usporedba riječi po sličnosti, ali su im pristupi drugačiji.

FuzzyWuzzy

Fuzzy uparivanje (eng. *fuzzy matching*) je opći naziv za pronalaženje nizova znakova koji su gotovo jednaki, skoro isti. „gotovo“ i „skoro“ su nejasni pojmovi, a o svrsi projekta ovisi što će „gotovo“ i „skoro“ značiti (URL11).

FuzzyWuzzy je Python biblioteka za uspoređivanje nizova znakova. Ima različite funkcije uspoređivanja, a koja će biti odabrana ovisi o tome što za pojedini projekt znače „gotovo“ i „skoro“.

FuzzyWuzzy sadrži nekoliko različitih funkcija za uspoređivanje nizova znakova (URL12):

- *ratio*(s1,s2)– implementacija Levenshtein Distance algoritma;
- *partial_ratio*(s1,s2) – kada su dva niza znakova prilično različite duljine, traži se najbolje podudaranje između ta dva niza duljine kraćeg.
- *token_sort_ratio*(s1,s2) – razdvaja nizove znakova na tokene (riječi) koje zatim poreda po abecedi te ih ponovno spoji u jedan niz. Nakon toga se nizovi znakova uspoređuju pomoću ratio() funkcije
- *token_set_ratio*(s1,s2) – oba niza znakova razdvoji na tokene te ih podijeli u dvije grupe, preklop i ostatak. Sličnost je veća što se više tokena nalazi u grupi preklop.

Jellyfish

Jellyfish je Python biblioteka za približno i fonetsko uspoređivanje teksta (niza znakova). Sadrži slijedeće algoritme za uspoređivanje teksta (URL13):

- Levenshtein Distance
- Damerau-Levenshtein Distance
- Jaro Distance
- Jaro-Winkler Distance
- Match Rating Approach Comparison
- Hamming Distance

2.4.5. GeoPy

GeoPy je Python klijent za nekoliko popularnih web servisa za geokodiranje. On omogućava jednostavno lociranje koordinata adresa, gradova i država na cijeloj Zemlji koristeći geokodere treće strane (eng. *third-party geocoders*) i druge izvore podataka. Neki od web servisa za geokodiranje koje je moguće koristiti su Google Maps, Bing Maps i Yahoo BOSS (URL14).

Primjer dobivanja koordinata iz adrese:

```
from geopy.geocoders import GoogleV3
geolocator = GoogleV3()
address, (latitude, longitude) = geolocator.geocode("175 5th Avenue NYC")
print(address, latitude, longitude)
```

Odgovor koji slijedi je:

```
175 5th Avenue, New York, NY 10010, USA 40.7410262 -73.9897806
```

2.5. HTML

HyperText Markup Language (u daljnjem tekstu: HTML) je jezik dizajniran za izradu internet stranica. Njime se definira kako će svaka internet stranica izgledati na korisnikovom ekranu. Pomoću njega se kreiraju veze s drugim stranicama. Veza (eng. *hyperlink*) je element u elektroničkom dokumentu pomoću kojeg se povezuje na drugo mjesto u dokumentu ili na neki drugi dokument. Najčešće se koristi na internetu ako bi se povezalo na druge dokumente, primjerice internet stranice, pdf dokumente i sl. (Stopper, Sieber, & Schnabel, 2008).

HTML dokumenti se sastoje od HTML tagova i običnog teksta. HTML tagovi su ključne riječi okružene izlomljenim zagradama, primjerice <html>. Najčešće dolaze u paru, početni i završni . Završni tag je jednak početnom, osim što prije ključne riječi se nalazi kosa crta. HTML elementi se pišu unutar HTML tagova (URL15).

Primjer HTML strukture:

```
<!DOCTYPE html>
<html>
        <head>
                <title></title>
        </head>
        <body>
                <h1></h1>
                <p></p>
                <p></p>
        </body>
</html>
```

2.6. QGIS

QGIS je GIS aplikacija otvorenog koda koja omogućuje vizualizaciju, editiranje i analiziranje rasterskih i vektorskih podataka. Projekt je započet u svibnju 2002. godine, kako bi se napravila alternativa komercijalnim programima slične svrhe. QGIS radi na većini Unix platformi, Windows i OS X. Razvijen je koristeći Qt toolkit i C++. QGIS ima jednostavno i ugodno grafičko korisničko sučenje (eng. *graphical user interface – GUI*).Teži biti dostupan korisnicima, pružajući uobičajene GIS funkcije i značajke. Početni cilj projekta bio je pružiti preglednik GIS podataka. QGIS je objavljen pod GNU General Public Licence (GPL) što znači da svatko može pregledati i promijeniti izvorni kod i osigurava da će uvijek biti slobodan. Osim već ugrađenih funkcija, moguće je dodati i nove pomoću dodataka (eng. *plugins*) (URL16).

2.7. GRASS GIS

GRASS GIS (eng. *Geographic Resources Analysis Support System*) je GIS program za upravljanje podacima, obradu slika, prostorno modeliranje i vizualizaciju velikog broja tipova podataka. To je slobodan program otvorenog koda objavljen pod GNU General Public Licence (GPL). GRASS GIS je službeni projekt OSGeo-a (eng. *Open Source Geospatial Fundation*). Sadrži preko 350 modula za izradu karata, rukovanje rasterskim i vektorskim podacima, mrežne analize, obradu multispektralnih snimki.

Razvoj je započet u laboratorijima američke vojske (eng. *U.S. Army Construction Engineering Research Laboratories*) kao alat za upravljanje zemljištem i prostorno planiranje. GRASS GIS je prerastao u moćan alat široke primjene u raznim područjima. Koristi se u akademske i komercijalne svrhe širom svijeta, a koriste ga i mnoge vladine agencije kao što je Američka svemirska agencija (eng. *National Aeronautics and Space Administration – NASA*), Američka nacionalna oceanska i svemirska agencija (eng. *National Oceanic and Atmospheric Administration – NOAA*), Američko ministarstvo poljoprivrede (eng. *U.S. Department of Agriculture – USDA*), kao i mnoge konzultantske tvrtke za okoliš (URL17).

2.8. Inkscape

Inkscape je aplikacija otvorenog koda za grafičku obradu vektora. Radi na Windows, Mac OS X i Linux platformama. Koristi se za izradu ikona, ilustracija, logo sličica, dijagrama, karata i web grafika. Ono što ga razlikuje od ostalih sličnih programa (Adobe Illustrator, Corel Draw

itd.) je što Inkscape kao temeljni format koristi Scalable Vector Graphics (SVG) format, otvoren W3C[1] standard temeljen na XML jeziku.

Inkscape ima sofisticirane grafičke alate s mogućnostima usporedivima s aplikacijama poput Adobe Illustrator ili CorelDraw. Može uvoziti i izvoziti različite formate, uključujući SVG, AI, EPS, PDF, PS i PNG. Ima opsežan skup alata, jednostavno sučelje i višejezičnu podršku. Dizajniran da bude proširiv tako da korisnik može Inkscape funkcionalnosti prilagoditi svojim potrebama i proširiti dodacima (URL18).

2.9. OpenShot Video Editor

OpenShot Video Editor je slobodan program otvorenog koda. Napisan je za Linux platformu. Objavljen je pod GPL licencom. OpenShot omogućava kreiranje filmova kombiniranjem videa, fotografija i glazbe te dodavanje podnaslova, prijelaza i efekata. Uz to, nudi i naprednije funkcije obrade videa, kao što su izrezivanje i korištenje samo dijelova već postojećeg filma, podešavanje razina zvuka, prijelaze između videozapisa, slaganje više slojevi videozapisa, dodavanje 3D efekata i slično. Odlikuje ga i korisničko sučelje jednostavno za korištenje. OpenShot omogućava spremanje videa u uobičajene formate, poput AVI, MP4, FLV, MOV i drugi (URL20).

2.10. Izvori podataka

U ovom poglavlju biti će prikazani i objašnjeni podaci koji su prikupljeni i korišteni kao ulazni podaci u ovom diplomskom radu. Glavni izvor podataka su članci prikupljeni web portala te nazivi gradova iz GeoNames baze geografskih imena. Tu su još OpenStreetMap podaci te granice županija Republike Hrvatske.

[1] World Wide Web Consortium (W3C) je međunarodna zajednica u kojoj članovi organizacije, zaposlenici i zajednica rade zajedno kako bi razvili Web standarde (URL19)

2.10.1. Internetske stranice

Kao izvor podataka odabrani su novinski članci sa sljedećih web portala:

- Istra News (http://www.istranews.in/);
- Glas Slavonije (http://www.glas-slavonije.hr/);
- Danas.hr (http://danas.net.hr/);
- Jutarnji list (http://www.jutarnji.hr/).

web portali su odabrani kao izvor podataka zato što je moguće napisati skriptu koja će provjeravati jesu li objavljeni novi podaci i po potrebi preuzeti ih. To eliminira ručno pretraživanje izvora podataka, primjerice listanje novina.

Internet je globalna mreža koja povezuje milijune računala. Razvio se od projekta američke vojske pod nazivom ARPAnet. Ta mreža je sredinom osamdesetih godina trebala poslužiti za brzi prijenos podataka i komunikaciju u slučaju nuklearnog napada. U trenutku kad je projekt predan sveučilišnim institucijama, mreža se velikom brzinom proširila da bi danas predstavljala najveću mrežu na svijetu (Srdić & Žezlina, 2001).

Više od 100 država je povezano i izmjenjuju podatke, vijesti i mišljenja. Prema *Internet World Stats*, 31. prosinca 2011. je bilo 2,267,233,742 korisnika Interneta širom svijeta. To je 32.7% svjetske populacije. Za razliku od internetskih servisa, koji se kontroliraju centralno, Internet je dizajniran decentralizirano. Svako računalo spojeno na Internet naziva se *host* i neovisno je (URL21).

World Wide Web (skraćeno WWW ili Web) sastoji se od svih javnih web stranica spojenih na Internet širom svijeta, uključujući i korisničke uređaje primjerice računala i mobilne telefone, kako bi pristupili sadržaju Weba. WWW je samo jedna od mnogih internetskih aplikacija i aplikacija računalnih mreža (URL22).

Web se sastoji od slijedećih tehnologija:

- HTML – Hypertext Markup Language
- HTTP – Hypertext Transfer Protocol
- Web serveri i web pretraživaći (eng. *Web browsers*)

URL je kratica za eng. *Uniform Resource Locator*. URL je oblikovan tekst kojeg koriste web preglednici, email klijenti i ostali softveri kako bi identificirali mrežni resurs na Internetu.

Mrežni resursi su dokumenti koji mogu biti jednostavne Web stranice, tekstualni dokumenti, slike ili programi. URL se sastoji od tri dijela (URL23):

- mrežnog protokola
- ime ili adresu domaćina (eng. *host*)
- lokaciju dokumenta ili izvora

Ovi dijelovi su razdvojeni posebnim znakovima na sljedeći način:

protocol :// host / location

Web stranica (eng. *website*) je mjesto (eng. *site, location*) na World Wide Webu. Svaka Web stranica sadrži početnu stranicu, koja je prvi dokument kojeg korisnik vidi kada dođe na tu stranicu. Stranica također može sadržavati dodatne dokumente (slike, audio i video zapise i dr.) (URL24).

Web portali nude jedinstvenu točku pristupa agregiranim informacijama. To znači da se na s jednog mjesta može pristupiti velikom broju informacija. Web portali nude bogatu navigacijsku strukturu. Na početnoj stranici web portala često se nalaze brojni hiperlinkovi koji vode do dodatnih sadržaja koje nudi web portal. (URL25)

2.10.2. OpenStreetMap

OpenStreetMap (OSM) (Ramm, Topf, & Chilton, 2011) projekt je pokrenuo Steve Coast u Engleskoj 2004. godine s ciljem kreiranja slobodne (eng. *free*) karte svijeta. Slobodno ne znači samo besplatno, već i slobodno od restrikcija koje ometaju produktivno korištenje podataka. Podaci se prikupljaju tako da ljudi hodaju, planinare, voze bicikle ili aute i pritom GPS uređajima prikupljaju podatke. Ti podaci se zatim pažljivo precrtavaju na računalo, dodaju se atributni podaci poput naziva ulica i učitavaju u centralnu bazu podataka. Na globalnoj razini, mnogi gradovi su već kartirani puno detaljnije nego što to nude Google, Yahoo ili Microsoft. Također, OpenStreetMap podaci su često ažurniji od tiskanih karata ili drugih internet servisa koji nude karte.

Iza svake karte stoji kolekcija podataka: ceste, šume, rijeke i svi drugi elementi prikazani na karti se učitavaju iz baze podataka u trenutku kreiranja karte. Umjetnost kartografije se sastoji uglavnom u odabiru pravog skupa podataka i odlučivanju kako će oni biti prikazani na karti – njihovi stilovi, boja, težine i sl. OpenStreetMap pruža mogućnost pristupa originalnim

podacima, što znači da svatko može biti svoj kartograf. Svatko može prema svojim potrebama odabrati podatke i značajke koji su mu bitni te sam odlučivati o izgledu karte. No ti podaci se mogu koristiti i u druge svrhe, primjerice za statističke analize ili za generiranje uputa za vožnju.

Odlučeno je da će OpenStreetMap projekt koristiti svoje formate podataka i prilagođene programe kako bi što bolje služili OSM svrsi. Vrlo je važno da podaci i programi mogu biti korišteni od strane ljudi bez prijašnjeg GIS iskustva. Također, tehnologija mora podržavati proizvoljne atribute podataka. Nemoguće je unaprijed definirati strukturu OSM podataka i očekivati da će svatko koristiti iste kategorije s jednakom razinom detaljnosti.

2.10.3. GeoNames

GeoNames je geografska baza podataka dostupna pod *creative commons* licencom. Slobodna je za preuzimanje. Sadrži preko 10 milijuna geografskih imena, sastoji se od preko 8 milijuna jedinstvenih obilježja od čega je 2.8 milijuna naseljenih mjesta i 5.5 milijuna alternativnih imena. Sva obilježja su kategorizirana u jednu od 9 klasa (Tablica 2). Svaka klasa ima podklase, kojih ukupno ima 645 (URL26).

Tablica 2. *Klase u GeoNames bazi podataka*

Oznaka klase	klasa
A	Država, regija..
H	Jezera, potoci..
L	Parkovi, područja..
P	Gradovi, naselja..
R	Rijeke, željeznice..
S	Zgrade, farme, mjesta..
T	Planine, brda, kamenje..
U	Podmorje
V	Šume, stepe..

GeoNames integrira geografske podatke kao što su nazivi gradova na raznim jezicima, nadmorsku visinu, broj stanovnika i ostale podatke iz raznih izvora. Geografska širina i dužina (eng. *latitude, longitude*) su u WGS84 koordinatnom sustavu. Korisnici mogu sami mijenjati, ispravljati i dodavati nova imena koristeći jednostavno korisničko sučenje.

Preuzeta je baza podataka za područje Republike Hrvatske. Ona sadrži nešto manje od 9600 unosa s oznakom „P“ (naseljena mjesta), koja će biti korištena kao referentna baza podataka za ovaj rad. Primjerice, Tablica 3 prikazuje GeoNames podatke za grad Osijek.

Tablica 3. *GeoNames podaci za Osijek*

Geonameid	**3193935**
Name	Osijek
asciiname	Osijek
alternatenames	Colonia Aelia Mursa, Esseg, Essegg, Essek, Eszek, Eszék, Mursa, Mursia, OSI, Osek, Osiek, Osigiek, Osijek, Osijeka, Osijekas, Osik, oshieku, xo siyekh, Οσιγιέκ, Осиек, Осијек, โอซีเยค, オシエク
latitude	45.55111
longitude	18.69389
feature class	P
feature code	PPLA
country code	HR
cc2	
admin1 code	10
admin2 code	3193934
admin3 code	
admin4 code	
population	88140
elevation	
Dem	87
timezone	Europe/Zagreb
modification date	2012-11-21

2.10.4. Državni zavod za statistiku

Sa stranica Državnog zavoda za statistiku (URL27) preuzete su granice županija u SHP formatu. One su korištene za provjeru u kojoj se županiji nalaze lokacije članaka. Kako bi to bilo moguće, oba seta podataka moraju biti u istom koordinatnom sustavu. Koordinatni sustav u kojem se nalaze granice županija je Hrvatski državni koordinatni sustava (HDKS), s razlikom što se HDKS dijeli na dvije zone, 5. i 6., dok je ovdje sve u istoj zoni. Pošto su podaci iz GeoNames baze podataka u WGS84 koordinatnom sustavu, potrebno je transformirati granice županija u taj koordinatni sustav. To je učinjeno pomoću GRASS GIS aplikacije.

Proj4 definicije početnog koordinatnog sustava, HDKS, i ciljnog, WGS84 koordinatnog sustava su:

```
PROJCS["HR_GK_1630",GEOGCS["GCS_Bessel_1841",DATUM["D_Bessel_1841",SPHEROID["Bessel_1841",6377397.155,299.1528128]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",2500000.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",16.5],PARAMETER["Scale_Factor",0.9997],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]]
```

```
GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]]
```

Tablica 4 prikazuje usporedbu parametara početnog i ciljanog koordinatnog sustava za županije Republike Hrvatske.

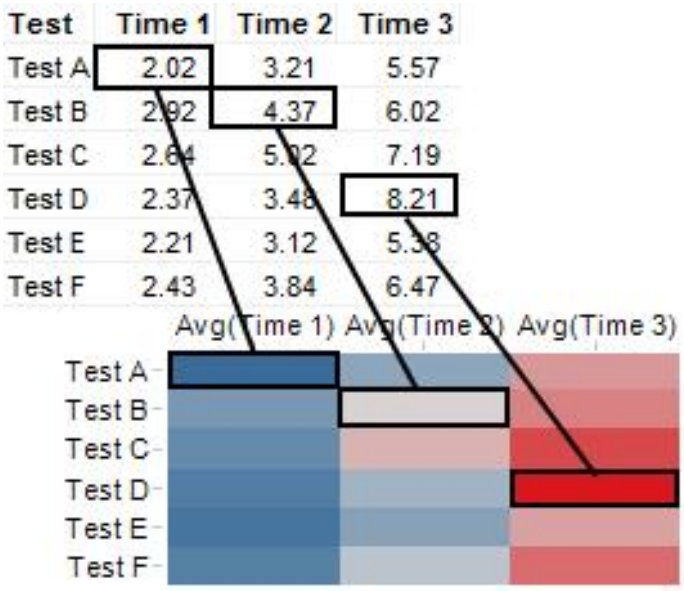
Tablica 4. *Usporedba početnog i ciljanog koordinatnog sustava*

	Gauss-Kruger	WGS84
Projekcija	Poprečna Merkatorova	
Datum	Bessel 1841	WGS84
Sferoid	"Bessel_1841" 6377397.155 299.1528128	"WGS_1984" 6378137 298.257223563
Početni meridijan	Greenwich 0.0	Greenwich 0.0
False_Easting	2500000.0	
False_Northing	0.0	
Središnji meridijan	16.5°	
Mjerilo preslikavanja	0.9997	
Mjerna jedinica	metar	stupnjevi

2.11. Karte žarišta

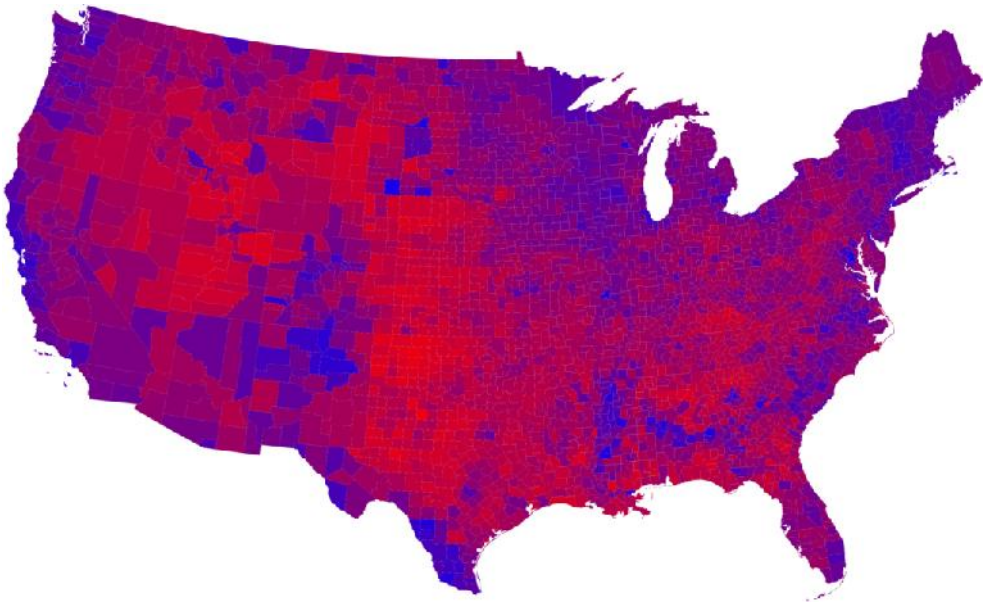
Karta žarišta (eng. *heatmap*) je dvodimenzionalna reprezentacija podataka u kojoj su vrijednosti predstavljene bojama. Jednostavna karta žarišta pruža vizualni sažetak informacija. Složenije karte žarišta omogućavaju shvaćanje kompleksnih skupova podataka (URL28).

Najjednostavniji način za shvatiti karte žarišta je zamisliti tablicu koja sadrži boje umjesto brojeva (slika 2). Primjerice, postavi se tamno plava boja za najnižu vrijednost, crvena boja za najveću vrijednost, a nijanse plave i crvene za vrijednosti između za dva ekstrema. Karte žarišta su prikladne za vizualizaciju velikih količina višedimenzionalnih podataka i mogu se koristiti za identifikaciju redova tablice koji sadrže slične vrijednosti, jer se oni prikazuju sličnom bojom. Slika prikazuje kako su vrijednosti u tablici predstavljene bojama u čelijama karte žarišta (URL29).



Slika 2. *Prikaz postupka dodjeljivanja boja vrijednostima u tablici* (URL29)

Karte žarišta imaju raširenu primjenu. Primjerice, koriste se u SAD-u za vrijeme predsjedničkih izbora (slika 3). Crveno-plava karta žarišta vrlo jasno pruža informaciju u kojoj državi je pobijedio pojedini kandidat (URL30).



Slika 3. *Karta predsjedničkih izbora SAD-a 2008. godine* (URL31)

3. Rezultati istraživanja

U nastavku objašnjen je postupak potreban za prikupljanje i obradu članaka preuzetih s četiri različita web portala. Taj postupak uključuje algoritme za obradu teksta na temelju sličnosti riječi. Također su prikazani rezultati testiranja algoritama za obradu teksta i odabir konačnog algoritma na osnovu najboljih rezultata postignutih u testovima.

Članci su prikupljeni sa sljedećih web portala:

- Istra News (http://www.istranews.in/);
- Glas Slavonije (http://www.glas-slavonije.hr/);
- Danas.hr (http://danas.net.hr/);
- Jutarnji list (http://www.jutarnji.hr/).

Obrada podataka prva tri portala trajala je od 18. svibnja 2014. do 31. kolovoza 2014., a obrada podataka četvrtog portala započela je 27. svibnja 2014.

Obrada podataka tekla je po ovim glavnim koracima:

1. Učitavanje URL-a članaka
2. Iz datoteke učitavanje URL-a već obrađenih članaka
3. Za svaki članak koji još nije obrađen:
   3.1. Izdvajanje čistog teksta članka
   3.2. „čitanje“ članka – traženje potencijalnih lokacija
   3.3. Izdvajanje koordinata iz baze podataka za prve tri potencijalne lokacije
   3.4. Traženje u kojoj se županiji nalazi pojedina lokacija
   3.5. Postoji li u toj županiji grad s istim nazivom na popisu poštanskih brojeva
   3.6. Postoji li na Google Kartama grad s istim nazivom na udaljenosti manjoj od 25 km od potencijalne lokacije
   3.7. Izdvajanje lokacije s najvećim koeficijentom
4. Zapis obrađenog članka kojem je pronađena lokacija

3.1. Tijek algoritma

Prvi korak obrade podataka je upoznavanje s HTML kodom pojedinog portala. Uzevši u obzir jedinstvenost pojedinog portala, potreban je jedinstveni pristup svakom portalu. Pomoću

Requests biblioteke izdvajaju se linkovi aktualnih članaka s početne stranice portala te se pojedinačno učitavaju.

Drugi korak je iz datoteke već obrađenih članaka učitati njihove URL adrese, kako se ne bi nepotrebno ponovno obrađivali već obrađeni podaci.

Treći korak algoritma je obrada neobrađenih članaka. Ovaj korak je podijeljen na sedam zadataka. Prvi zadatak je izdvajanje čistog teksta članka. HTML kod svakog članka sadrži poveznice na članke slične tematike, mogućosti dijeljenja članaka na Facebooku i Twitteru, a za obradu je potrebno izdvojiti čisti tekst članka. Prilikom obrade teksta članka potrebno je ukloniti interpunkcijske znakove s ciljem redukcije njihovog utjecaja na duljinu riječi te na algoritam traženja riječi i računanja sličnosti. Interpunkcijski znakovi su uklonjeni pomoću NLTK biblioteke.

Drugi zadatak je pretraživanje teksta. S obzirom da se u hrvatskom jeziku nazivi gradova pišu velikim početnim slovom, koriste se samo riječi napisane velikim početnim slovom za usporedbu s podacima u GeoNames bazi podataka. U slučaju gradova koji se sastoje od više riječi, uvjet je da prva i zadnja riječ budu napisane velikim početnim slovom.

Za potrebe ovog zadatka bilo je potrebno odabrati najprikladniji algoritam za uspoređivanje teksta s obzirom na specifičnosti zadatka – uspoređuju se kratki nizovi znakova.

3.2. Testiranje i odabir algoritma za obradu teksta

Testirani su algoritmi koje nude FuzzyWuzzy i Jellyfish biblioteke, kako bi se otkrio optimalan algoritam za ovu vrstu ulaznih podataka. Nakon testiranja spomenutih algoritama, najbolje rezultate dao je algoritam Jaro udaljenost (eng. *Jaro distance*). Algoritmi su testirani nad skupom podataka od 20 članaka (popis njihovih URL adresa je u prilogu).

Analizirana su sva četiri algoritma FuzzyWuzzy biblioteke:

- Ratio
- Partial Ratio
- Token Sort Ratio
- Token Set Ratio.

Tablica s rezultatima testiranja se nalazi u prilogu (Tablica 1). Algoritam *Ratio* pokazao se kao najbrži u ovoj biblioteci. Točno je pročitao četrnaest članaka i jednog (četrnaestog) djelomično

točno. Ovdje je uz točno mjesto Donje Orešje s istim brojem bodova odabrao i krivo mjesto Orešje. Sedmi članak je prvi koji je pogrešno pročitan. To je članak bez stvarne lokacije, no u njemu se spominje političarka Kolinda Grabar Kitarović i je njezino prvo prezime je prepoznato kao grad Grabar. U devetom članku je zbog padeža mjesto Sela dobilo prednost pred točnim mjestom Dugo Selo. U desetom članku je točno mjesto Zagreb, no u njemu se spominje i njegov gradonačelnik, pa je prezime Bandić prepoznati kao grad Bandića. U dvanaestom članku je kao mjesto Zakopa prepoznata riječ Zakona. U devetnaestom članku je, također zbog padeža, prepoznato mjesto Krka umjesto točnog Krk.

Algoritam *Partial Ratio* pokazao se kao najsporiji i najmanje točan algoritam ove biblioteke. Točno je pročitao samo sedam članaka (2, 4, 13, 15, 17, 19 i 20) i šest djelomično točno (1, 3, 5, 10, 11 i 14) što znači da je uz točna mjesta zabilježeno i neko pogrešno mjesto. Od pogrešno pročitanih članaka tu je članak broj šest, koji je članak bez prave lokacije, to on je pronašao mjesta Berak u nazivu sveučilišta Berkley i mjesto Sapci u nazivu vrste *Homo Sapiens*. U sedmom članku je u riječi Republika prepoznato mjesto Rep. U osmom članku je zbog prezimena Lucić zapisano mjesto Lucići. U devetom su članku umjesto točnog mjesta Dugo Selo zapisana dva mjesta, Dugo i Sela. U dvanaestom je članku, isto kao i prethodni algoritam, u riječi Zakona prepoznao mjesto Zakopa. U šesnaestom je članku umjesto točnog mjesta Gromačnik zapisano mjesto Gromača. U osamnaestom su članku zapisana mjesta Župa i Draga, što su u članku riječi Županijskog i ime Dragan.

Algoritmi *Token Sort Ratio* i *Token Set Ratio* dali su potpuno iste rezultate, s tim da je *Token Sort Ratio* algoritam malo brži. Uz to, imaju i jednak broj točno pročitanih članaka kao i *Ratio* algoritam. Imaju četrnaest točno pročitanih i jedan djelomično točno, četrnaesti, gdje je uz Donje Orešje zapisano i pogrešno mjesto Orešje. Isto kao i *Ratio* algoritam, ni ova dva algoritma nisu imala problema s prvih šest članaka. U sedmom članku, koji nema prave lokacije, prezime predsjednika Josipovića prepoznato je kao mjesto Josipovac. U desetom je članku prezime zagrebačkog gradonačelnika Bandića s mjestom Bandića dalo veću sličnost od točnog mjesta Zagreba. U jedanaestom je članku prezime Župančić prepoznato kao mjesto Župani. U dvanaestom je članku, kao ostali algoritmi ove biblioteke, riječ Zakona prepoznata kao mjesto Zakopa, iako grad nema pravu lokaciju. Zadnji članak koji je pogrešno pročitan je devetnaesti, gdje je zbog padeža mjesto Krka dobilo više bodova od točnog mjesta Krk.

Treba napomenuti da se vrijeme prikazano u tablici ne odnosi samo na vrijeme potrebno algoritmu za obradu teksta da izvrši zadatak, već se odnosi i na preuzimanje i obradu članaka

te ispis rezultata. Budući da je u najboljem slučaju potrebno oko 40 min za obradu 20 linkova, vrijeme je presudan faktor u obradi većeg skupa podataka.

Testirana su četiri algoritma iz Jellyfish biblioteke:

- Levenshtein Distance
- Damerau-Levenshtein Distance
- Jaro Distance
- Jaro-Winkler Distance.

Tablica s rezultatima testiranja se nalazi u prilogu (Tablica 2). Algoritmi *Levenshtein Distance* i *Damerau-Levenshtein Distance* nisu dali dobre rezultate. Uspješno su pročitali samo dva članka, drugi i trinaesti, dok je osmi članak djelomično dobro pročitan. U tom članku je uz točno mjesto Gunja izdvojeno i pogrešno mjesto Plat. U trinaestom članku se spominje dio grada Rijeke, Pećine, pa je tako iz baze izdvojeno mjesto Pećina. Iako su po nazivima vrlo slični, ipak se ovdje ne radi o istim gradovima. U ostalim člancima neke druge riječi su pokazale veću sličnost s gradovima iz GeoNames baze podataka. Primjerice, u trećem članku se spominje Puntamika, dio grada Zadra koja se kao takva ne nalazi u bazi podataka, pa je tako kao sličan izdvojeno pogrešno mjesto Punat. Slično, u petom članku se javlja već spomenuti problem gradova sličnih naziva, Kašt i Kaštel, te varijante koje ti nazivi poprimaju promjenom po padežima. U desetom članku se spominje gradonačelnik Zagreba Milan Bandić, a postoji i mjesto Bandići, te je algoritam ovdje pokazao visoko podudaranje riječi. Uz sve to, ova dva algoritma su i najsporiji algoritmi ove biblioteke.

Algoritam *Jaro Distance* se pokazao kao najtočniji i najbrži algoritam ove biblioteke. Pogrešno je pročitao četiri članaka (7, 9, 10 i 19) i djelomično točno članak broj 14 gdje je uz točno mjesto Donje Orešje izdvojeno i pogrešno mjesto Orešje. Sedmi članak je članak bez lokacije, no spominje se političarka Kolinda Grabar Kitarović. Postoji i mjesto Grabar, te je naravno to pokazalo idealno poklapanje riječi. U devetom članku je zbog padeža u kojem se pojavljuje mjesto Dugo Selo, mjesto Sela pokazalo veću sličnost od točnog mjesta. U desetom članku se uz grad Zagreb spominje i njegov gradonačelnik Milan Bandić. Zbog padeža je njegovo prezime pokazalo potpuno preklapanje s mjestom Bandića te je tako dobilo više bodova od točnog grada. Zbog padeža je pogrešno pročitan i devetnaesti članak u kojem se govori o otoku Krku, ali je kao točan grad odabrano mjesto Krka.

Algoritam *Jaro Winkler Distance* je sličan *Jaro Distance* algoritmu pa je pokazao i neke slične mane, iako ima nešto veći broj pogrešno pročitanih članaka. To su članci 3, 7, 9, 10, 11, 12 i 19 te članak 14 koji je djelomično dobro klasificiran. Kao što je vidljivo iz tablice, većina pogrešno klasificiranih podataka je ista kao i u prethodnom algoritmu, s istim pogreškama. U sedmom članku se uz Kolindu Grabar Kitarović spominje i predsjednik republike Ivo Josipović te je u ovom članku najveću sličnost pokazalo mjesto Josipovac. U jedanaestom članku se spominje Predsjednik Općinskog suda u Samoboru Darko Župančić te je njegovo prezime pokazalo veliku sličnost s mjestom Župani. U dvanaestom članku je riječ Zakona pokazala sličnost s mjestom Zakopa.

Kao i u prethodnom testiranju, vrijeme se i ovdje ne odnosi samo na algoritam za obradu teksta, već i na učitavanje i obradu članaka i ispis rezultata. No vidljivo je da se ovdje vrijeme mjeri u nekoliko minuta, što je bitno manje nego u prethodnom testu.

3.3. Nastavak tijeka algoritma

Iz testiranja je vidljivo da je odabran algoritam *Jaro udaljenosti* iz *Jellyfish* biblioteke za uspoređivanje potencijalnih lokacija iz članaka s referentnim podacima iz GeoNames baze podataka. Također, potencijalna lokacija iz članka mora zadovoljiti uvjet da joj je korijen riječi jednak korijenu naziva grada iz baze podataka. U ovom slučaju je korijen riječi dobiven tako da je gradu iz baze podataka oduzmu zadnja 2 slova. Spremaju se nazivi gradova za koje je koeficijent sličnosti dobiven usporedbom naziva grada pronađenog u članku i naziva grada iz baze podataka veći ili jednak 0.9 i koje su zadovoljile uvjet korijena riječi.

Pretpostavka je da se lokacija nekog događaja spominje pri vrhu članka. U tu svrhu članak je podijeljen na trećine prebrojavanjem riječi i zatim dijeljenjem tog broja s 0.33 za prvu trećinu i 0.66 za drugu trećinu. Ako se potencijalni grad nalazi u prvoj trećini, koeficijentu sličnosti se dodaje koeficijent 0.3. Nadalje, ukoliko se nalazi u drugoj trećini dodaje se vrijednost 0.2, a ako je u trećoj trećini onda se dodaje 0.1.

U ovom trenutku najveći mogući koeficijent vjerojatnosti je 1. Izračunat je tako da se 70% iznosa odnosi na koeficijent sličnosti dobiven pomoću algoritma *Jaro udaljenosti* i 30% na položaj riječi u članku.

Za vrijeme čitanja članka broji se koliko je puta spomenuta neka potencijalna lokacija. Za svaki put što je spomenuta dodaje se 0.025 na koeficijent vjerojatnosti.

U nekim člancima objavljenim na portalima Glas Slavonije i Jutarnji list grad o kojem se govori istaknut je tiskanim slovima na početku članka (slika 4). U takvim slučajevima nema potrebe za pretraživanjem cijelog članka. Takvi gradovi dobivaju koeficijent vjerojatnosti 5, s ciljem lakšeg izdvajanja lokacije iz članka koja je sigurno točna.


ŽUPANJA - Predsjednik HVIDRE Josip Đakić susreo se u Županji s predsjednikom Udruge HVIDRE Županja Ivom Grgićem kako bi

Slika 4. *Primjer naglašavanja lokacije članka tiskanim slovima na početku članka*

Na portalu Istra News u nekim člancima na vrhu piše naziv grada o kojem se govori u članku, ili ukoliko je mjesto radnje neko manje mjesto, naveden je veći grad u njegovoj blizini (slika 5).


Rovinj premašio dva milijuna noćenja
Objavljeno: 13.08.2014. \\ Događaji. \\ Rovinj. \\ 582 pregleda \\ 515 29

Slika 5. *Primjer pridruživanja mjesta članku*

Nakon što se pročita cijeli članak, uzimaju se prva tri potencijalna grada i poredaju silazno po veličini koeficijenta. Zatim se iz GeoNames baze podataka preuzimaju koordinate za svaki grad. Na portalu Istra News se te lokacije prvo uspoređuju s položajem grada označenim na vrhu članka (ukoliko takav postoji), koordinate su mu također preuzete iz GeoNames baze podataka. Ukoliko je potencijalni grad u blizini grada (25 km), tada se njegovom koeficijentu vjerojatnosti dodaje 1. Ukoliko nije, taj grad se odbacuje i prihvaća se grad naznačen na vrhu članka s koeficijentom vjerojatnosti 3. U ovom slučaju se dodjeljuje 3 a ne 5 jer je to približno točna lokacija. Primjerice, na vrhu članka uvijek pišu veći gradovi, dok se zapravo u članku radi o nekom manjem mjestu pokraj tog većeg grada.

Koordinate gradova su u WGS84 koordinatnom sustavu te je njihov zapis u decimalnim stupnjevima. Kako nije uobičajeno udaljenosti prikazivati u stupnjevima, za ove potrebe su koordinate gradova preračunate u Hrvatski terestrički referentni sustav (HTRS96/TM).

Treći zadatak trećeg koraka je iz GeoNames baze podataka preuzimanje koordinata za svaku potencijalnu lokaciju i njihovo zapisivanje.

Četvrti zadatak je provjera u kojoj se županiji nalazi taj grad. Ukoliko se grad ne nalazi ni u jednoj županiji, što je moguće zbog različitih izvora podataka te grad može upasti u more ili u susjednu državu, uzima se najbliža županija. Kako su županije izvorno bile u HDKS

koordinatnom sustavu, bilo ih je potrebno transformirati u WGS84 koordinatni sustav, jer su u njemu koordinate svih gradova iz GeoNames baze podataka. To je napravljeno pomoću GRASS GIS aplikacije.

Kada je određena županija, peti zadatak je s popisa poštanskih brojeva pronaći postoji li u toj županiji grad s istim imenom. Ako postoji, koeficijent vjerojatnosti potencijalnog grada povećava se za 0.5.

Šesti zadatak je provjera postoji li na Google Kartama grad s istim imenom u krugu od 25km od onog iz GeoNames baze podataka. Ukoliko postoji, još 0.5 se dodaje na koeficijent vjerojatnosti. U suprotnome, koeficijentu vjerojatnosti se oduzima 0.5. To se postiže upotrebom Google Geocodera. Riječ je o Python biblioteci koja kao ulazne podatke uzima ulicu, grad i državu, a budući da je u radu korišten format *grad, država*, što je ponekad stvaralo probleme kod pretraživanja manjih mjesta. Konačno, kada se ova procedura provede za sva 3 grada, kao konačna lokacija bira se onaj grad s najvećim koeficijentom vjerojatnosti, što je zadnji, sedmi, zadatak trećeg koraka algoritma. U slučaju da više gradova ima isti koeficijent, zapisuju se svi.

Za svaki članak se sprema i datum kada je on napisan. Kako portali nemaju jednak oblik datuma (tablica 5), potrebno ih je uskladiti. Portali Istra News i Danas.hr imaju jednak oblik datuma: *01.06.2014*. Portal Glas Slavonije ima dva oblika datuma. Primjerice, nedavno objavljeni članci imaju oblik *Objavljeno prije 6 h i 49 min*. To vrijeme se oduzima od vremena kada je članak pročitan. Ukoliko je rezultat pozitivan broj, kao vrijeme izdavanja članka sprema se datum kada je članak pročitan, ako ne onda se sprema datum od dana ranije. Datumi starijih članaka imaju oblik *1. lipnja, 2014.* U ovom slučaju je oblik mjeseca potrebno pretvoriti u brojčani oblik, i dodati 0 ako je dan predstavljen jednoznamenkastim brojem. Portal Jutarnji list također ima dva oblika datuma: za nedavno objavljene članke *Objavljeno: prije 16 min* i za starije članke *01.06.2014.* I ovdje je potrebno oblik datuma nedavno objavljenih članaka pretvoriti u oblik *01.06.2014.* kako bi svi zapisi bili jednaki, što će kasnije omogućiti jednostavnije analize dobivenih podataka.

Tablica 5. *Formati datuma na obrađenim portalima*

Istra News	**Glas Slavonije**	**Danas.hr**	**Jutarnji list**
01.06.2014.	Objavljeno prije 6 h i 49 min	01.06.2014.	Objavljeno: prije 16 min
	1. lipnja, 2014.		01.06.2014.

Ovim postupkom obrađeno je 7052 članaka objavljenih u razdoblju od 18. svibnja do 31. kolovoza. Rezultati algoritma ispisuju se za svaki portal u posebnu datoteku. Ispis se vrši u CSV (eng. *Comma-separated values*) formatu, s „ ; “ kao razdjelnikom.

http://www.jutarnji.hr/za-lijecenje-kujice-iz-gunje-fakultet-trazi-28-416-kuna/1207108/;Zagreb;2.06666666667;45.81444;15.97798;21;0.0;10000;(u’Zagreb, Croatia’,45.8150108,15.981919);0.3126;18.07.2014.

3.4. Kontrola ispravnosti algoritma

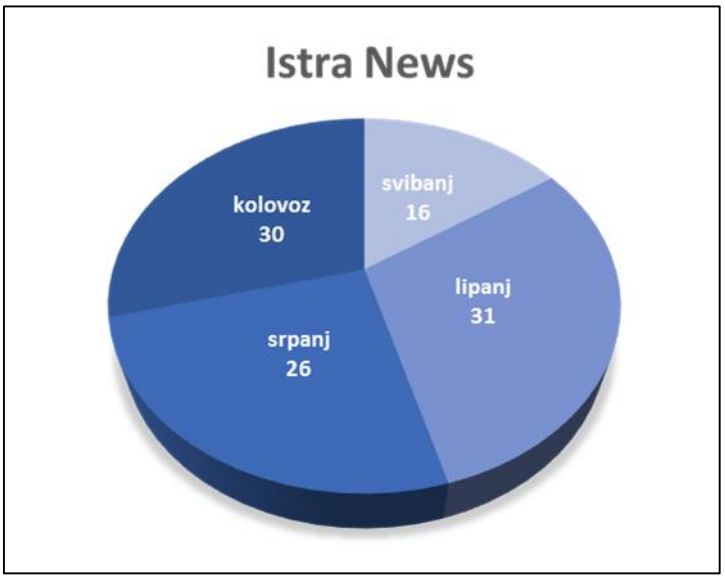
Nakon što je napisan cijeli algoritam, provjereni su njegovi rezultati na istim člancima kao i u poglavlju 3.2. Kontrolna vrijednost postavljena je osobnim čitanjem članaka te se ona odnosi na sve nazive naseljenih mjesta koji se spominju u tekstu, bez obzira na kontekst. Tablica 3 u prilogu prikazuje usporedbu rezultata *Jaro Distance* algoritma za usporedbu teksta i cijelog algoritma. Jedini članak kojeg je *Jaro Distance* algoritam obradio točno, a cijeli algoritam nije, je peti članak. Razlog leži u tome što ima više gradova s nazivom Kaštela, a onaj koji je pronađen korištenjem Google Maps servisa nije onaj koji je preuzet iz GeoNames baze podataka. Zato su ta dva grada udaljena te se koeficijent vjerojatnosti nije povećao, dok se za mjesto Marina je. Sedmi i deveti članak su u oba slučaja krivo obrađeni. Sedmi članak ima lokaciju Grabar, zbog prezimena političarke Kolinde Grabar Kitarović, a ne bi trebao uopće imati lokaciju. U devetom članku je zbog padeža mjesto Sela dobilo prednost nad mjestom Dugo Selo. Desetom i devetnaestom članku se nakon cijelog algoritma ipak izdvojila točna lokacija, unatoč tome što je neka netočna lokacija nakon algoritma za obradu teksta imala veći koeficijent vjerojatnosti.

4. Analiza rezultata

U nastavku su prikazani rezultati prikupljanja i obrade podataka novinskih članaka dostupnih preko web portala. Kao što je ranije navedeno, obrađena su četiri web portala. Neki su lokalnog karaktera i fokusirani na područje županija s povremenim člancima o ostalim događajima u državi, dok ostali pokrivaju događaje iz cijele države. Članci sa portala jutarnji.hr prikupljani su u razdoblju od 27. svibnja do 31. kolovoza, dok su s ostalih portala prikupljani od 18. svibnja do 31. kolovoza. Ukupno je obrađeno 7052 članaka kojima se pridružilo 7248 lokacija događanja.

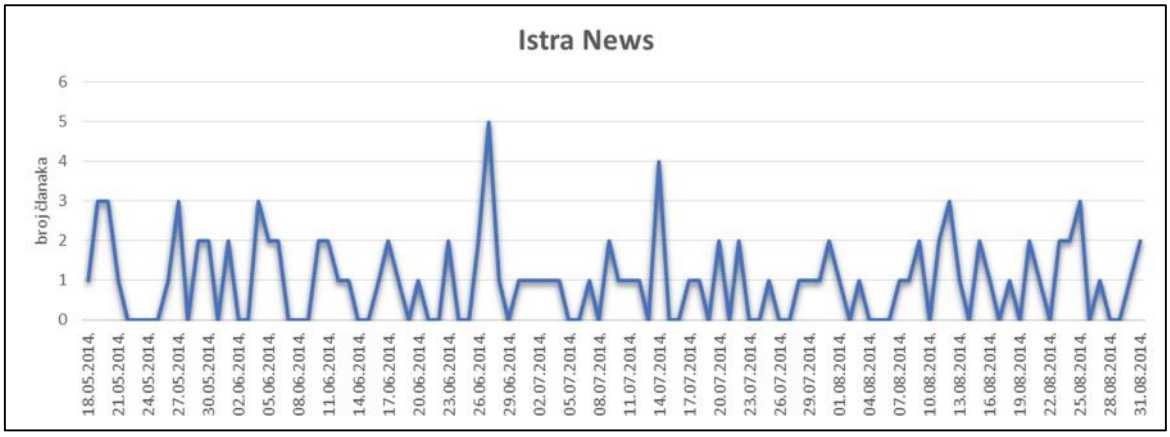
4.1. Istranews

Kao što sam naslov kaže, ovaj portal se uglavnom bavi novostima koje se događaju u Istri. U razdoblju od 18. svibnja do 1. rujna obrađeno je 103 članka kojima su pronađene lokacije, od čega 16 u svibnju, 31 u lipnju, 26 u srpnju i 30 u kolovozu (graf 1).



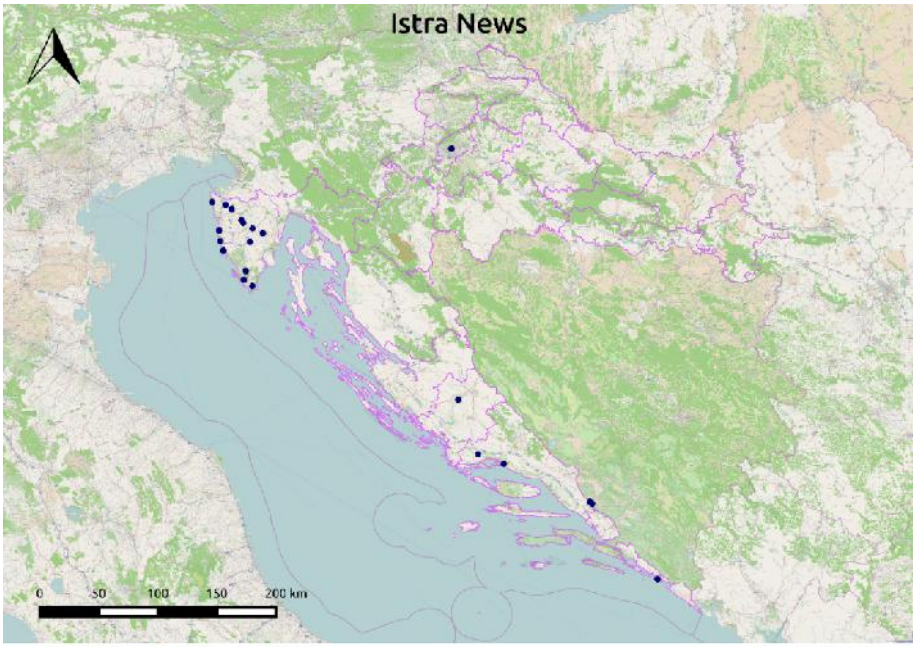
Graf 1. *Odnosi broja članaka po mjesecima za Istra News portal*

U odabranom vremenskom periodu portal je objavljivao prosječno jedan članak dnevno (graf 2). Nije primijećena smanjena aktivnost vikendom i praznicima, niti poseban događaj o kojem bi se više pisalo. Primjerice, u razdoblju od 16. srpnja do 26. srpnja održan je teniski ATP turnir u Umagu. Na turniru su nastupili neki od najboljih tenisača svijeta i najbolji hrvatski tenisači, no u tom razdoblju nije primijećena veća aktivnost na portalu. Dapače, nije napisan ni jedan članak o tom događaju. Stoga je moguće da i neki drugi događaji s područja Istre nisu obrađeni u člancima.



Graf 2. *Dnevna aktivnost portala Istra News*

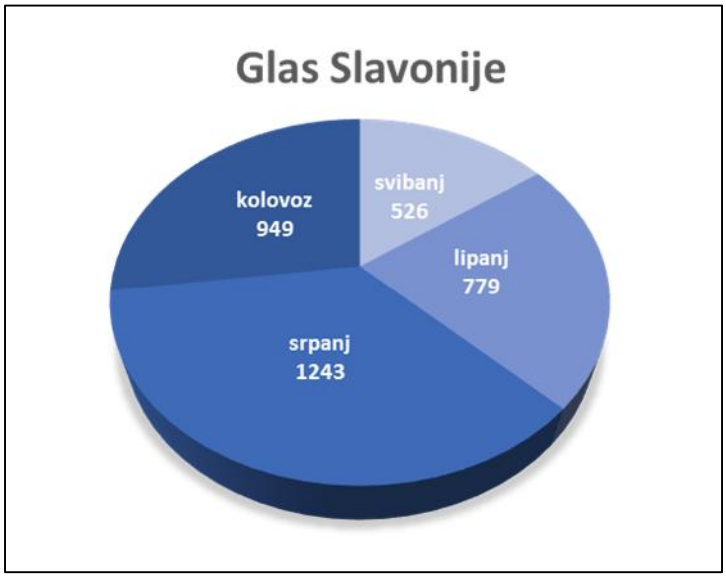
Slika 6 prikazuje geografski raspored članaka objavljenih na Istranews portalu. Vidljivo je da se velika većina njih odnosi na područje Istre. Izvan Istre locirano je samo sedam članaka. Točno je smješten po jedan članak u Zagrebu i Dubrovniku. U Istri postoji mjesto Labin, no i kod Trogira postoji mjesto Labin, tako da je članak u kojem je Labin mjesto događanja dobio koordinate Labina kod Trogira. Ostala su još četiri članka s lokacijom izvan Istre. U jednom se spominju Puljani, stanovnici grada Pule, no algoritam ih je prepoznao kao mjesto Puljane. U idućem članku je politička stranka Orah prepoznata kao mjesto Orah. Zatim je država Kamerun prepoznata kao mjesto Kamen. Na kraju je osobno ime Luka Šulić prepoznato kao mjesto Luka.



Slika 6 *Geografski prikaz članaka Istranews portala*

4.2. Glas Slavonije

U razdoblju od 18. svibnja do 1. rujna obrađen je 3371 članak, a pronađeno je 3497 lokacija članaka. To je zato jer neki članci imaju više od jedne lokacije, zbog jednakog broja bodova koje prikupe lokacije. U svibnju je prikupljeno 526 lokacija, u lipnju 779, u srpnju 1243 i u kolovozu 949 lokacija (graf 3).



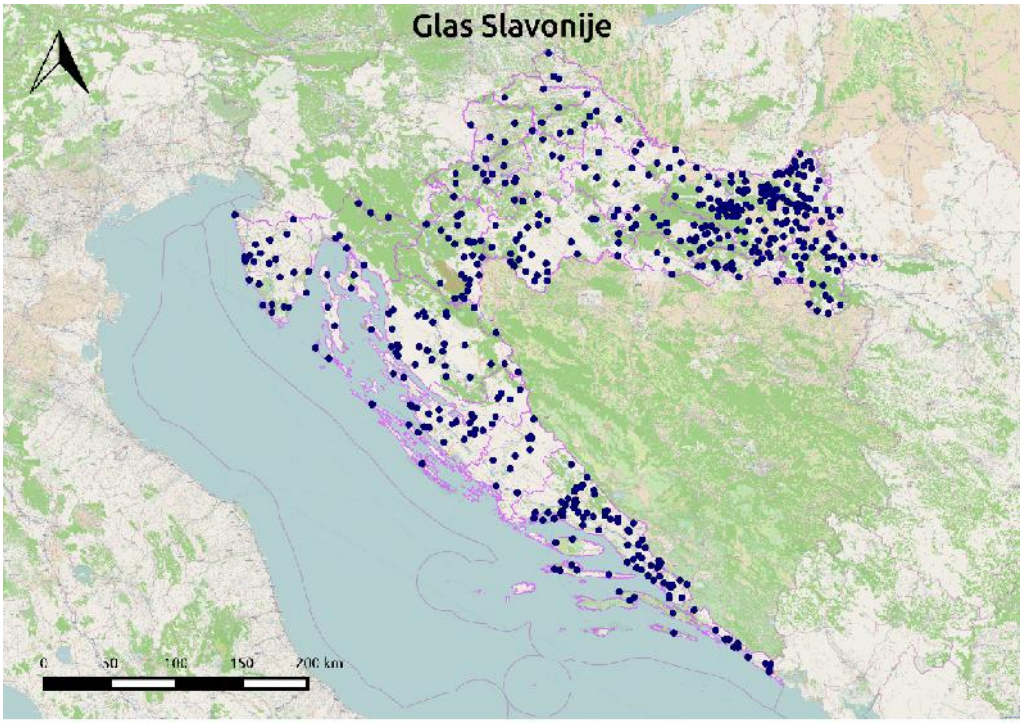
Graf 3. *Odnosi broja članaka po mjesecima za portal Glas Slavonije*

Kao što je prikazano na grafu 4, manje je objavljenih članaka nedjeljom te praznicima. Također, u periodu od 25. do 30. lipnja nije se moglo pristupiti portalu, tako da u tom periodu nema objavljenih članaka. Pretpostavka je da su njihovi novinari te dane prikupljali novosti i pisali članke te ih objavili u idućih nekoliko dana, jer se tada vidi neobično visoka aktivnost.

Graf 4. *Dnevna aktivnost portala Glas Slavonije*

Portal Glas Slavonije nije orijentiran samo na područje Slavonije, već prati i ostale novosti iz Hrvatske i svijeta (slika 7). Ipak, vidljivo je da su članci brojniji i gušći na području Slavonije.

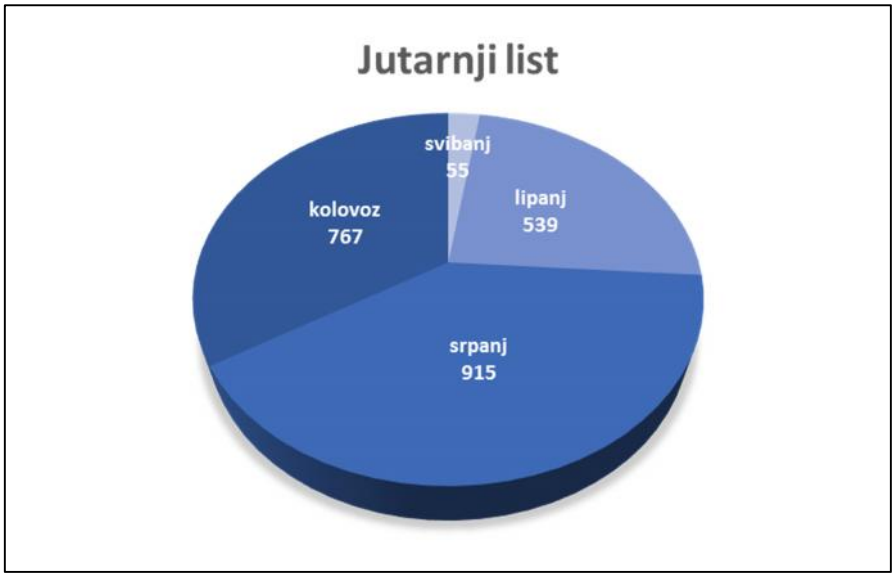
Članci nisu koncentrirani samo na veće gradove, već se bave i događajima i novostima u manjim mjestima. Vidljiva je nešto gušća aktivnost u Srednjoj i Južnoj Dalmaciji. Razlog tome je što u Dalmaciji postoji mnogo mjesta s nazivima poput Marasi, Nikolići, Milanovići, Perković, Markulin i Dragičević, što su i relativno česta hrvatska prezimena. Algoritam ne prepoznaje kontekst te logično ta prezimena imaju veliku sličnost s gradovima iz GeoNames baze podataka. Zbog toga se kao lokacija određuje to mjesto umjesto onog pravog mjesta događanja, koje se možda spominje negdje drugdje u članku.



Slika 7. *Geografski prikaz članaka portala Glas Slavonije*

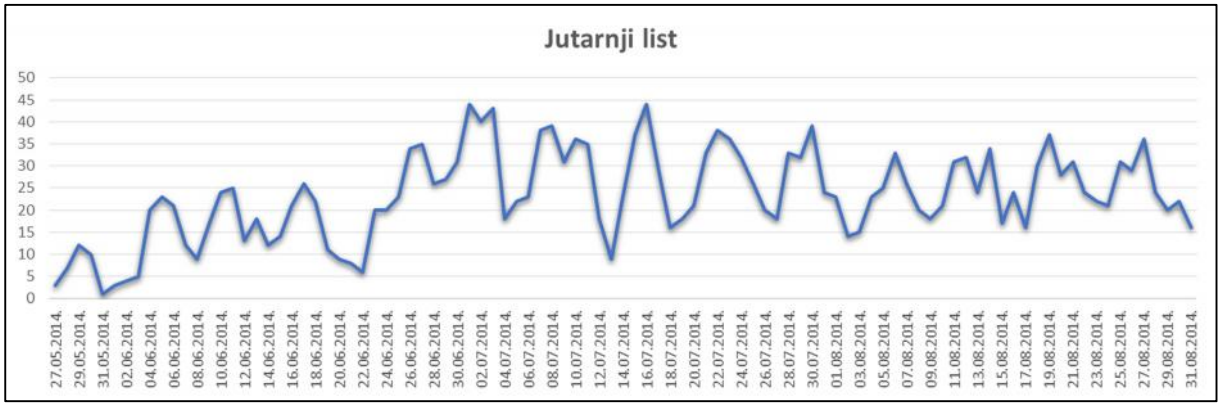
4.3. Jutarnji list

Članci s ovog portala obrađivani su od 27. svibnja do 1. rujna. U tom razdoblju obrađeno je 2246 članaka i prikupljeno je 2276 lokacija članaka. U svibnju je prikupljeno 55 lokacija, u lipnju 539, u srpnju 915 i u kolovozu 767 lokacija (graf 5).



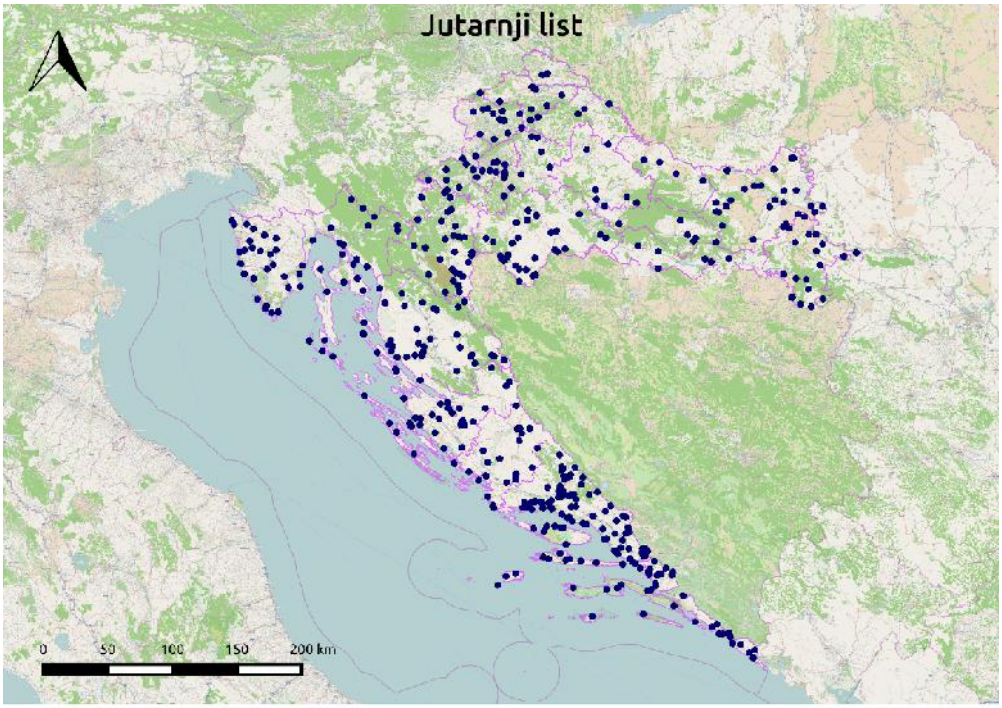
Graf 5. *Odnosi broja članaka po mjesecima za portal Jutarnji list*

Kao što je vidljivo na grafu 6, nešto je manje članaka objavljenih vikendom i praznicima, no i te dane je vidljiva aktivnost na portalu.



Graf 6. *Dnevna aktivnost portala Jutarnji list*

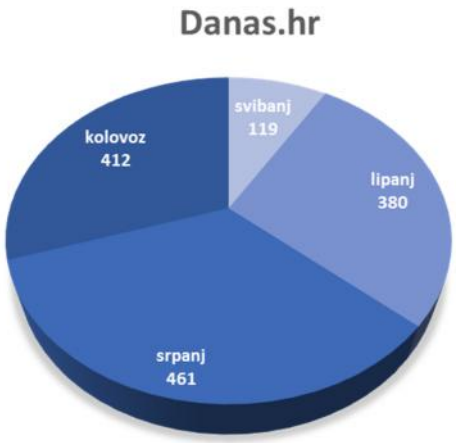
Kako je vidljivo na slici 8, portal Jutarnji list bavi se događajima u cijeloj Hrvatskoj. Kao i kod prethodnog portala, vidljiva je veća gustoća članaka u Srednjoj i Južnoj Dalmaciji, iz razloga što tamo postoji puno manjih mjesta s jednakim nazivima kao neka hrvatska prezimena, a kako algoritam ne prepoznaje kontekst te ne zna da se radi o prezimenima, uspješno ih prepoznaje kao nazive naseljenih mjesta.



Slika 8. *Geografski prikaz članaka portala Jutarnji list*

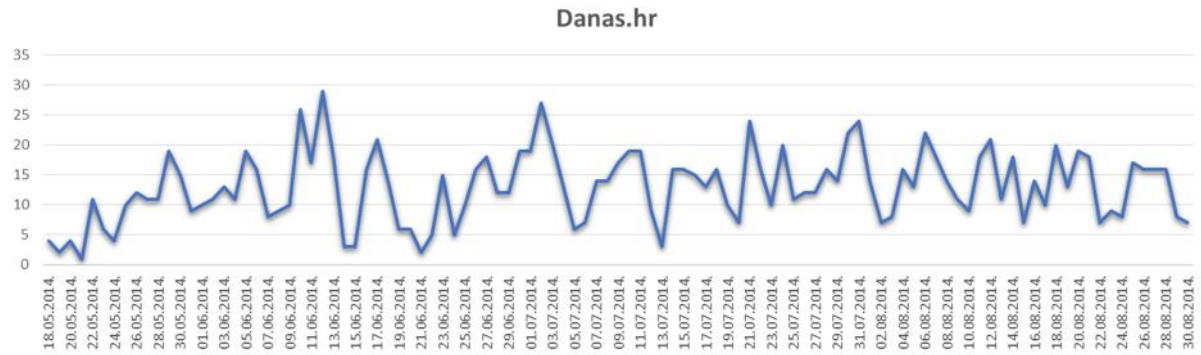
4.4. Danas.hr

Članci ovog portala preuzimani su i obrađivani u razdoblju od 18. svibnja do 31. kolovoza. U tom razdoblju obrađeno je ukupno 1332 članka, no neki članci imaju po dvije ili više lokacija s jednakim koeficijentom vjerojatnosti te su sve one zabilježene. Tako ima ukupno 1372 lokacije članaka, 119 u svibnju, 380 u lipnju, 461 u srpnju i 412 u kolovozu (graf 7).



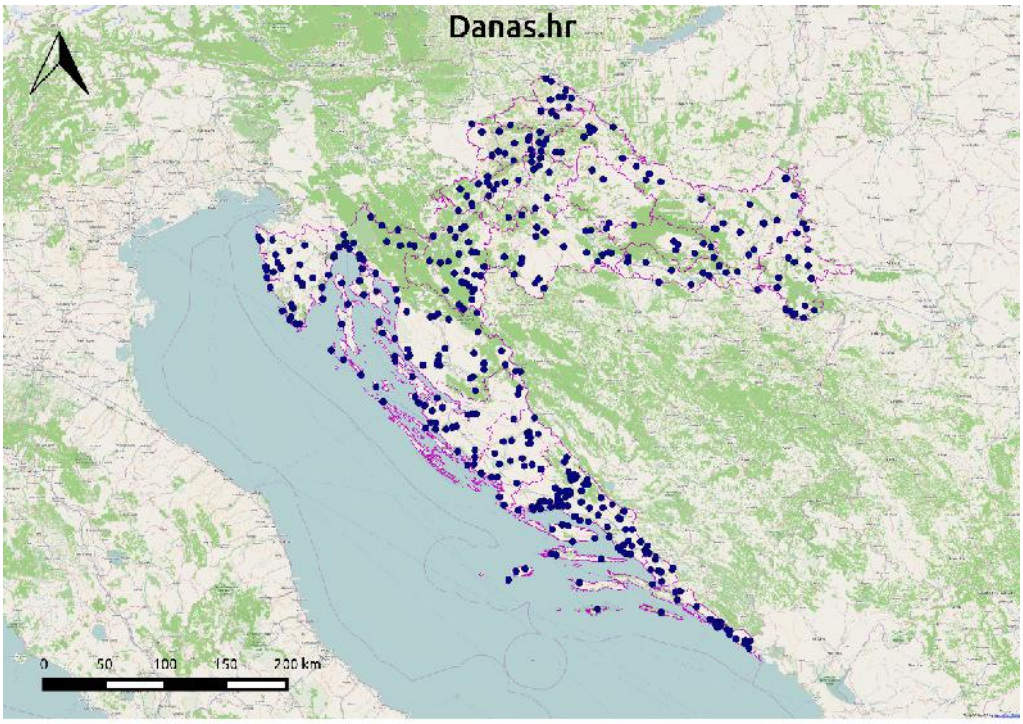
Graf 7. *Odnosi broja članaka po mjesecima za portal Danas.hr*

Kako je prikazano na donjem grafu, vidljive su oscilacije u količini objavljenih članaka po danu, no njih nije moguće povezati s vikendima i praznicima (graf 9).



Graf 8. *Dnevna aktivnost portala Danas.hr*

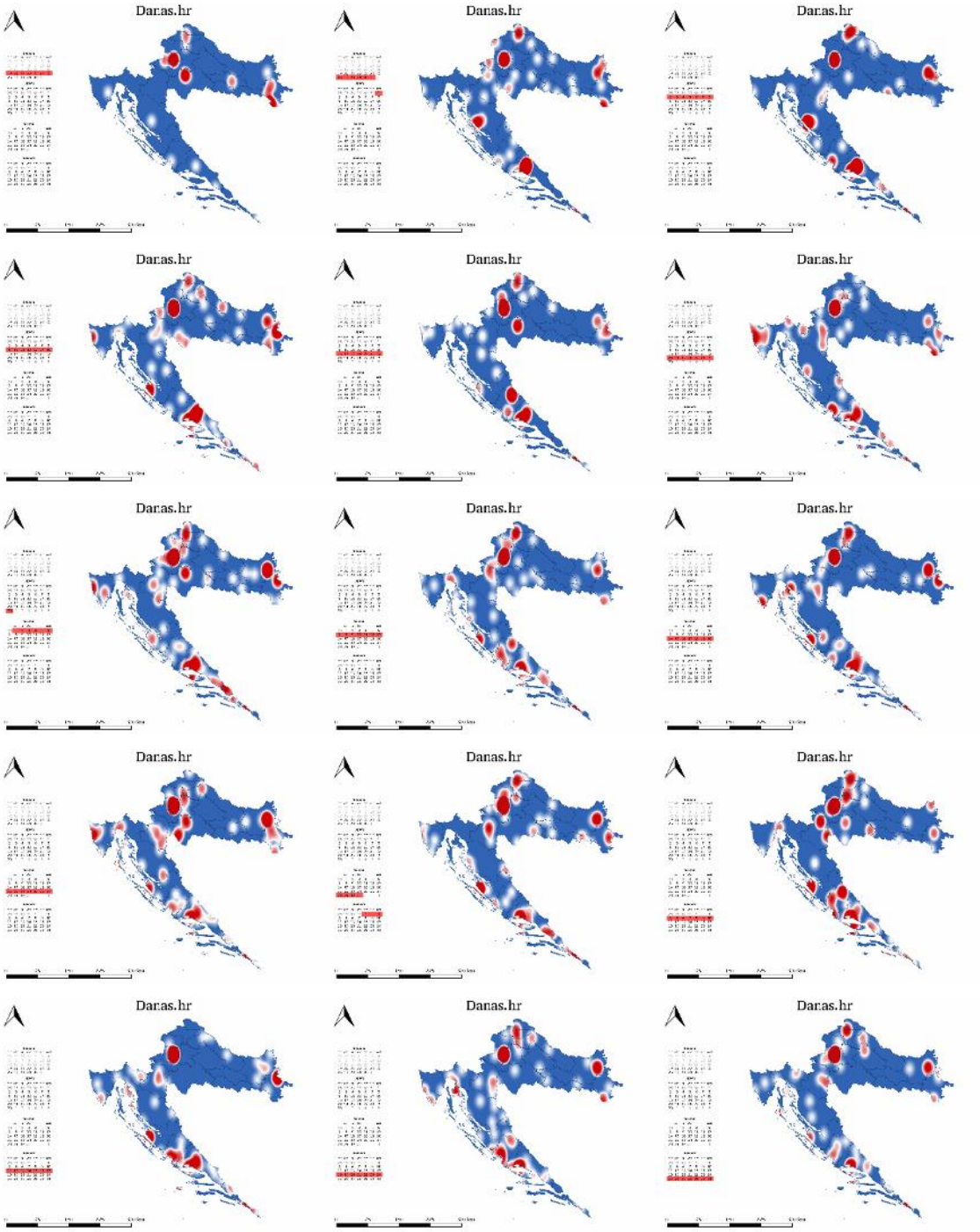
Slika 9 prikazuje sve prikupljene lokacije ovog portala u spomenutom razdoblju. Vidljivo je da se portal bavi novostima s cijelog područja Republike Hrvatske. Na području Splita i Vinkovaca su nešto gušće lokacije članaka. Kao i kod prethodnih portala, to se može povezati s nazivima mjesta koja su jednaka ili vrlo slična često spominjanim prezimenima.



Slika 9. *Geografski prikaz članaka portala Danas.hr*

4.5. Vizualizacija rezultata

Za vizualizaciju prikupljenih članaka napravljene su Karte žarišta. Karte žarišta su karte na kojima će biti lako vidljiva područja na kojima je objavljeno najviše članaka. Svaka karta prikazuje razdoblje od jednog tjedna. Karte žarišta su napravljene pomoću QGIS aplikacije.



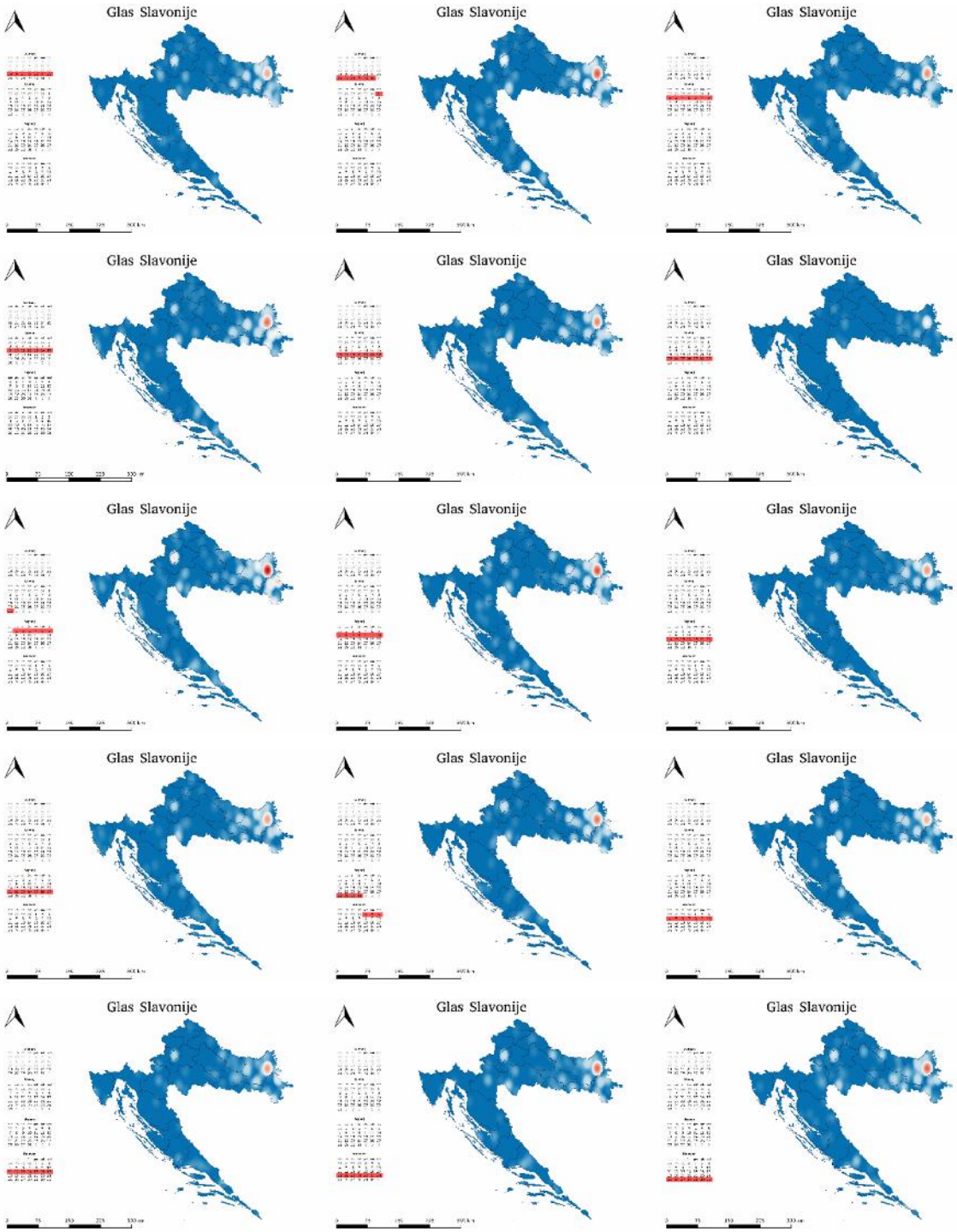
Slika 10. *Karte žarišta za portal Danas.hr na tjednoj bazi*

Iz ovih slika (slika 10) je vidljivo da su članci koncentrirani uglavnom na područja Zagreba i Splita i nešto manje Osijeka. Na prvoj slici (za tjedan od 19. svibnja do 25. svibnja) do izražaja dolazi područje Gunje i Rajeva Sela. U tom razdoblju su na tom području bile poplave, te je logično da je najviše novih vijesti dolazilo od tamo, dok se o drugim dijelovima nije previše pisalo.

Za razliku od prethodnog portala, portal Istra News pokazuje vrlo malu aktivnost na tjednoj bazi (slika 11). Najviše objavljenih članaka govori o događajima u Rovinju i Puli. Već je spomenut primjer ATP turnira u Umagu, koji je mogao izazvati veliku aktivnost s obzirom na poznate domaće i svjetske tenisače koji su sudjelovali, no o toj temi nije napisan ni jedan članak. Slično, u periodu od 26. do 30. srpnja u Motovunu je održan poznati Motovun Film Festival, no u tom periodu nije napisan ni jedan članak čija bi lokacija bila Motovun.

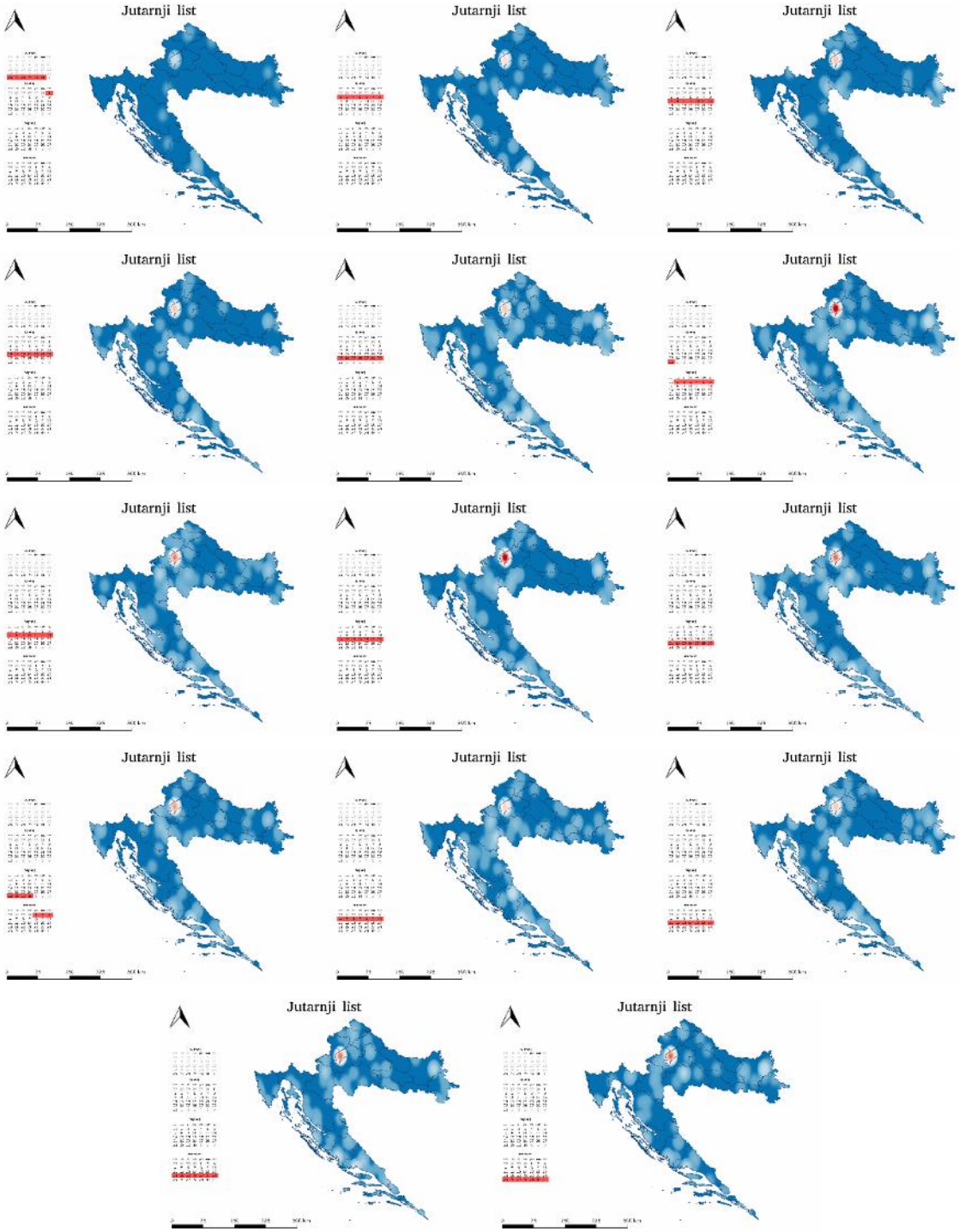
Slika 11. *Karte žarišta za portal Istra News na tjednoj bazi*

Glas Slavonije (slika 12) je regionalni portal te tako najviše objavljenih članaka govori o tom području, odnosno Slavoniji. Osijek je jedini naglašeni grad, no zastupljeno je cijelo područje. Za razliku od Istra News portala, Glas Slavonije prati i događaje u ostatku zemlje, posebno u Zagrebu, a nešto manje i ostale veće gradove.



Slika 12. *Karte žarišta za portal Glas Slavonije na tjednoj bazi*

Jutarnji list (slika 13) je još jedan portal koji se bavi događajima u cijeloj zemlji. Najizraženiji grad je Zagreb. Od ostalih mjesta najviše prati veće gradove poput Splita, Osijeka, Dubrovnika, Karlovca itd.



Slika 13. *Karte žarišta za portal Jutarnji list na tjednoj bazi*

5. Rasprava

Tokom pisanja rada pojavili su se određeni problemi. Jedan od njih je što se podaci iz različiti izvora ne podudaraju potpuno. Tako je moguće da grad s koordinatama iz GeoNames baze podataka nije unutar granica Hrvatske prema podacima o granicama županija. To se dogodilo primjerice za grad Poreč, koji je upao u more, te je bilo potrebno u algoritam dodati da ukoliko se grad ne nalazi ni u jednoj županiji, da se priroda najbližoj.

Ostali problemi bili su vezani za obradu teksta. Jedan od tih je što su imena nekih manjih naseljenih mjesta ista ili vrlo slična imenima ili prezimenima osoba, primjerice gradonačelnik Zagreba Milan Bandić se relativno često pojavljuje u člancima, te se njega identificira kao naselje Milan. Isto tako, političare Linića i Milanovića algoritam prepoznaje kao naselja Linići i Milanovići. Taj problem bi se mogao riješiti tako što će se algoritam proširiti te ukoliko se uz prezime spominje i osobno ime ili titula, tada se ni to prezime neće uzimati u obzir kao potencijalna lokacija članka.

Drugi problem na koji se nailazi je više naselja s jednakim imenima. Primjerice, postoji više naselja Kaštel. Algoritam ne razumije kontekst članka, te ne može zaključiti o kojem se Kaštelu radi. Također, u Hrvatskoj postoji i više mjesta s nazivom Rijeka, što onemogućuje pronalaženje pravog grada. No, nije problem samo u jednakim nazivima gradova u Hrvatskoj, već i u istim ili vrlo sličnim nazivima hrvatskih gradova s gradovima i državama u svijetu. Tako u Hrvatskoj postoji mjesto Kijevo, koje je vrlo slično gradu Kijev, glavnom gradu Ukrajine. Isto tako, u hrvatskoj postoji i mjesto Kosovo, tako da ponekad vijesti iz svijeta dobiju koordinate u Hrvatskoj.

Još jedan problem su hrvatski dvoslovi (dž, lj i nj). To su slova koja se sastoje od dva znaka, a ne dva slova koja stoje jedno kraj drugog. Problem se javlja jer ih nitko ne piše kao jedno slovo, već kao dva. Na primjeru Županje, algoritam rastavi nj (zato jer to može napraviti s obzirom da su dva slova) te zatim Županju uspoređuje s mjestom Župani.

Sljedeći korak u unapređenju algoritma bio bi strojno učenje. To bi bio polu-automatski sustav koji bi iz svakog obrađenog članka na osnovu ljudskog iskustva i intervencije „naučio“ nešto te sa svakim člankom kvalitetnije pronalazio lokacije. To bi u konačnici rezultiralo sve manjom i rjeđom ljudskom intervencijom.

Algoritam se također može unaprijediti tako što se ne bi pretraživali članci svih tematika, već bi se kroz vrijeme i prostor mogao pratiti određeni događaj – putovanja određene osobe, koncerti, predizborne kampanje, sportski događaji i dr.

6. Zaključak

U ovom diplomskom radu uspostavljena je metodologija automatizirane obrade prirodnog teksta s ciljem izdvajanja geografskih lokacija. Za izvore podataka odabrana su četiri web portala i baza podataka geografskih imena GeoNames. Web portali predstavljaju izvor tekstualnih informacija različitih oblika i dobru polaznu osnovu za semantičko obogaćivanje teksta prostornim informacijama metoda procesiranja prirodnog jezika. Iz baze podataka GeoNames izdvojena su i korištena jedino imena naseljenih mjesta i njihove koordinate. Ova baza podataka se pokazala kao pouzdan i kvalitetan izvor podataka.

Tijekom same obrade podataka korišten je programski jezik Python s brojnim dodacima. Riječ je o velikom broju alata za obradu teksta. Nakon testiranja Jaro Distance algoritam se pokazao kao najbolji za usporedbu naziva naseljenih mjesta.

Pri vizualizaciji rezultata koristile su se aplikacije otvorenog koda koje su uspješno prikazale područja koja pojedini Web portali pokrivaju pomoću karata žarišta.

Popis literature

Bird, S., Loper, E., & Klein, E. (2009). *Natural Language Processing with Python.* O'Reilly Media Inc. Preuzeto 29. svibnja 2014. iz http://www.nltk.org/book/

Downey, A. B. (2013). *Think Python* (Version 2.0.12 izd.). Dohvaćeno iz http://www.greenteapress.com/thinkpython/thinkpython.pdf

Liu, X., Wei, F., Zhang, S., & Zhou, M. (siječanj 2013). Named Entity Recognition for Tweets. *ACM Transactions on Intelligent Systems and Technology, 4*.

Nadeau, D., & Sekine, S. (1. siječnja 2007). A survey of named entity recognition and classification.

Ramm, F., Topf, J., & Chilton, S. (2011). *OpenStreetMap Using and Enhancing the Free Map of the World.* Cambridge: UIT Cambridge Ltd.

Srdić, I., & Žezlina, B. (2001). *Informatika 2.* Zagreb, Hrvatska: Profil International.

Stopper, R., Sieber, R., & Schnabel, O. (7. kolovoza 2008). Introduction to Multimedia Cartography. Dohvaćeno iz http://www.e-cartouche.ch/copyright.php

Vockner, B., & Mittlböck, M. (17. ožujka 2014). Geo-Enrichment and Semantic Enhancement of Metadata Sets to Augment Discovery in Geoportals. Dohvaćeno iz http://www.mdpi.com/2220-9964/3/1/345/htm

URL1: Youtube - Geospatial Revolution / Episode One, https://www.youtube.com/watch?v=poMGRbfgp38 (11.9.2014.)

URL2: IBM - Using semantic enrichment to enhance big data solutions, http://www-01.ibm.com/software/ebusiness/jstart/semantic/ (4.6.2014.)

URL3: The Levenshtein-Algorithm, http://www.levenshtein.net/ (8.7.2014.)

URL4: Levenshtein Distance and the Triangle Inequality, http://richardminerich.com/2012/09/levenshtein-distance-and-the-triangle-inequality/ (9.7.2014.)

URL5: Record Linkage Algorithms in F# – Jaro-Winkler Distance (Part 1), http://richardminerich.com/2011/09/record-linkage-algorithms-in-f-jaro-winkler-distance-part-1/ (28.5.2014.)

URL6: Record Linkage Algorithms in F# – Jaro-Winkler Distance (Part 2), http://richardminerich.com/2011/09/record-linkage-algorithms-in-f-%E2%80%93-jaro-winkler-distance-part-2/ (29.5.2014.)

URL7: The Soundex Algorithm, http://www.blackwasp.co.uk/Soundex.aspx (7.9.2014.)

URL8: Python Software Foundation, https://www.python.org/ (24.5.2014.)

URL9: Requests, http://docs.python-requests.org/en/latest/ (10.5.2014.)

URL10: Beautifulsoup, http://www.crummy.com/software/BeautifulSoup/ (12.5.2014.)

URL11: Record Linkage Algorithms in F# – Jaro-Winkler Distance (Part 2), http://streamhacker.com/2011/10/31/fuzzy-string-matching-python/ (29.5.2014.)

URL12: FuzzyWuzzy: Fuzzy String Matching in Python, http://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/ (23.5.2014.)

URL13: Python Package Index, Jellyfish 0.2.2, https://pypi.python.org/pypi/jellyfish/0.2.2 (13.6.2014.)

URL14: GeoPy's documentation, http://geopy.readthedocs.org/en/latest/#data (3.6.2014.)

URL15: HTML Introduction, http://www.w3schools.com/html/html_intro.asp (30.5.2014.)

URL16: Documentation for QGIS 2.2, http://docs.qgis.org/2.2/en/docs/user_manual/preamble/foreword.html (23.5.2014.)

URL17: GRASS GIS, http://grass.osgeo.org/documentation/general-overview/ (28.6.2014.)

URL18: Inkscape Overview, http://www.inkscape.org/en/about/ (10.9.2014.)

URL19: About W3C, http://www.w3.org/Consortium/ (11.9.2014.)

URL20: OpenShot Video Editor, http://www.openshot.org/ (3.9.2014.)

URL21: Webopedia – Internet, http://www.webopedia.com/TERM/I/Internet.html (23.8.2014.)

URL22: About Technology - WWW - World Wide Web, http://compnetworking.about.com/cs/worldwideweb/g/bldef_www.htm (21.8.2014.)

URL23: About Technology – URL, http://compnetworking.about.com/od/internetaccessbestuses/g/bldef-url.htm (21.8.2014.)

URL24: Webopedia – Web site, http://www.webopedia.com/TERM/W/web_site.html (22.8.2014.)

URL25: About Technology – Web portal, http://compnetworking.about.com/od/internetaccessbestuses/l/aa011900a.htm (23.8.2014.)

URL26: About GeoNames, http://www.geonames.org/about.html (17.5.2014.)

URL27: Državni zavod za statistiku, http://www.dzs.hr/ (15.6.2014.)

URL28: http://searchbusinessanalytics.techtarget.com/definition/heat-map (31.8.2014.)

URL29: What is a Heat Map?, http://custom-analytics.thomsonreuterslifesciences.com/SpotfireWeb/Help/dxpwebclient/heat_what_is_a_heat_map.htm (31.8.2014.)

URL30: SearchBusinessAnalytics – heat map, http://searchbusinessanalytics.techtarget.com/definition/heat-map (31.8.2014.)

URL31: Maps of the 2008 US presidential election results, http://www.dashboardinsight.com/articles/new-concepts-in-business-intelligence/maps-of-the-2008-us-presidential-election-results.aspx (31.8.2014.)

Popis slika

Popis grafova

Popis tablica

Prilozi

Popis članaka na kojima su testirani algoritmi:

1. http://danas.net.hr/crna-kronika/policija-upozorava-pazite-na-stanove-sezona-je-godisnjih-odmora
2. http://danas.net.hr/hrvatska/izvucen-joker-broj-tezak-gotovo-tri-milijuna-kuna
3. http://danas.net.hr/crna-kronika/nocna-drama-spaseni-otac-i-sin-4-s-gumenjaka-bez-goriva
4. http://danas.net.hr/novac/sedmi-weekend-media-festival-propituje-nove-modele-komunikacije
5. http://danas.net.hr/crna-kronika/policija-na-oprezu-zbog-moguce-osvete-nakon-ubojstva-20-godisnjaka
6. http://danas.net.hr/znanost/ljudi-koji-nisu-ljudske-vrste-nose-posebne-gene
7. http://danas.net.hr/hrvatska/operacija-barbika-tko-stoji-iza-povjerljivog-dokumenta-na-13-stranica
8. http://danas.net.hr/hrvatska/stanovnici-gunje-zivjet-ce-u-kontejnerima
9. http://danas.net.hr/crna-kronika/maturantima-busa-ukrali-100000-kuna
10. http://danas.net.hr/hrvatska/nista-od-naplate-parkiranja-u-cijelom-zagrebu
11. http://danas.net.hr/hrvatska/sudac-cistacici-dao-otkaz-jer-se-pokusala-ubiti
12. http://danas.net.hr/hrvatska/po-vladinom-prijedlogu-poslodavci-ce-moci-radnike-posudjivati-drugima
13. http://danas.net.hr/hrvatska/ovo-ljeto-rijecanima-ce-trebati-jako-cvrsti-zivci
14. http://danas.net.hr/crna-kronika/umrla-nakon-sto-ju-je-54-godisnjakinja-pregazila-traktorom
15. http://danas.net.hr/crna-kronika/krao-samo-bogate-pa-dobio-4-godine-zatvora
16. http://danas.net.hr/crna-kronika/strasna-tragedija-autom-naletio-na-6-godisnjakinju-i-usmrtio-je
17. http://danas.net.hr/crna-kronika/vandali-demolirali-desetak-automobila-u-centru-medju-njima-i-skupocjeni-jaguar
18. http://danas.net.hr/crna-kronika/monstrum-paravinja-pobjesnio-zbog-odluke-suda
19. http://danas.net.hr/crna-kronika/infarkt-na-gumenjaku-bio-je-koban
20. http://danas.net.hr/crna-kronika/nakon-nekoliko-dana-potrage-pronadjeno-njegovo-mrtvo-tijelo

Tablica 1. Usporedba algoritama FuzzyWuzzy biblioteke

Članak br.	Ratio	Partial Ratio	Token Sort Ratio	Token Set Ratio	Točan grad
1	Split 0.937	Split 1.0 *Paz 1.0*	Split 0.937	Split 0.937	Split
2	Zagreb 0.944	Zagreb 1.0	Zagreb 0.944	Zagreb 0.944	Zagreb
3	Rijeka 1.0	Rijeka 1.0 *Punta 1.0*	Rijeka 1.0	Rijeka 1.0	Diklo, Puntamika, Zadar, (Rijeka)
4	Rovinj 1.0	Rovinj 1.0	Rovinj 1.0	Rovinj 1.0	Rovinj
5	Kaštel 0.951	Kaštel 1.0 *Kašt 1.0*	Kaštel 0.937	Kaštel 0.937	Kaštel
6	-	*Berak 0.86* *Sapci 0.86*	-	-	-
7	*Grabar 0.9*	*Rep 1.0*	*Josipovac 0.923*	*Josipovac 0.923*	-
8	Gunja 0.86	*Lucići 0.895*	Gunja 0.86	Gunja 0.86	Gunja, Strošinci
9	*Sela 1.0*	*Sela 1.0* *Dugo 1.0*	Sela 1.0	Sela 1.0	Dugo Selo
10	*Bandića 1.0*	Zagreb 1.0 *Bandića 1.0*	*Bandića 1.0*	*Bandića 1.0*	Zagreb
11	Sveti Ivan 0.9 Zagreb 0.9	*Župa 1.0* Samobor 1.0	*Župani 1.0*	*Župani 1.0*	Samobor, Zagreb, (Sveti Ivan)
12	*Zakopa 0.881*	*Zakopa 0.881*	*Zakopa 0.881*	*Zakopa 0.881*	-
13	Pećina 0.916	Pećina 1.0	Pećina 0.881 Rijeka 0.881	Pećina 0.881 Rijeka 0.881	Rijeka, Pećine (dio Rijeke)
14	*Orešje 0.9* Donje Orešje 0.9	*Orešje 0.9* Donje Orešje 0.9	*Orešje 0.9* Donje Orešje 0.9	*Orešje 0.9* Donje Oešje 0.9	Donje Orešje Sveti Ivan Zelina
15	Zagreb 1.0	Zagreb 1.0	Zagreb 1.0	Zagreb 1.0	Zagreb
16	Gromačnik 0.965	*Gromača 0.909*	Gromačnik 0.958	Gromačnik 0.958	Slavonski Brod, Gromačnik
17	Dubrovnik 0.865	Dubrovnik 0.9	Dubrovnik 0.865	Dubrovnik 0.865	Dubrovnik
18	Šibenik 0.958	*Župa 1.0* *Draga 1.0*	Šibenik 0.944	Šibenik 0.944	Šibenik
19	*Krka 1.0*	Cres 1.0	*Krka 1.0*	*Krka 1.0*	Cres, Krk
20	Ključ 1.0 Lučko 1.0	Luč 1.0 Lučko 1.0	Lučko 1.0	Lučko 1.0	Lučko, Ključ, Tounj, Zagreb, Mrežnica (rijeka), Ogulin
Vrijeme	20.25 min	63.1 min	42.77 min	53.54 min	
Σ	14.5/20	9/20	14.5/20	14.5/20	

Tablica 2. Usporedba algoritama Jellyfish biblioteke

Članak br.	Levenshtein Distance	Damerau Levenshtein Distance	Jaro Distance	Jaro Winkler Distance	Točan grad
1	*Pazin 0.986*	*Pazin 0.986*	Split 0.961	Split 0.976	Split
2	Zagreb 0.993	Zagreb 0.993	Zagreb 0.966	Zagreb 0.98	Zagreb
3	*Preko 0.986*	*Preko 0.986*	Rijeka 1.0	Rijeka 1.0	Diklo, Puntamika, Zadar, (Rijeka)
4	*Medak 0.986*	*Medak 0.986*	Rovinj 1.0	Rovinj 1.0	Rovinj
5	*Kašt 0.979 Policer 0.979*	*Kašt 0.979 Policer 0.979*	Kaštel 0.970	Kaštel 0.982	Kaštel
6	*Ljuta 0.986 Nova 0.986*	*Ljuta 0.986 Nova 0.986*	-	-	-
7	*Barbiri 0.986*	*Barbiri 0.986*	*Grabar 0.9*	*Josipovac 0.946*	-
8	Gunja 0.993 *Plat 0.993*	Gunja 0.993 *Plat 0.993*	Strošinci 0.834	Gunja 0.944	Gunja, Strošinci
9	*Dugo 0.993*	*Dugo 0.993*	*Sela 1.0*	*Sela 1.0*	Dugo Selo
10	*Bandići 0.993*	*Bandići 0.993*	*Bandića 1.0*	*Bandića 1.0*	Zagreb
11	*Suza 0.986*	*Suza 0.986*	Sveti Ivan 0.9 Zagreb 0.9	*Župani 0.949*	Samobor, Zagreb, (Sveti Ivan)
12	*Lipe 0.972 Lipa 0.972*	*Lipe 0.972 Lipa 0.972*	-	*Zakopa 0.953*	-
13	Pećina 0.986	Pećina 0.986	Pećina 0.948	Pećina 0.968	Rijeka, Pećine (dio Rijeke)
14	*Umol 0.986*	*Umol 0.986*	Donje Orešje 0.9 *Orešje 0.9*	*Orešje 0.9* Donje Orešje 0.9	Donje Orešje Sveti Ivan Zelina
15	*Kras 0.993*	*Kras 0.993*	Zagreb 1.0	Zagreb 1.0	Zagreb
16	*Strn 0.972*	*Strn 0.972*	Gromačnik 0.978	Gromačnik 0.987	Slavonski Brod, Gromačnik
17	*Ruda 0.965 Rupa 0.965*	*Ruda 0.965 Rupa 0.965*	Dubrovnik 0.876	Dubrovnik 0.886	Dubrovnik
18	*Dragani 0.993*	*Dragani 0.993*	Šibenik 0.974	Šibenik 0.984	Šibenik
19	*Draga 0.993*	*Draga 0.993*	*Krka 1.0*	*Krka 1.0*	Cres, Krk
20	*Nard 0.979*	*Nard 0.979*	Ključ 1.0 Lučko 1.0	Ključ 1.0 Lučko 1.0	Lučko, Ključ, Tounj, Zagreb, Mrežnica (rijeka), Ogulin
Vrijeme	2.72 min	2.56 min	0.69 min	0.69 min	
Σ	2.5/20	2.5/20	15.5/20	13.5/20	

Tablica 3. Usporedba rezultata Jaro Distance algoritma za usporedbu teksta i cijelog algoritma

Članak br.	Jaro Distance	Cijeli algoritam	Točan grad
1	Split 0.961	Split 1.986	Split
2	Zagreb 0.966	Zagreb 2.016	Zagreb
3	Rijeka 1.0	Zadar 1.975	Diklo, Puntamika, Zadar, (Rijeka)
4	Rovinj 1.0	Rovinj 2.025	Rovinj
5	Kaštel 0.970	*Marina 0.8*	Kaštel
6	-	-	-
7	*Grabar 0.9*	*Grabar 0.5*	-
8	Strošinci 0.834	Strošinci 1.834	Gunja, Strošinci
9	*Sela 1.0*	*Sela 2.0*	Dugo Selo
10	*Bandića 1.0*	Zagreb 2.016	Zagreb
11	Sveti Ivan 0.9 Zagreb 0.9	Samobor 1.945	Samobor, Zagreb, Sveti Ivan
12	-	-	-
13	Pećina 0.948	Pećina 0.448	Rijeka, Pećine (dio Rijeke)
14	Donje Orešje 0.9 *Orešje 0.9*	Donje Orešje 1.9 *Orešje 1.9*	Donje Orešje, Sveti Ivan Zelina
15	Zagreb 1.0	Zagreb 2.0	Zagreb
16	Gromačnik 0.978	Gromačnik 1.978	Slavonski Brod, Gromačnik
17	Dubrovnik 0.876	Dubrovnik 1.876	Dubrovnik
18	Šibenik 0.974	Šibenik 1.999	Šibenik
19	*Krka 1.0*	Krk 1.991	Cres, Krk
20	Ključ 1.0 Lučko 1.0	Tounj 2.0 Lučko 2.0	Lučko, Ključ, Tounj, Zagreb, Mrežnica (rijeka), Ogulin
Σ	15.5/20	16.5/20	

CD koji sadrži datoteke ovog rada

danas.avi

danas.py

danas.txt

funkcije.py

glas-slavonije.avi

glas-slavonije.py

glas-slavonije.txt

istranews.avi

istranews.py

istranews.txt

jutarnji.avi

jutarnji.py

jutarnji.txt

Životopis

OSOBNE INFORMACIJE

Ime i prezime	Nikolina Vidonis
Datum i mjesto rođenja	13.4.1990., Kopar, Slovenija
Adresa prebivališta	Ulica Vladimira Nazora 7 52470 Umag
e-mail	nvidonis@gmail.com
LinkedIn profil	hr.linkedin.com/in/nikolinavidonis

OBRAZOVANJE

14. srpnja - 25. srpnja 2014.	Sudjelovanje u Ljetnoj školi GIS-a
2012. – 2014.	Diplomski studij geoinformatike Geodetski fakultet, Zagreb
prosinac 2012.	Sudjelovanje na radionici PyLadies
2009. – 2012.	Preddiplomski studij geodezije i geoinformatike Geodetski fakultet, Zagreb
2005. – 2009.	Opća gimnazija SŠ Mate Balote, Poreč

EDUKACIJSKA POSTIGNUĆA

ak. god. 2012./2013.	Nagrada fakulteta za odličan uspjeh 3% najboljih studenata na godini

RADNO ISKUSTVO

listopad 2013. – lipanj 2014.	demonstrator pri Katedri za geoinformatiku
1.7.2013. – 1.9.2013.	Student radnik Terenski rad i izrada elaborata CONSTRUCTA GEO. D.O.O., Umag

OSOBNE VJEŠTINE

Poznavanje stranih jezika	
engleski	Čitanje: dobro Pisanje: dobro Govor: dobro
talijanski	Čitanje: izvrsno Pisanje: izvrsno Govor: izvrsno
Socijalne vještine	Pristupačnost, ljubaznost i komunikativnost razvijena tijekom obavljanja demonstratura na fakultetu
Organizacijske vještine	Organizirana timski nastrojena osoba, stečeno radom u timu
Računalne vještine	GIS softveri: QGIS, GRASS GIS, SAGA GIS, Geomedia CAD softveri: Autodesk Map 3D, ZwCAD, OCAD, grafički softveri: Inkscape, Google Sketchup Programski jezici: Python Ostalo: TNTmips, MS Office, OpenOffice
Vozačka dozvola	B kategorija

DODATNE INFORMACIJE

Tijekom ak. god. 2013./2014. sudjelovala sam u izradi studentskog časopisa Ekscentar, br. 17 s dva stručna članka:

Izrada turističke mrežne karte grada Duge Rese pomoću GIS Cloud tehnologije (J. Antolović, M. Giljanović, V. Jurić, R. Kozić, F. Todić, N. Vidonis), str.45-49

StarFire SBAS – uspostava, korištenje, performanse, perspektive (N. Vidonis, H. Vukašinović, M. Žugčić), str.71-76

Tijekom ak. god. 2012./2013. sudjelovala sam u radionici za studentski časopis Ekscentar, br. 16: *Arhiva, pohrana i distribucija prostornih podataka*