Dokumentacija projekta – Upravljanje znanjem

1. Prvi dio projekta – scrapanje informacija s web portala

U prvom djelu projekta bilo je potrebno izabrati portal te scrapati sve članke datirane u periodu od 01.01.2020 do 30.11.2020. Izabrani portal zove se „*baranjainfo*“ te je u idućim koracima opisan kod i rezultat koji su dobivenu web scrapanjem portala.

Napomena: u priloženim programskim skriptama ukratko su opisani algoritmi i svrha korištenja istog, te će se u ovom dokumentu opisati i pojasniti glavni djelovi programskih skripti.

1.1. Izrada web scrapera – lista linkova

```
# prikupljanje url-a stranica iz određenog perioda 01.01.2020 - 30.11.2020.
linkovi=[]

for i in range(1, 210):
    res = requests.get('https://www.baranjainfo.hr/page/'+str(i)+'/?s')
    soup = BeautifulSoup(res.text, 'lxml')

    glavni_blok = soup.find('div', class_='td-ss-main-content')
    blok = glavni_blok.find_all('h3', class_='entry-title td-module-title')
    for svaki in blok:
        link = svaki.find('a')
        linkovi.append(link['href'])
```

Slika 1. Prikupljanje linkova

Na slici 1. prikazan je dio koda koji služi za prikupljanje url-a svih članaka u periodu od 01.01.2020 do 30.11.2020. Brojevi unutar zagrade (1 i 210) označavaju broj stranica arhive unutar kojih se nalaze članci iz navedenog perioda. Unutar svakog bloka traži se tag „*a*“ koji sadrži link pojedinog članka te se navedeni spremaju u listu linkova. Rezultat izvršavanja ovog koda izgleda ovako (slika 2.):

File Edit Format View Help

```
https://www.baranjainfo.hr/kompostiranje-iskoristavanje-otpada-u-svrhu-plodnijeg-vrta/
https://www.baranjainfo.hr/aladrovic-za-pomoc-poduzetnicima-predvideno-25-milijarde-kuna/
https://www.baranjainfo.hr/slobodna-radna-mjesta-u-baranji-3/
https://www.baranjainfo.hr/dani-tehnicke-ispravnosti-vozila-2020/
https://www.baranjainfo.hr/preminuo-glumacki-velikan-mustafa-nadarevic/
https://www.baranjainfo.hr/001stozer-u-hrvatskoj-danas-3308-novih-slucajeva-zaraze-preminulo-4
https://www.baranjainfo.hr/u-osjecko-baranjskoj-zupaniji-168-nova-slucaja-zaraze/
https://www.baranjainfo.hr/beros-vjerojatno-sam-se-zarazio-u-ministarstvu-iako-sam-se-trudio-z
https://www.baranjainfo.hr/dnevni-servis-prometne-nesrece-14/
https://www.baranjainfo.hr/kako-bolje-komunicirati/
https://www.baranjainfo.hr/od-ponoci-na-snazi-stroze-epidemioloske-mjere/
https://www.baranjainfo.hr/bjelica-srusio-mamica-osijek-pobijedio-vodeci-dinamo/
https://www.baranjainfo.hr/dnevni-servis-prometne-nesrece-13/
https://www.baranjainfo.hr/u-hrvatskoj-rekordna-3573-nova-slucaja-zaraze-koronavirusom/
```

Slika 2. Lista url-a stranica

Ovaj dokument sadrži ~ 2000 članaka s navedenog portala.

1.2. Izrada web scrapera – prikupljanje potrebnih informacija s članaka

Nakon stvaranja csv pisača, slijedi izrada algoritma za prikupljanje potrebnih informacija iz svakog članaka unutar navedenog perioda. Informacije koje su uspješno prikupljenje su naslov, tekst članka, datum objave članka, kategorija unutar koje se nalazi članak te autor članka.

```
# scrapanje svih stranica sa popisa url-a za naslov, sadržaj, datu, autora i kategoriju
contents = []
with open('linkovi.csv','r') as csvf: # otvaranje datoteke s linkovima
    urls = csv.reader(csvf)
    for url in urls:
        contents.append(url) # dodavanje svakog url-a u listu sadržaja - contents

for url in contents:  # parsanje svakog url-a
    page = urlopen(url[0]).read()
    soup = BeautifulSoup(page, 'html5lib')
    response = requests.get(url)

    article = soup.find('article')
```

Slika 3. Izrada soup objekta za pretraživanje članka

Svaki url članka (slika 3.) iz prethodno stvorene datoteke pridodaje se u listu „*contents*“ te se svaki url pretražuje kako bi se pronašao tag „*article*“ unutar kojeg se html elementa nalazi sav potreban sadržaj za ovaj projekt. Navedeni sadržaj spremamo u objekt „*article*“.

```
#naslov članka
headline = article.h1.text

#tekst
paragraphs = ["".join(x.findAll(text=True)) for x in article.findAllNext("p")]

#dohvaćanje datuma
datum = article.find('div', class_='td-module-meta-info').time.text
#print(datum)

#tko je napisao članak
autor = article.find('div', class_='td-post-author-name').a
authors = autor.text
#print(autor.text)

#dohvaćanje kategorije
category = article.find('div', class_='td-post-header').ul
for cat in category.find_all('a'):
    cats = cat.text
    #print(cats)
```

Slika 4. Prikupljanje pojedinih informacija iz članaka

Na slici 4. prikazano je prikupljanje prethodno navedenih informacija unutar članka. Naslovu pristupamo na način da tražimo tekst unutar html elementa *„h1“*. Za prikupljanje svih paragrafa potrebno je pomoću funkcije *„join“* spojiti sav tekstualni sadržaj unutar svakog html elementa *„p“* koji se nalazi unutar *„article“* elementa. Datum se nalazi unutar diva određene klase, kao i autor. Kateogrije, kao i datum i autora, pretražujemo sve *„div“* elemente određene klase te prikupljamo tekst unutar svakog *„a“* html elementa. Nakon što su prikupljene navedene infromacije, isti sadržaj ispisujemo u csv datoteku te za potrebe projekta obavlja se konverzija iste u json datoteku. Ovaj dio koda nalazi se unutar *„test.py“* programske skripte. Rezultirajuća csv datoteka izgleda ovako (slika 5.):

Broj	Link	Naslov	Tekst	Datum objave	Autor	Kategorija
1	https://www.baranjainfo	Dnevni servis: Pı	U protekl	25. studeni 2020.	Sara Ilić	VIJESTI
2	https://www.baranjainfo	Dnevni servis: O	Sutra, 26.	25. studeni 2020.	Marko Balukčić	VIJESTI
3	https://www.baranjainfo	Međunarodni da	Međunar	25. studeni 2020.	Sara Ilić	NAJNOVIJE
4	https://www.baranjainfo	Stožer: 3.603 nov	U protekl	25. studeni 2020.	Sara Ilić	VIJESTI
5	https://www.baranjainfo	Kamion se prevr	Danas ok	25. studeni 2020.	Sara Ilić	VIJESTI
6	https://www.baranjainfo	U OBŽ crni rekor	Na konfe	25. studeni 2020.	Sara Ilić	VIJESTI
7	https://www.baranjainfo	OBŽ: Rezultati p	Financijsk	25. studeni 2020.	Marko Balukčić	ŽIVOT
8	https://www.baranjainfo	Župani i Plenkov	Predsjedr	25. studeni 2020.	Sara Ilić	VIJESTI
9	https://www.baranjainfo	Europska komisi	Europska	25. studeni 2020.	Sara Ilić	VIJESTI
10	https://www.baranjainfo	Sveta Kata, snije	Danas se	25. studeni 2020.	Baranja info	VIJESTI
11	https://www.baranjainfo	Božinović najavi	Ministar u	24. studeni 2020.	Sara Ilić	VIJESTI
12	https://www.baranjainfo	Hrvatska: 2.323 r	U protekl	24. studeni 2020.	Sara Ilić	VIJESTI
13	https://www.baranjainfo	Dnevni servis: Pı	U protekl	24. studeni 2020.	Sara Ilić	VIJESTI
14	https://www.baranjainfo	U Osječko-baran	PRIOPĆEN	24. studeni 2020.	Sara Ilić	VIJESTI
15	https://www.baranjainfo	Otvoreni natječa	Agencija :	24. studeni 2020.	Sara Ilić	VIJESTI

Slika 5. Csv datoteka s informacijama o člancima

Stupac *„Broj“* pridodan je tablici u svrhu kreacije jedinstvenog broja članka za potrebe konverzije csv datoteke u json datoteku.

2. Drugi dio projekta - Analiza vezana za tematiku koronavirusa

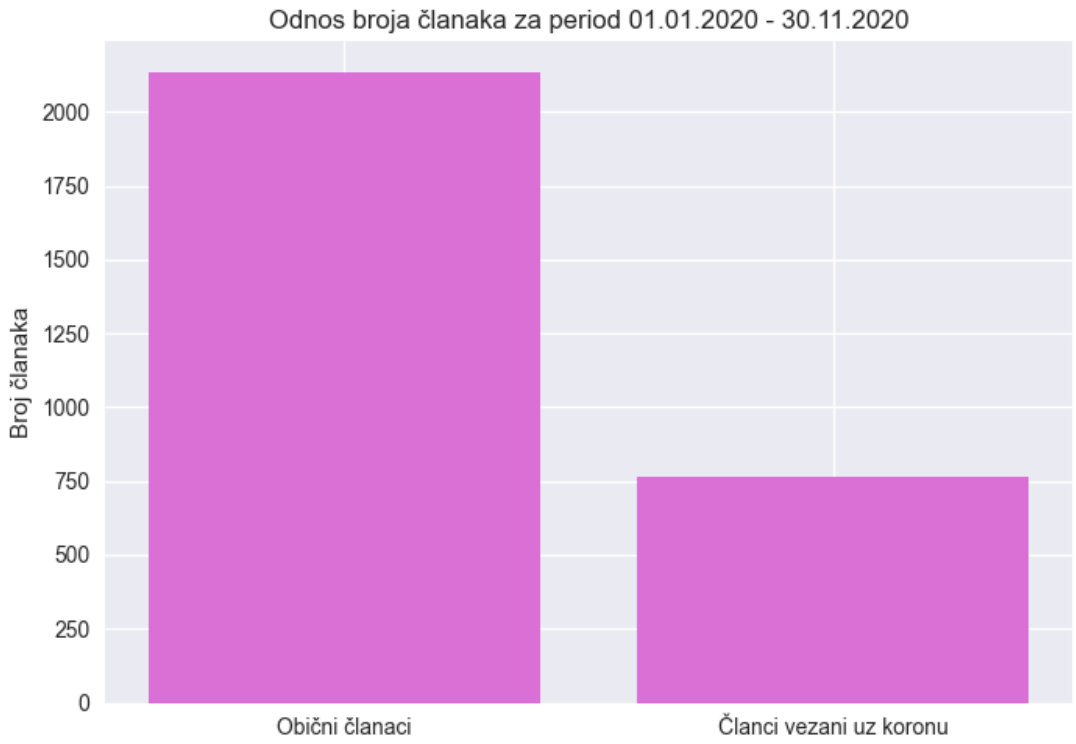
U ovom poglavlju opisana je analiza navedenih članaka prema određenim koracima.

2.1. Kvantificiranje članaka u medijima i vizualizacije

Unutar ovog koraka potrebno je numerički kvantificirati ukupan broj članaka unutar navedenog perioda i ukupan broj članaka vezan uz koronu. Isto tako, numerički su prikazani brojevi članaka grupirani prema danu, mjesecu i kategoriji. Prije same analize, bilo je potrebno proučiti podatke (kojeg su tipa podaci, da li psotoje NaN vrijednosti,...). Unutar programske skripte, nalazi se kod i kratak opis uz navedene postupke: izbacivanje nepotrebnih varijabli (sam link i broj članka nisu važni za navedenu analizu), varijabla *„Datum objave“* preimenovana je u *„Datum“* zbog praktičnosti (Python je dosta osjetljiv na razmake u varijablama) te je ista varijabla iz vrste podataka *„object“* (u prijevodu tekst, string) konvertirana u tip podataka *„Datetime“* jer ćemo u budućnosti trebati pristupati pojedinim elementima ove varijable (npr. tribamo pristupiti elementu „mjesec“ kako bi mogli grupirati podatke prema mjesecima) te su svi stringovi (naslov, tekst, autor i kategorija) konvertirani na način da se u tekstovima nalaze samo mala slova.

2.1.1. Ukupan broj objava na portalu i broj objava vezanih za korona tematiku

Ukupan broj članaka unutar datoteke iznosi 2134. Kako bi dobili broj članaka vezanih uz korona tematiku, potrebno je izradili listu s pojmovima vezanih uz korona tematiku (stožer, cjepivo, korona, koronavirus, bolnice, Capak, novooboljeli, karantena,...) te nakon toga stvaramo novi Data Frame koji sadrži članke iz prvotnog Data Frame-a čiji naslovi i tekstovi sadrže neke od navedenih ključnih riječi. Pozivanjem funkcije *„shape“* dobijemo broj redova (članaka) koji se odnose na korona tematiku i on iznosi 763. Pomoću navedneih brojki možemo izračunati i postotak članaka koji se odnose na korona tematiku, koji iznosi ~36%. Pomoću *matplotlib* biblioteke možemo grafički prikazati navedeni omjer. Grafikon se nalazi na slici 6.



Slika 6. Odnos ukupnog broja članaka i broja članaka vezanih uz koronu

2.1.2. Odnos broja objava i broja objava vezanih uz koronu prema datumima

```
# grupirano prema danima i brojevi clanaka
grupirano_po_datumima = podaci.groupby("Datum")
brojevi_clanaka_datum = grupirano_po_datumima.size().reset_index(name="Broj_članaka")
grupirano_po_datumima.head()

#izradag grupiranog skupa podataka prema datumima
grupirana_datum = grupirano_po_datumima[["Naslov", "Tekst", "Autor", "Kategorija"]].agg(lambda column: " ".joir
grupirana_datum

#izradag grupiranog skupa podataka prema datumima vezano uz koronu
grupirano_po_datumima_korona = ukupno_korona.groupby("Datum")
brojevi_clanaka_datum_korona = grupirano_po_datumima_korona.size().reset_index(name="Broj_članaka_korona")
grupirano_po_datumima.head()

#join dataframe-a za tablični prikaz
final_datum_korona = brojevi_clanaka_datum.merge(brojevi_clanaka_datum_korona, on='Datum')
final_datum_korona['Datum'] = pd.to_datetime(final_datum_korona['Datum']).dt.date
```

Slika 7. Grupirane članaka prema datumima

Na slici 7. prikazan je kod za dobivanje tabličnog prikaza ukupnog broja članaka i broja članaka vezanih uz koronu, grupiranih prema datumima. Najprije grupiramo skup podataka prema datumima i pomoću funkcije *„size“* možemo dobiti ukupan broj članaka. Kako bi grupirali sve naslove, tekstove, autore i kategorije, koristimo funkciju za agregiranje *„agg“* koja spaja sve navedene varijable. Kako bi dobili broj članaka vezanih uz koronu koirstimo isti postupak samo nad drugim skupom podataka – *ukupno_korona* koji sadrži sve članke vezane uz korona tematiku. Data Frame-ovi ovih postupaka izgledaju ovako (slika 8.):

Index	Datum	Broj_članaka
0	2020-01-01 00:00:00	5
1	2020-01-02 00:00:00	9
2	2020-01-03 00:00:00	6
3	2020-01-04 00:00:00	3
4	2020-01-05 00:00:00	13
5	2020-01-06 00:00:00	7
6	2020-01-07 00:00:00	6
7	2020-01-08 00:00:00	8

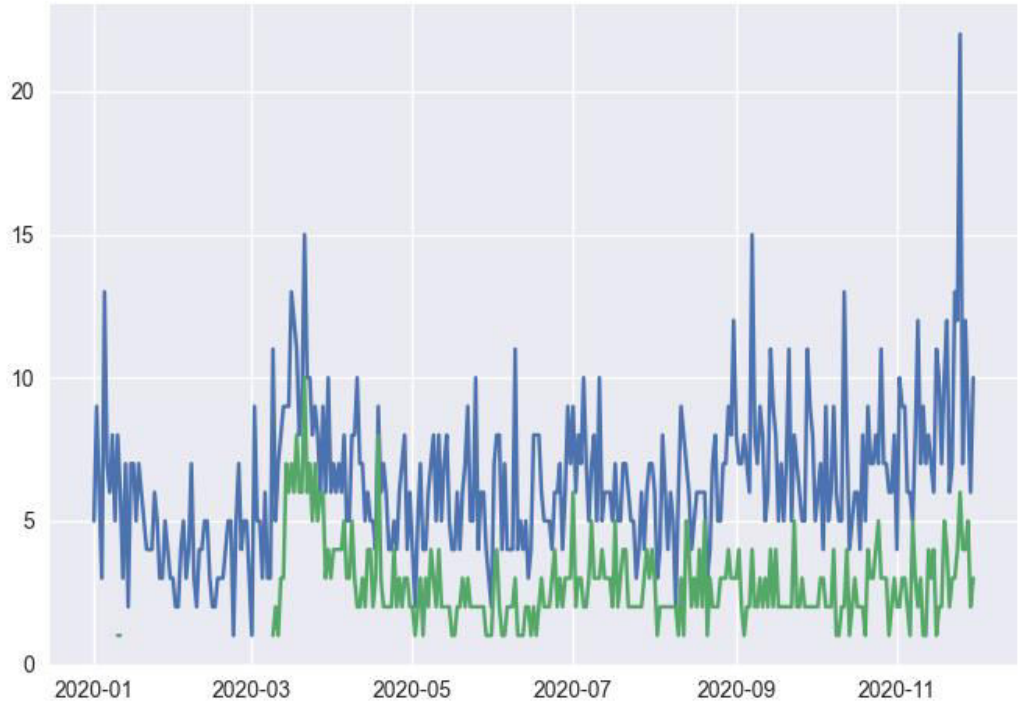
Slika 8. Ukupan broj članaka prema datumima

Na kraju koristimo funkciju *„merge“* kako bi spojili navedena dva Data Frame-a kako bi dobili ukupan prikaz brojeva članaka te brojeva članaka vezanih uz koronu grupiranih prema datumima. Spojeni Data Frame izgleda ovako (slika 9.):

Index	Datum	Broj_članaka	Broj_članaka_korona
0	2020-01-01	5	nan
1	2020-01-02	9	nan
2	2020-01-03	6	nan
3	2020-01-04	3	nan
4	2020-01-05	13	nan
5	2020-01-06	7	nan
6	2020-01-07	6	nan
7	2020-01-08	8	nan
8	2020-01-09	5	nan
9	2020-01-10	8	1

Slika 9. Brojevi članaka prema datumima

Na slici 9. vidimo tablični prikaz brojeva članaka prema datumima. Unutar funkcije *„merge“* korišten je parametar *„how“* postavljen na „*outer*“ kako bi se zadržali svi redovi Data Frame-ova. Iz prvih 10 redova možemo vidjeti da se prvi članak o koroni pojavljuje tek 10.01.2020, dok za ostale datume vidimo brojeve članaka neovisno o tematici. Da bi dobili samo brojeve članaka koji se pojavljuju u oba Data Frame-a možemo koristiti „*inner*“ join. Na slici 10. nalazi se grafički prikaz navedenog Data Frame-a. Linije su malo grube, ali to je rezultat velikog broja dana unutar navedeih mjeseci.



Slika 10. Broj članaka prema datumima

Može se primjetiti nagli rast broja članaka vezano uz koronu u trećem mjesecu kada je virus i stigao u Hrvatsku. Iako je broj članaka uvijek manji od ukupnog broja, možemo vidjeti sukladan rast tj. rastom broja članaka povećava se također i broj članaka o koroni.

2.1.3. Odnos broja objava i broja objava vezanih uz koronu prema mjesecima

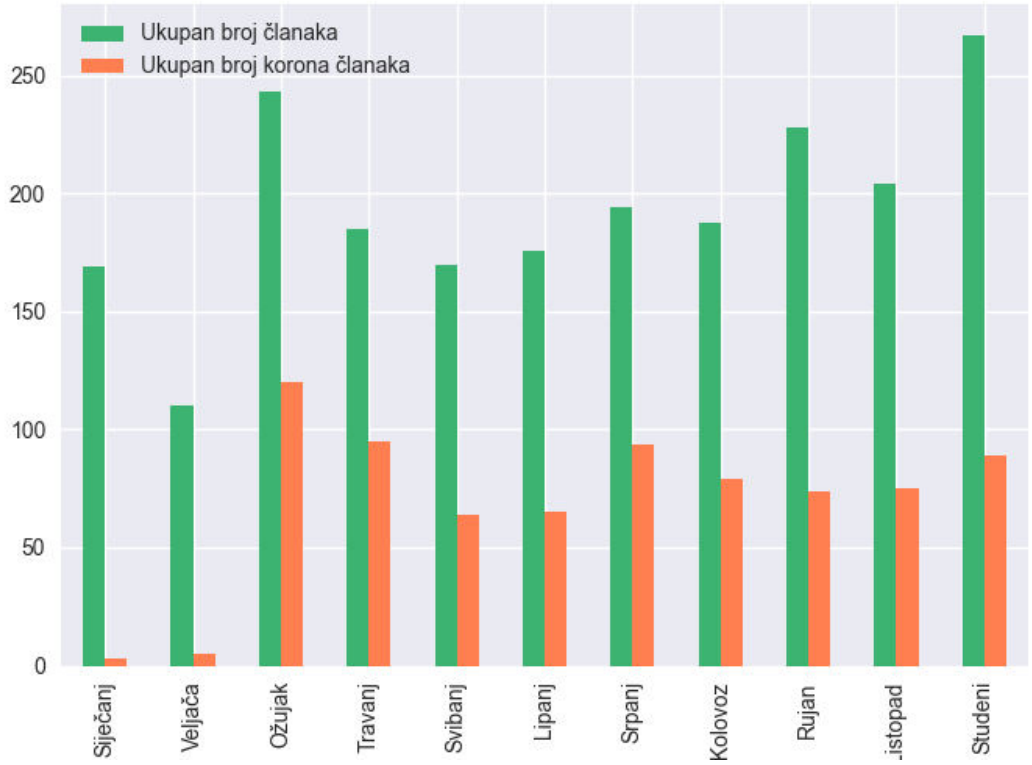
Broj članaka grupiran prema mjesecu dobiven je na sličan način, te umjesto da smo označili cijelu varijablu kao varijablu grupacije (prethodno „*Datum"*) u ovom slučaju je bilo potrebno pristupiti dijelu varijable koji označava mjesec (*month*) te istoj pristupamo tako da grupiramo Data Frame prema varijabli *„podaci.groupby(podaci.Datum.dt.month)"* koja označava pojedini mjesec unutar datuma. Jedino što je korišteno unutar ove podanalize a ne kod prethodne jest da su preimenovane vrijednosti unutar varijable „Datum" tako da umjesto brojeva 01, 02, 03.. koje označavaju mjesece koriste puni nazivi mjeseca (Siječanj, Veljača, Ožujak, itd.). Grupiramo Data Frame „*podaci*" i Data Frame „*ukupno_korona"* prema mjesecima, također koristimo

funkciju *„size“* za dobivanje broja članaka te spajamo navedena dva Data Frame-a prema mjesecima (također koristimo *„outer“ join*). Tablični prikaz prikazan je na slici 11.

Datum	Broj_članaka	Broj_članaka_korona
Siječanj	169	3
Veljača	110	5
Ožujak	243	120
Travanj	185	95
Svibanj	170	64
Lipanj	176	65
Srpanj	194	94
Kolovoz	188	79
Rujan	228	74
Listopad	204	75
Studeni	267	89

Slika 11. Tablični prikaz podataka prema mjesecima

Sa slike 11. možemo primjetiti nagli porast broja članaka vezanih uz koronu u ožujku te minimalno smanjenje i kontinuirano objavljivanje članaka vezanih uz koronu kroz proljeće, ljeto i jesen.



Slika 12. Bar plot broja članaka prema mjesecima

Na slici 12. prikazan je ukupan broj članaka i broj članaka vezanih uz koronu prema mjesecima pomoću *„bar“* grafikona. Ovo vizualizacijom bolje vidimo omjer broja članaka, dok isto znanje možemo prikazati i linijskim grafikonom gdje se bolje vidi kontinuiranost objavljivanja navedenih članaka, Linijski grafikon prikazan je na slici 13.

Slika 13. Linijski grafikon broja članaka prema mjesecima

Sa slike 13. možemo primjetiti kako je, naglim rastom ukupnog broja članaka u ožujku također porastao i broj članaka o koroni, tj. to je predstavljalo prijelomni mjesec tijekom navedene epidemije. Tada je virus bio novina i više se pisalo po raznim portalima o novonastalim događajima.

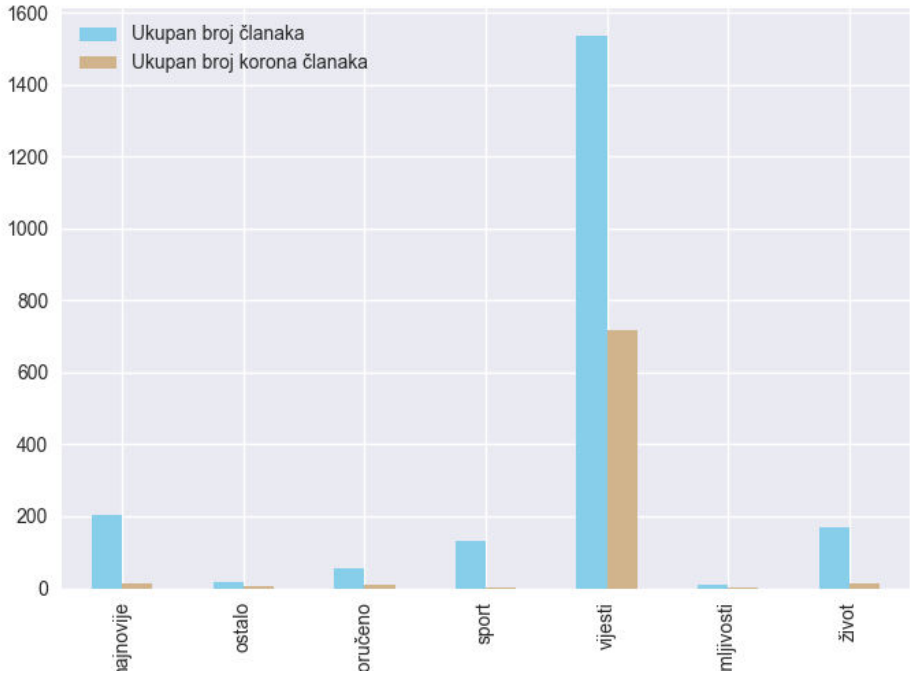
2.1.3. Odnos broja objava i broja objava vezanih uz koronu prema kategorijama

Također, kao što smo dobili podatke za datume i mjesece, možemo grupirati oba Data Frame-a prema varijabli *„Kategorija“*. Koriste se iste funkcije i metode kao i kod prethodna dva koraka te je na slici 14. prikazan tablični prikaz rezultata.

Kategorija	Broj_članaka	Broj_članaka_korona
najnovije	204	12
ostalo	16	4
preporučeno	56	9
sport	131	2
vijesti	1536	718
zanimljivosti	9	3
život	168	15

Slika 14. Brojevi članaka prema kategorijama

Iz navedenog tabličnog prikaza na slici 14. možemo vidjeti da najveći broj korona članaka pripada kategoriji *„vijesti“*, dosta manji pod kategorije *„najnovije“* i *život“* te se vrlo mali broj članaka vezanih uz koronu nalazi unutar ostalih kategorija. Iste rezultate možemo prikazati i grafički pomoću bar grafikona gdje se jasno vidi razlika između kategorija (slika 15.).



Slika 15. Brojevi članaka prema kategorijama

2.2. Analiza jezičnog diskursa korištenog na portalu

U ovom koraku analizira se jezični diskurs korišten u objavama vezanim uz korona tematiku. Analiziraju se tekstovi naslova, kao i tekstovi članaka (tagovi se ne analiziraju s obzirom da portal nema tagova).

2.2.1. Čišćenje članaka vezanih uz koronu

```
# izbacivanje posebnih znakova i zaustavnih riječi
zaustavne = ["a", "ako", "ali", "bi", "bih", "bila", "bili", "bilo", "bio", "bismo", "b
pattern = r'\b(?:{})\b'.format('|'.join(zaustavne))
novi_korona = ukupno_korona
novi_korona['Naslov1'] = novi_korona["Naslov"].str.replace(pattern, '')
novi_korona['Tekst1'] = novi_korona["Tekst"].str.replace(pattern, '')

# drop starih varijabli
novi_korona = novi_korona.drop(columns = ["Naslov", "Tekst"])
novi_korona['Datum'] = pd.to_datetime(novi_korona['Datum']).dt.date
```

Slika 16. Kod za izbacivanje zaustavnih riječi

Na slici 16. vidljiv je kod za izbacivanje zaustavnih riječi iz teksta. Data Frame *„novi_korona“* ssastoji se od svih članaka vezanih uz korona tematiku grupiranih prema mjesecima. Lista *„zaustavne“* sastoji se od 180 zaustavnih riječi hrvatskog jezika pronađenih na internetu. Varijabla *„pattern“* predstavlja regex (regularni izraz) pomoću kojeg se pretražuju navedene riječi unutar liste zaustavnih riječi. Stvaramo nove varijable (stupce) unutar Data Frame-a koje sadrže pročišćen tekst te kasnije uklanjamo varijable sa starim tekstom iz Data Frame-a. Isti regex i metode koriste se za uklananje priloga, prijedloga, zamjenica, čestica i sličnih riječi koje ne nose osobitu informacijsku vrijednost. Ta lista unutar koda zove se *„rijeci“* te sadrži 464 različitih riječi koje je potrebno izbaciti iz teksta. Također stvaramo nove varijable s pročišćenim tekstom i uklanjamo stare, nepročišćene varijable.

```
# izbacivanje specijalnih znakova
spec_chars = ["!",'\"', "#", "%", "\$", "&","\'","\(","\)","\*","\+",",","-","\.","/",":",";","<", "=",">",
pattern1 = r'^\w+{}$'.format('|'.join(spec_chars))
print(pattern1)
novi_korona['Naslov1'] = novi_korona["Naslov"].str.replace(pattern1, '')
novi_korona['Tekst1'] = novi_korona["Tekst"].str.replace(pattern1, '')
novi_korona = novi_korona.drop(columns = ["Naslov", "Tekst"])
novi_korona.info()
```

Slika 17. Kod za uklanjanje specijalnih znakova

Na slici 17. vidljiv je kod za uklananje specijalnih znakova. Koristi se malo drugačiji regularni izraz, gdje se specificira da tražimo specijalni znak na početku, kraju riječi ili između riječi i mijenjamo ga sa praznim stringom. Unutar ove liste potrebno je bilo prepaziti na specijalne znakove koji nose određeno značenje unutar regularnih izraza u Pythonu, kao što su \, + (označava pojavljivanje jednog ili više znaka), * (označava pojavljivanje od nula ili više puta određenog znaka), *!, ?* i slične. Nakon daljnjeg procesiranja, opisanog u skripti koda, dobijemo ovakav Data Frame:

Index	Datum	Naslov	Tekst	
0	1	rezultati pojačanih mjera prometu vikenda…	siječnja području policijske uprave o…	vi
1	2	obž oboljelih stožer civilne zaštite spre…	prostorijama osječkobaranjske županije…	vi
2	3	capak zaraženo djece godina mlađi st…	ravnatelj hrvatskog zavoda javno zdravstv…	vi
3	4	stožer slučajeva koronavirusa dvije osob…	hrvatskoj protekla potvrđeno slučaj…	vi
4	5	osječkobaranjska županija novooboljelih os…	kliničkom bolničkom centru osijek tes…	vi
5	6	hrvatskoj novooboljelihradi ublažavanja p…	hrvatskoj zabilježena nova slučaja …	vi
6	7	stožer novooboljelih umrla osoba novoobo…	nacionalni stožer civilne zaštite izvijest…	vi
7	8	korona krize osječkobaranjska županija odo…	upravni odjel turizam kulturu sport osj…	vi
8	9	stožer hrvatskoj slučajeva zaraze peter…	protekla zabilježeno slučajeva o…	vi
9	10	novopozitivnih priprema dvorana gradski …	priopćenje stožera civilne zaštite osječko…	vi
10	11	vlada bolnicama pacijenata koronavirusom…	ministar zdravstva vili beroš izjavio pon… obustava održavanja svih amaterskih kulturn…	vi

Slika 18. Pročišćeni Data Frame

Na slici 18. vidimo pročišćeni Data Frame koji sadrži sve članke vezane uz koronu, grupiran prema mjesecima. Vidimo iz priloženog dijela teksta da ne postoje specijalni znakovi (točka, zarezi, dvotočke koji su se inače nalazili u naslovima i slično) te možemo koristiti navedeni skup podataka za daljnju analizu. Varijabla *„Datum“* predstavlja mjesece u godini te svaki redak voga skupa podataka označava članke jednog mjeseca. Liste najčešćih 25 riječi korištenih po mjesecu možemo dobiti na slijedeći način:

```
# liste najčešćih riječi po mjesecima
sijecanj = grupirana_korona.loc[0, ['Naslov', 'Tekst']]
sijecanj_count = sijecanj.str.split(expand=True).stack().value_counts(ascending = False)[:25]
```

Slika 19. Lista najčešćih riječi za siječanj

Pomoću metode *„loc“* možemo indeksirati Data Frame, gdje tražimo pripadajući redak za navedeni mjesec (0 je za siječanj, 1 za veljaču, 2 za ožujak, itd.) te važne varijable koje sadrže tekst koji analiziramo (*Naslov i Tekst*). Zatim „splitamo“ listu i tražimo frekvencije vrijednosti unutar nje i navodimo *„False“* za redoslijed kako bi dobili na prvim mjestima riječi sa najvećom frekvencijom i biramo prvih 25 riječi. Ovaj postupak ponovljen je za svih 11 mjeseci i prikaz riječi izgleda ovako:

Index	0
osoba	235
osobe	159
hrvatskoj	95
zaraze	82
samoizolaciji	78
osijek	77
županije	74
respiratoru	68
zaštite	68
slučajeva	67

Slika 20. Frekvencije riječi

Na slici 20. vidimo primjer ispisane liste riječi i njenih pripadajućih frekvencija. Ovakva lista izrađena je za sve mjesece unutar ove godine kako bi se mogle kasnije analizirati sličnosti korištenih riječi između različitih mjeseci. Nisu prikazani dijelovi koda i tablični prikazi za sve mjesece unutar ovog dokumenta jer su dosta repetetivni, kod je isti samo se mijenjaju imena varijabli i indeks kojem se pristupa drugačijem redu unutar skupa podataka.

2.2.2. Vizualizacija najčešće korištenih riječi prema mjesecima

Nakon kreacije pojedinih lista, potrebno ih je konvertirati u rječnike (u Pythonu je to prilično jednostavno, koristila se *„to_dict“* funkcija) te ih tada možemo vizualizirati pomoću *„Word Cloud“* funkcije. Opet, prikazan je postupak vezan uz jedan od navedenih mjeseci, s obzirom da je postupak identičan kod izrade svih vizualizacija.

```
wc_studeni = WordCloud(max_font_size=40, background_color = "white", \
                       collocations=False).generate(" ".join([(k + ' ') * v for k,v in string_studeni.items()])
fig = plt.figure()
plt.imshow(wc_studeni, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Slika 21. Izrada Word Cloud-a

Na slici 21. prikazan je kod za izradu Word Cloud-a za mjesece, u ovom primjeru studeni. U funkciji navodimo veličinu fonta riječi s najvećom frekvencijom, boju pozadine slike, te generiramo Word Cloud na temelju frekvencije i riječi unutar vrijednosti riječnika (prethdono napravljenog rječnika za svaki mjesec). Pomoću funkcije *„imshow“* vizualiziramo Word Cloud te isključujemo osi grafikona kako nam slika ne bi imala grid. Primjer za mjesec studeni izgleda ovako:

Slika 22. Word Cloud za studeni

Iz priložene slike možemo primjetiti da su, prema veličini fonta, riječi koje su najčeće korištene u člancima vezanih uz korona tematiku u mjesecu studenom slijedeće: osoba, županije, osijek,

civilne, stožera, mjere, području, samoizolaciji, pacijenata itd. Ovakav grafikon izrađen je za sve mjesece i priloženi su uz dokumentaciju i programsku skriptu projekta.

2.2.3. Jaccard Indeks

Jaccard indeks jedna je od metoda za izračun sličnosti stringova bazirana na tokenima (riječima). Temelji se na pronalasku zajedničkih riječi te dijeljenju navedenog broja sa ukupunim brojem jedinstvenih riječi. Formula za izračun Jaccard indeksa glasi:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

gdje brojnik predstavlja presjek skupova (broj zajedničkih riječi), a nazivnik uniju skupova (ukupan broj riječi korištenih u listama).

```
data = [jan, feb, mar, apr, may, jun, jul, aug, sep, okt, nov]
dm = [[ textdistance.jaccard(a, b) for b in data] for a in data]
textdistance_table ='\n'.join([''.join([f'{item:6.2f}' for item in row]) for row in dm])

df = pd.read_csv(StringIO(re.sub(r'[-+|]', '', textdistance_table)), sep='\s{2,}', engine='python', names = ["
df = df.rename(index = { 0 : "Siječanj", 1 : "Veljača", 2 : "Ožujak", 3: "Travanj", 4: "Svibanj", 5: "Lipanj",
```

Slika 23. Kod za matricu jaccard indeksa između listi

Na slici 23. prikazan je kod za dobijanje matrice čije su vrijednosti Jaccard indeksi između članaka pojedinih mjeseci. Varijable *jan, feb, mar,...* predstavljaju stringove dobivene iz prethodno napravljenih listi (*siječanj_count, veljača_count,...*) te se navedeni stringovi skupljaju u jednu listu nazvanu „*data*“. U varijablu „*dm*“ spremamo sve jaccard indekse za *a* i *b* kojima pristupamo određenim mjesecima unutar liste. Zatim izrađujemo tablicu, gdje će nazivi redova i stupaca biti mjeseci u godini a vrijednosti na presjecima Jaccard indeksi između navedena dva mjeseca. Kada navedenu tablicu pretvorimo u Data Frame i preimenujemo indexe iz brojeva u mjesece, Data Frame s Jaccard indeksima izgleda ovako:

Index	Siječanj	Veljača	Ožujak	Travanj	Svibanj	Lipanj	Srpanj	Kolovoz	Rujan	Listopad	Studeni
Siječanj	1	0.76	0.71	0.72	0.67	0.68	0.71	0.73	0.69	0.71	0.75
Veljača	0.76	1	0.71	0.75	0.72	0.68	0.69	0.73	0.73	0.71	0.72
Ožujak	0.71	0.71	1	0.86	0.8	0.81	0.79	0.84	0.8	0.81	0.79
Travanj	0.72	0.75	0.86	1	0.8	0.79	0.8	0.86	0.83	0.83	0.82
Svibanj	0.67	0.72	0.8	0.8	1	0.87	0.83	0.82	0.87	0.83	0.78
Lipanj	0.68	0.68	0.81	0.79	0.87	1	0.87	0.85	0.86	0.88	0.8
Srpanj	0.71	0.69	0.79	0.8	0.83	0.87	1	0.86	0.86	0.89	0.83
Kolovoz	0.73	0.73	0.84	0.86	0.82	0.85	0.86	1	0.88	0.87	0.85
Rujan	0.69	0.73	0.8	0.83	0.87	0.86	0.86	0.88	1	0.85	0.82
Listopad	0.71	0.71	0.81	0.83	0.83	0.88	0.89	0.87	0.85	1	0.86
Studeni	0.75	0.72	0.79	0.82	0.78	0.8	0.83	0.85	0.82	0.86	1

Slika 24. Tablica Jaccard indeksa između pojedinih mjeseci

Na slici 24. prikazana je matrica Jaccard indeksa. Vidimo da se na dijagonali nalaze maksimalne vrijednosti, s obzirom da svaki mjesec sam sa sobom ima 100% podudarnosti. Vrijednosti se bilježe sa decimalnim brojevima između 0 i 1, gdje 0 znači da liste riječi nisu nimalo slične, dok 1 znači da su identične. Promatrajući ovu tablicu, vidimo da siječanj i veljača imaju najmanje sličnosti sa ostalim mjesecima, s obzirom da je broj korona članaka u tim mjesecima vrlo malen, dok npr. između ostalih mjeseci vidimo Jaccard indeks od 0.7 do skoro 0.9 što znači da je jezični diskurs veoma sličan između navedenih mjeseci tj. riječi koje su korištene unutar članaka s najvećom frekvencijom dosta su slične u periodu između ožujka i studenog. Ako već želimo izdvojiti najviše slične mjesece prema korištenim riječima, to su: listopad i srpanj, listopad i lipanj te rujan i kolovoz s Jaccard indeksima 0.89 i 0.88 respektivno.

2.2.4. Sorensen

Ova metoda također predstavlja metodu za izračun sličnosti teksta baziran na riječima. Logika ove metode jest pronaći zajedničke riječi te dobiveni broj podijeliti sa ukupnim brojem riječi oba skupa. Formula:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

nam prikazuje da je, kod Sorensen metode, brojnik dvostruka vrijednost od presjeka oba skupa. Ideja iza ovog brojnika jest ta da, ako se riječ nalazi u ova skupa, njezin ukupan broj je dvostruka vrijednost presjeka (čime se uklanjaju duplikati). Nazivnik predstavlja zbroj riječi u oba skupa, koji za razliku od unije kod Jaccard indeksa, ne uklanja duplikate. Iz ovog razloga, ovaj algoritam će uvijek precijeniti sličnost dvaju skupova.

Index	Siječanj	Veljača	Ožujak	Travanj	Svibanj	Lipanj	Srpanj	Kolovoz	Rujan	Listopad	Studeni
Siječanj	1	0.86	0.83	0.84	0.8	0.81	0.83	0.85	0.81	0.83	0.86
Veljača	0.86	1	0.83	0.86	0.84	0.81	0.81	0.84	0.84	0.83	0.84
Ožujak	0.83	0.83	1	0.92	0.89	0.9	0.88	0.91	0.89	0.89	0.88
Travanj	0.84	0.86	0.92	1	0.89	0.88	0.89	0.93	0.91	0.91	0.9
Svibanj	0.8	0.84	0.89	0.89	1	0.93	0.91	0.9	0.93	0.91	0.88
Lipanj	0.81	0.81	0.9	0.88	0.93	1	0.93	0.92	0.93	0.94	0.89
Srpanj	0.83	0.81	0.88	0.89	0.91	0.93	1	0.92	0.92	0.94	0.91
Kolovoz	0.85	0.84	0.91	0.93	0.9	0.92	0.92	1	0.94	0.93	0.92
Rujan	0.81	0.84	0.89	0.91	0.93	0.93	0.92	0.94	1	0.92	0.9
Listopad	0.83	0.83	0.89	0.91	0.91	0.94	0.94	0.93	0.92	1	0.92
Studeni	0.86	0.84	0.88	0.9	0.88	0.89	0.91	0.92	0.9	0.92	1

Slika 25. Primjena Sorensen algoritma nad listama

Sa slike 25. vidimo, sukladno definiciji metode, da su sličnosti listi za pojedini mjesec precijenjene, što je prikazano većim indeksima za razliku od Jaccard indeksa. Ovdje su npr. listopad i lipanj procjenjeni kao 94% slični, no imaju barem 5-6 riječi razlike što dokazuje točnost definicije. Kao realniji prikaz sličnosti, izabrala bih Jaccard indeks.

2.2.5. Overlap koeficijent

Overlap koeficijent, ili drugim nazivom Szymkiewicz–Simpson koeficijent je mjera sličnosti koja mjeri preklapanje (engl. *overlap*) između dva skupa (teksta). Jako je sličan Jaccard indeksu i

definiran je kao presjek skupova podijeljen sa veličinom manjeg od dvaju skupova koji se analiziraju. Formula za izračun overlap koeficijenta glasi:

$$\text{overlap}(X, Y) = \frac{|X \cap Y|}{\min(|X|, |Y|)}$$

U slučaju kada je X podskup skupa Y, tada je koeficijent jednak 1. Nakon primjene metode unutar iste biblioteke (*„textdistance“*) dobijemo slijedeći rezultat:

Index	Siječanj	Veljača	Ožujak	Travanj	Svibanj	Lipanj	Srpanj	Kolovoz	Rujan	Listopad	Studeni
Siječanj	1	0.88	0.83	0.87	0.84	0.83	0.85	0.88	0.87	0.84	0.87
Veljača	0.88	1	0.84	0.87	0.87	0.82	0.82	0.86	0.88	0.83	0.84
Ožujak	0.83	0.84	1	0.94	0.93	0.92	0.9	0.94	0.94	0.9	0.89
Travanj	0.87	0.87	0.94	1	0.91	0.88	0.89	0.93	0.93	0.92	0.92
Svibanj	0.84	0.87	0.93	0.91	1	0.95	0.93	0.91	0.94	0.94	0.91
Lipanj	0.83	0.82	0.92	0.88	0.95	1	0.93	0.92	0.96	0.95	0.91
Srpanj	0.85	0.82	0.9	0.89	0.93	0.93	1	0.93	0.96	0.95	0.92
Kolovoz	0.88	0.86	0.94	0.93	0.91	0.92	0.93	1	0.96	0.95	0.94
Rujan	0.87	0.88	0.94	0.93	0.94	0.96	0.96	0.96	1	0.96	0.95
Listopad	0.84	0.83	0.9	0.92	0.94	0.95	0.95	0.95	0.96	1	0.93
Studeni	0.87	0.84	0.89	0.92	0.91	0.91	0.92	0.94	0.95	0.93	1

Slika 26. Overlap između pojedinih mjeseci

Na slici 26. možemo primjetiti da smo upotrebnom Overlap funkcije, kao i kod Sorensen funkcije, dobili veće koeficijente između skupova, od kojih su neki koeficijenti čak i veći nego prilikom upotrebe Sorensen algoritma. Kao najreprezentativniju metodu, izabrala bih Jaccard indeks.

3. Zaključak

Uzimajući u obzir konačne rezultate analize, možemo zaključiti da su članci pisani veoma šturo, s obzirom da se ponavljaju većinom iste riječi tijekom navedenog perioda. Članci su pisani u ozbiljnom i neutralnom (niti pozitivnom niti negativnom) tonu, ako uzmemo u obzir da su neke od najčešće korištenih riječi redom: osoba, mjere, županije, stožer, smaoizolacija, osječko baranjska, respirator i ostale gdje ne vidimo raznolik diskurs niti poseban ton pridodan određenom diskursu (pozitivan ili negativan). Prema rezultatima, moglo bi se reći da su se u člancima prenosile osnovne, odnosno najvažnije i najnovije vijesti, kao što su koliki je određeni dan bilo boljelih, koje su mjere donešene, nove obavijesti stožera, koliko se osoba nalazi u bolnici i slično, bez dodatnog proširenja sadržaja. U kontekstu negativnosti, spomenula bih samo da dosta mjeseci kao dosta frekventnu riječl sadrži riječ „novooboljelih“ dok se riječi „preboljeli“ i slične ne nadovde kao riječi koje su često zastupljene unutar članaka, pa bi se moglo reći da su članci pisani više u negativnom nego u pozitivnom smislu, ali to je i za očekivati od medija s obzirom da je riječ o epidemiji, te se u tom kontekstu, kao i kod ostale tematike, više spominju negativne posljedice nego pozitivne.