



REPUBLIKA SLOVENIJA

MINISTRSTVO ZA DIGITALNO PREOBRAZBO
Davčna ulica 1, 1000 Ljubljana



EVROPSKA UNIJA
EVROPSKI
SOCIALNI SKLAD
NALOŽBA V VAŠO PRIHODNOST



Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco

Elaborat

Dokument je namenjen orisu koncepta arhitekturne in funkcionalne zasnove informacijske rešitve za semantične in druge vrste naprednih obravnav besedil po njihovi vsebini oziroma pomenu, ki se uporabljajo pri delu in procesih državne uprave.

Avtorji: Jure Jeraj, Stevanče Nikoloski, Sabina Mamić

Datum: 22. 9. 2023

**Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično
obravnavo naravnega jezika z umetno inteligenco**

Jure Jeraj, Stevanče Nikoloski, Sabina Mamić

Datum nastanka: 22. 9. 2023 4. 10. 2023

Datum zadnje spremembe: 15. 11. 2024 16:46

Različica dokumenta: "Številka različice"

Pogodba št.: C3130-23-282013

Naročnik: Ministrstvo za digitalno preobrazbo

Skrbnik pogodbe: Jure Jeraj

Tabela sprememb

Različica	Datum	Opis
0.1	1. 8. 2023	Prva različica dokumenta
0.2	8. 9. 2023	Opis MLOps procesov
0.3	12. 9. 2023	Dopolnitev dokumentacije
0.4	15. 9. 2023	Spremenjena grafika dokumenta
1.0	22. 9. 2023	Zaključek dokumenta

Tabela 1: Različice dokumenta

Kazalo vsebine

0. Podatki o dokumentu.....	10
1. Uvod in namen	11
2. Povzetek dosedanjih aktivnosti (pilotni projekt).....	13
2.1. Analiza funkcionalnosti na področju semantične in drugih naprednih analiz besedil.....	13
2.1.1. Branje dokumentov s strežnika	14
2.1.2. Branje ontologij s strežnika	14
2.1.3. Branje člankov (Contributions to Contemporary History)	15
2.1.4. Branje člankov Elektrotehniškega vestnika.....	16
2.1.5. Predobdelava dokumenta	17
2.1.6. Pridobitev vektorskih predstavitev besedil	19
2.1.7. Uporaba razdalj in podobnosti	20
2.1.8. Odkrivanje skupin in izris kart dokumentov	20
2.1.9. Vektorske predstavitve besed	21
2.1.10. Odkrivanje skupin in izris kart dokumentov na podlagi besed specifičnih za skupine	22
2.1.11. Iskanje značilk z uporabo vložitev fastText	23
2.1.12. Iskanje značilk z uporabo obogatitve besed	24
2.1.13. Iskanje značilk z uporabo transformacije TF-IDF	24
2.1.14. Iskanje značilk z uporabo metod na grafih besed.....	25
Primerjava pristopov za specifične besede	26
2.1.15. Primerjava pristopov za luščenje značilk na anotiranih besedilih iz revije Priskevki za novejšo zgodovino.....	27
2.1.16. Primerjava pristopov za luščenje značilk na označenih besedilih iz revije Elektrotehniški vestnik	28
2.1.17. Odkrivanje skupin in izris kart dokumentov iz revije Elektrotehniški vestnik	30
2.1.18. Primerjava pristopov za luščenje značilk na primeru korpusa Schutz 2008.....	31

2.1.19. Odkrivanje skupin in izris kart dokumentov	32
2.1.20. Primerjava pristopov za luščenje značilk na primeru korpusa SemEval	34
2.1.21. Primerjava pristopov za luščenje značilk na primeru povzetkov člankov, ki vsebujejo besedo "longevity"	36
2.1.22. Primerjava nelematiziranih pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Longevity" - not lemmatized	38
2.1.23. Primerjava pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Covid-19"	39
Primerjava pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Longevity"	42
2.1.24. Vizualizacija gruč vložitev ključnih fraz AIIIM člankov o longevity in Covid-19	46
2.1.25. Primer v programskem paketu Orange	49
3. Procesi, vloge (uporabnikov) in primeri uporabe	53
3.1. MDP, Direktorat za podporo uporabnikov, eUprava.....	53
3.2. Ministrstvo za javno upravo, Direktorat za kakovost, Sektor za odpravo administrativnih ovir in boljšo zakonodajo	55
3.2.1. Primer uporabe	56
3.3. Služba vlade za zakonodajo.....	59
3.3.1. Primer uporabe	59
3.4. MDP, Direktorat za razvoj digitalnih rešitev in podatkovno ekonomijo	60
3.4.1. PzSI – Platforma za semantično interoperabilnost.....	62
3.4.2. Primer uporabe	62
4. Logična arhitektura	64
4.1. Integracija aplikacije »Semantični analizator« z drugimi zunanji aplikacijami	64
4.2. Arhitektura aplikacije »Semantični analizator«	65
4.2.1. Komunikacijsko vodilo	65
4.2.2. Algoritem za vložitve (angl. embeddings) podatkov.....	67

4.2.3. Podatkovne shrambe.....	68
4.2.4. Aplikacije	72
4.2.5. Avtentikacija.....	72
4.2.6. Dnevnik delovanja	72
4.2.7. ML modul: Opis procesa MLOps.....	73
5. Opredelitev uporabniških vmesnikov.....	77
5.1. Uporabniški vmesnik za vnos novih dokumentov, pridobivanje in validacijo podobnih dokumentov ter napredno poenotenje besedil.....	77
5.1.1. Vnos dokumentov in validacija uvrstitve dokumenta v gručo.....	77
5.1.2. Pridobivanje, pregled ter validacija podobnih vsebin	78
5.1.3. Napredno poenotenje vsebin	78
5.2. Uporabniški vmesnik za upravljanje z MLOps modulom.....	79
5.3. Uporabniški vmesnik za nadzor orodja SEMANT	79
6. Fizična arhitektura	80
7. Strojna in programska oprema	81
7.1. Specifikacija strojne opreme	81
7.2. Specifikacija programske opreme.....	82
8. Upravljalski procesi	84
8.1. Proces za shranjevanje novih vsebin	84
8.2. Proces za pridobivanje podobnih vsebin	86
8.3. Proces za napredno poenotenje oz. izbira enotnih besedil v službi za zakonodajo ..	87
8.4. Proces za proženje postopka rekaliibracije ML modela.....	88
8.5. Korespondenca med uporabniki in skrbniki sistema SEMANT	89
9. Nadaljnji razvoj informacijske rešitve (možni predlogi)	91
9.1. Proces za izdelavo povzetka dokumenta	91
9.2. Detekcija anomalij oz. kontradikcij	92

Kazalo tabel

Tabela 1: Različice dokumenta	3
Tabela 2: Prejemniki dokumenta	Napaka! Zaznamek ni definiran.
Tabela 3: Seznam izvedenih delavnic.....	Napaka! Zaznamek ni definiran.
Tabela 4: Seznam prejetih dokumentov, elektronske pošte.....	10
Tabela 5: Viri pilota	13
Tabela 6: Pet najpogostejših besed in uteži za primer »Spremembna stopnje DDV za stanovanjske nepremičnine«.....	25
Tabela 7: Metode na grafih besed: TextRank in RAKE.....	25
Tabela 8: Viri Enotnega kontaktnega centra	54
Tabela 9: Viri Direktorata za kakovost	56
Tabela 10: Viri Službe vlade za zakonodajo	60
Tabela 11: Viri Ministrstva za digitalno preobrazbo	61

Kazalo slik

Slika 1: WordCloud najpogostejših besed Contributions to Contemporary History	16
Slika 2: Besedni oblak najpogostejših besed člankov Elektrotehniškega vestnika	17
Slika 3: Besedni oblak najpogostejših besed v Zakonu o registrih s predobdelavo z delitvijo na pojavnice	18
Slika 4: Besedni oblak najpogostejših besed v Zakonu o registrih s predobdelavo z odstranjevanjem strukturnih delov zakonskih aktov	19
Slika 5: Odkrivanje skupin v dokumente predlog vladi (5 skupin).....	21
Slika 6: Gručenje vektorskih predstavitev besed	22
Slika 7: Odkrivanje skupin in izris kart dokumentov na podlagi besed specifičnih za skupine	23
Slika 8: Primerjalna analiza različnih pristopov za izbor specifičnih besed	27

Slika 9: Primerjava pristopov za luščenje ključnih besed na anotiranih besedilih iz revije »Prispevki za novejšo zgodovino«.....	28
Slika 10: Primerjava pristopov za luščenje ključnih besed na anotiranih besedilih iz revije »Elektrotehniški vestnik«.....	29
Slika 11: Razlaga izraznih kart dokumentov	30
Slika 12: Primerjava pristopov za luščenje ključnih besed na izhodiščnih podatkih Schutz 2008	32
Slika 13: Razlaga izraznih kart dokumentov	33
Slika 14: Primerjava pristopov za luščenje ključnih besed na izhodiščnih podatkih SemEval (brez transformacij)	34
Slika 15: Primerjava pristopov za luščenje ključnih besed na izhodiščnih podatkih SemEval (s transformejem RoBERTa).....	35
Slika 16: Primerjava pristopov za luščenje ključnih besed na povzetkih člankov s ključno besedo "Longevity" (brez lematizacije)	37
Slika 17: Primerjava pristopov za luščenje ključnih besed na povzetkih člankov s ključno besedo "Longevity" (z lematizacijo).....	38
Slika 18: Primerjava nelematiziranih pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Longevity" - not lemmatized	39
Slika 19: Primerjava pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Covid-19"	40
Slika 20: Histogrami za prikaz izluščenih ključnih besed iz število besedil povezanih na Covid 19	41
Slika 21: Primerjava pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Longevity"	44
Slika 22: Histogrami za prikaz izluščenih ključnih besed iz število besedil povezanih na ključne bvesede "Longevity"	45
Slika 23: Gruče ključnih fraz AIIIM člankov o Covid 19. Zgoraj levo je TF-IDF, zgoraj desno je YAKE, spodaj levo je BERT transformerja in spodaj desno je TextRank.....	47
Slika 24: Gruče ključnih fraz AIIIM člankov o Longevity. Zgoraj levo je TF-IDF, zgoraj desno je YAKE, spodaj levo je BERT transformerja in spodaj desno je TextRank.....	48
Slika 25: Potek dela v Orange. Primer uporabe: Določanje ključnih besed besediom iz sistema Predlagaj Vladi	49
Slika 26: Ocenjevanja ključnih besed s pomočjo metode TF-IDF.....	51

Slika 27: Zemljevid besedil dokumentov s predlogi vladi RS s prikazom skupin.	52
Slika 28: Primer uporabe Semanta za SOAOBZ	56
Slika 29: Primer uporabe semantičnega analizatorja v sistemu STOP birokraciji.....	58
Slika 30: Visokonivojska logična arhitektura	64
Slika 31: Podrobna logična arhitektura aplikacije SEMANT	65
Slika 32: Proces MLOps	73
Slika 33: Specifikacija infrastrukture za NVIDIA L40S 8xGPU 48GB, 1.5TB RAM, 2x Genoa Epyc CPU, 5x 1.9TB SSD-je, konfigurirani v RAID6 + 1 spare.....	82
Slika 34: Proces za shranjevanje novih vsebin	84
Slika 35: Proces za pridobivanje novih vsebin.....	86
Slika 36: Proces za napredno poenotenje	88
Slika 37: Proces za proženje rekaliibracije ML modela.....	89
Slika 38: Korespondenca med uporabniki in skrbniki sistema SEMANT.....	90
Slika 39: Diagram procesa za izdelavo povzetek dokumenta	92

0. Podatki o dokumentu

Dokument je pripravljen na osnovi Pogodbe št. C3130-23-282013 za koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco. Dokument je pripravljen v sodelovanju z MDP v sklopu delovnih sestankov in delavnic, ki so bili izvedeni v obdobju od 06.06.2023 do 18.07.2023. Dokument je pripravljen na osnovi do sedaj znanih informacij in prejetih dokumentov.

Dokument je posredovan v odprti obliki in ga lahko naročnik kasneje tudi sam dopolnjuje.

Tabela 2: Seznam prejetih dokumentov, elektronske pošte

1. Uvod in namen

Ministrstvo za digitalno preobrazbo RS (v nadaljevanju MDP) naroča dokument (elaborat), ki bo orisal koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantične in druge vrste naprednih obravnav besedil po njihovi vsebini oziroma pomenu, ki se uporabljajo pri delu in procesih državne uprave.

Na njegovi osnovi je potrebno razviti stabilno, razširljivo in prilagodljivo orodje na podlagi modernih arhitektur informacijskih rešitev. Orodje mora biti povezljivo z poljubnim obstoječim portalom ali aplikacijami v državni upravi, ki uporabljajo baze/zbirke besedil z uporabniškimi informacijami kot so vprašanja in odgovori, različni predlogi in pobude ter druge informacije.

Idejna zasnova koncepta temelji na pilotnem projektu Semantični analizator, tako da aplikacija sledi logiki in funkcionalnostim, razvitim v sklopu pilota. Semantični analizator (<https://nio.gov.si/nio/asset/semanticni+analizator+pametni+iskalnik+besedil>) deluje v okviru programskega paketa Orange. Funkcionalnosti iz pilotnega projekta, ki so se izkazale kot koristne in so popisane v poglavju 2.1, morajo biti vključene tudi v orodje SEMANT. Poleg tega je v dokumentu opredeljena zahtevana logična in fizična arhitektura informacijske rešitve ter način povezovanja s portali in aplikacijami državne uprave tudi z vidika potrebne strojne opreme ter ljudi, vlog in procesov, ki bodo potrebni za vzdrževanje in nemoteno delovanje informacijske rešitve z uporabo zanesljivih in ažurnih zbirk besedil ter modelov obdelave in analize besedil.

Informacijska rešitev SEMANT razširjuje pilotni projekt raziskav in razvoja semantičnega analizatorja. Namen informacijske rešitve SEMANT je, da se zaposlenim v državni upravi ponudi orodje, s katerim bodo lahko analizirali (velike) zbirke besedil po njihovi vsebini (npr. za pomoč pri iskanju odgovorov na vprašanja uporabnikov storitvenih portalov). Orodje mora omogočati osnovno oziroma okvirno opredelitev vsebinskih področij, ki se naslavlajo v posameznih zbirkah, pregled in prikaz rangiranih besedil po vsebinski sorodnosti, opredelitev okvirne vsebine izbranih besedil oziroma posameznega besedila s predlaganimi ključnimi izrazi ter povezavo med vsebinsko podobnimi zbirkami, tako da na podlagi enega ali več izbranih besedil iz ene zbirke identificira po vsebini sorodna besedila v drugi zbirki (rangirano glede na stopnjo sorodnosti, od najbolj do najmanj podobnih). Drugi sklop

funkcionalnosti pa mora omogočati povezavo med pojmi oziroma entitetami iz ontologij temeljnih registrov in evidenc državne uprave z besedili oziroma zbirkami besedil, v katerih so informacije (definicije, opisi pojmov, entitet, lastnosti, povezav) glede oblike in vsebine posameznih registrov in evidenc (npr. zakonodaja).

2. Povzetek dosedanjih aktivnosti (pilotni projekt)

Povzetki so pripravljeni in objavljeni na portalu NIO, vendar ni popisa vseh primerov uporabe:

Ime vira	Naslov vira
Semantični analizator besedil	https://nio.gov.si/nio/asset/semanticni+analizator+besedil?lang=sl
Semantični analizator – pametni iskalnik	https://nio.gov.si/nio/asset/semanticni+analizator+pametni+iskalniki+besedil

Tabela 3: Viri pilota

2.1. Analiza funkcionalnosti na področju semantične in drugih naprednih analiz besedil

Semantični analizator je odprtokodno prototipno orodje, ki uporablja tehnike umetne inteligence za analizo slovenskih besedil. Omogoča identifikacijo ključnih pojmov in tistih, ki manjkajo v določenem besednjaku, na primer v besednjaku javne uprave. Orodje je razvito v sodelovanju s Fakulteto za računalništvo in informatiko Univerze v Ljubljani in je del programskega paketa Orange. Analizira besedila, izlušči ključne pojme in na podlagi njihove sočasne pojavnosti v različnih besedilih določa sorodnost vsebin. Tako lahko uporabniki hitro pregledajo ter razvrstijo vsebine v skupine sorodnih dokumentov. Orodje omogoča iskanje dokumentov na podlagi nabora ključnih pojmov, pri čemer omogoča tudi iskanje po sopomenkah ključnih pojmov. Karakteristični pojmi služijo kot povezava med dokumenti, saj razkrivajo soodvisnosti in vsebinske povezave. Orange se lahko uporablja tudi za iskanje zakonskih dokumentov, ki se sklicujejo na določene podatkovne vire iz Centralnega besednjaka. Vsebine za analizo so dostopne na določeni spletni povezavi, prav tako tudi odprtokodni prototip z dokumentacijo.

Predlagan koncept za razvoj prosto dostopne spletne aplikacije temelji na tem prototipu.

V nadaljevanju so popisani vsi primeri uporabe prototipnega orodja »Semantični analizator – Pametni iskalnik besedil«.

2.1.1. Branje dokumentov s strežnika

Ta primer uporablja vmesnik (API) za prenos korpusov s strežnika ter branje in izpisovanje dokumentov z metapodatki.

Povezava do kode:

<https://github.com/biolab/text-semantics/blob/main/examples/01-01-loading-documents.ipynb>

2.1.2. Branje ontologij s strežnika

Ta primer uporablja vmesnik za prenos ontologije iz strežnika, branje in izpisovanje ontologije. Primer izpisa za ontologijo core-sskj-only.owl:

```
— Agencija
— BancniRacun
— Dejavnost
— Delez
— DelniskaDruzba
— Dovoljenje
— Drazba
— Drustvo
— Funkcija
— Imenik
— Indikator
— Informacija
— Izdatek
— Izplacilo
  — Placa
— Izvajalec
— Katalog
— Klasifikacija
— Kraj
— Ministrstvo
— Motor
— Narocilo
— Narocnik
— Naslov
— Oblika
  — FormatPodatkov
— Odlocha
— Organ
— Oseba
  — Clan
  — Nadzornik
  — Zastopnik
— Podatek
— Podjetje
```

```
├── Ponudba
├── Pravica
├── PravnaPodlaga
├── Prijavitelj
├── Seznam
├── Sklep
├── Slovar
├── SpletnoMesto
│   ├── SpletnaStran
├── Sporazum
├── Standard
├── Stanje
├── Status
├── Subvencija
├── Tabela
├── TelekomunikacijskaStoritev
│   ├── ElektronskaPosta
│   ├── Telefaks
│   └── Telefon
├── Uporaba
├── Uprava
├── Ustanovitelj
├── Vozilo
│   └── Plovilo
├── Zavezanec
├── ZbirkaPodatkov
├── Zbornica
├── Concept
├── ConceptScheme
├── Collection
│   └── OrderedCollection
```

Povezava do kode:

<https://github.com/biolab/text-semantics/blob/main/examples/01-02-ontologies.ipynb>

2.1.3. Branje člankov (Contributions to Contemporary History)

Primer uporablja vmesnik za prenos člankov s strežnika ter branje in izpisovanje dokumentov z metapodatki. Najpogostejše besede so prikazane preko WordCloud vizualizacije.

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 1: WordCloud najpogostejših besed Contributions to Contemporary History

Povezava do kode:

<https://github.com/biolab/text-semantics/blob/main/examples/01-03-CTCH-exploration.ipynb>

2.1.4. Branje člankov Elektrotehniškega vestnika

Ta primer uporablja vmesnik za prenos člankov s strežnika ter branje in izpisovanje dokumentov z metapodatki. Z uporabo algoritma TF-IDF (term frequency-inverse document frequency) je izbranih deset najpogostejših besed v korpusu. Za lažji prikaz najpogostejših besed v korpusu člankov Elektrotehniškega vestnika je uporabljena besedni oblak (WordCloud), ki je prikazan na Slika 2.

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 2: Besedni oblak najpogostejših besed člankov Elektrotehniškega vestnika

Povezava do kode:

<https://github.com/biolab/text-semantics/blob/main/examples/01-04-el.vestnik-exploration.ipynb>

2.1.5. Predobdelava dokumenta

Predobdelava razdeli dokument na pojavnice, odstrani odvečne besede ter najpogostejše besede prikaže v oblaku besed. Deset najpogostejših besed pri predobdelavi dokumenta z delitvijo na pojavnice so:

```
člen: 27
odstavek: 18
smrt: 16
oseba: 12
matičen: 11
register: 9
vpisati: 7
pristojen: 6
organ: 6
zakon: 5
```

Primer besednega oblaka za Zakon o registrih s predobdelavo z delitvijo na pojavnice je prikazan na Slika 3.



Slika 3: Besedni oblak najpogostejših besed v Zakonu o registrih s predobdelavo z delitvijo na pojavnice

Povezava do kode:

<https://github.com/biolab/text-semantic/blob/main/examples/02-01-document-exploration.ipynb>

Drug način predobdelave besedila je odstranjevanje strukturnih delov zakonskih aktov. Deset najpogostejših besed pri predobdelavi enakega dokumenta z odstranjevanjem strukturnih delov je:

matičen: 8
register: 7
odstavek: 6
istospolen: 6
partnerski: 6
skupnost: 6
podatek: 5
zakon: 4
registracija: 4
beseda: 4

Primer besednega oblaka za Zakon o registrih s predobdelavo z odstranjevanjem strukturnih delov zakonskih aktov je prikazan na Slika 4.



Slika 4: Besedni oblak najpogostejših besed v Zakonu o registriranih strokovnih delavcih z odstranjevanjem strukturnih delov zakonskih aktov

Povezava do kode:

<https://github.com/biolab/text-semantic/blob/main/examples/02-02-preprocessing-results.ipynb>

2.1.6. Pridobitev vektorskih predstavitev besedil

V tem zvezku predstavimo, kako lahko pridobimo vektorske predstavitve (vložitve) besed in dokumentov za analizo besedil.

V pilotnem primeru so preko vmesnika pridobljena besedila zadnjih 100 predlogov vladi, ki vsebujejo vsaj 50 znakov (s tem pogojem se priskrbi, da po predprocesiranju ni praznega seznama pojavnic). Pridobljenih je 99 dokumentov, ki se predobdelajo tako, da se iz besed najprej odstranijo končnice, potem pa se besede pretvorijo v seznam lematiziranih pojavnic.

Po tem procesu je mogoče vsak dokument predstaviti kot vektor v vektorskem prostoru, ki ga določa vreča besed. Vsak atribut vreče besed predstavlja eno besedo v slovarju, vsaka vrstica pa en dokument.

Vreča besed se shrani v obliki tabele, ki predstavlja število pojavitev posamezne besede v posameznem dokumentu. Tabelo je mogoče prilagoditi tako, da se upošteva pogostost besed - manj pogoste, a pomembne besede bodo imele višjo vrednost kot take, ki so

vseprisotne. Poleg te metode so dokumenti predstavljeni z uporabo modela fastText, ki temelji na nevronskih mrežah in je prednaučen na velikem korpusu dokumentov. V osnovi je fastText naučen, da besede predstavi z nizkodimenzionalnimi vektorji, vendar se lahko vektorji dokumentov dobijo s povprečenjem vektorjev besed, ki se v dokumentu nahajajo.

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/03_01_vector_representation_of_documents.ipynb

2.1.7. Uporaba razdalj in podobnosti

Razdaljo med dvema dokumentoma je mogoče izračunati kot razdaljo med njhovima vektorskima predstavitevama. S fastText ali vrečo besed je za želeni dokument mogoče s pomočjo algoritma »najbližji sosedi« poiskati določeno število dokumentov, ki so mu glede na izbrani vektorski prostor najbližje.

Denimo, da obstaja seznam besed, za katere je potrebno ugotoviti, ali določeni dokument dobro opisujejo. To nalogo je mogoče rešiti na način, ki je podoben zgoraj opisanemu. Ker je zaradi uporabe algoritma fastText na voljo vložitev besed, tako vložene besede pa so združene v dokumente, so tako vložene besede kot dokumenti vektorji v istem vektorskem prostoru. Razdalja (podobnosti) med besedami in dokumenti se izračuna s pomočjo kosinusne podobnosti.

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/03_02_distances_and_similarities.ipynb

2.1.8. Odkrivanje skupin in izris kart dokumentov

Dimenzionalnost vektorskih predstavitev dokumentov lahko zmanjšamo na 2, kar nam omogoča prikaz dvodimenzionalne karte dokumentov, na kateri vsaka točka predstavlja dokument. Poleg tega je možno v dvodimenzionalnem prostoru odkriti skupine podobnih dokumentov in vsako skupino na karti označiti z različno barvo. To nam omogoča dober vpogled v celotno množico dokumentov.

Na primeru besedil zadnjih 100 predlogov vladi, ki vsebujejo vsaj 50 znakov, je odkritih 5 besednih skupin, ki so prikazane na Slika 5, in sicer:

- skupina 0 (plačevati, davek, izplačevati, poslovati, kupovati),
- skupina 1 (preprečevanje, zniževanje, zmanjševanje, smrtnost, neučinkovitost),
- skupina 2 (otrok, vrtec, srednješolec, učitelj, šolati),
- skupina 3 (vlada, razpustiti, anarhija, ustava, represija) in
- skupina 4 (aplikacija, dpp, bluetooth, gps, okužen).



Slika 5: Odkrivanje skupin v dokumente predlog vladi (5 skupin)

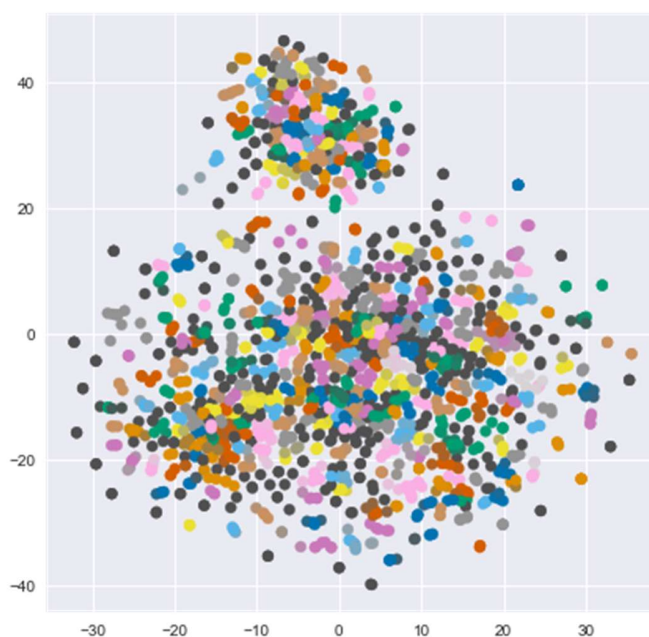
Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/03_03_document_maps.ipynb

2.1.9. Vektorske predstavitve besed

V tem primeru uporabe se vložitve besed dokumentov iz 100 predlogov vladi izračunajo s pomočjo algoritma fastText, ki temelji na nevronskih mrežah in je prednaučen na velikem korpusu dokumentov, pri čemer besede predstavi z nizkodimenzionalnimi vektorji.

Vložitve razdelimo v dve gruči z uporabo algoritma gručenja. Vrhnja, manjša gruča vsebuje glagole (aktivirati, dajati, deti, dodeliti, dosegati, držati, iskati, izdajati, izkazati, izogniti, izvesti, komunicirati, kršiti, lajšati, motivirati, nahajati, nameščati, načeti, obnašati, obveščati idr.).



Slika 6: Gručenje vektorskih predstavitev besed

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/03_04_vector_representation_of_words.ipynb

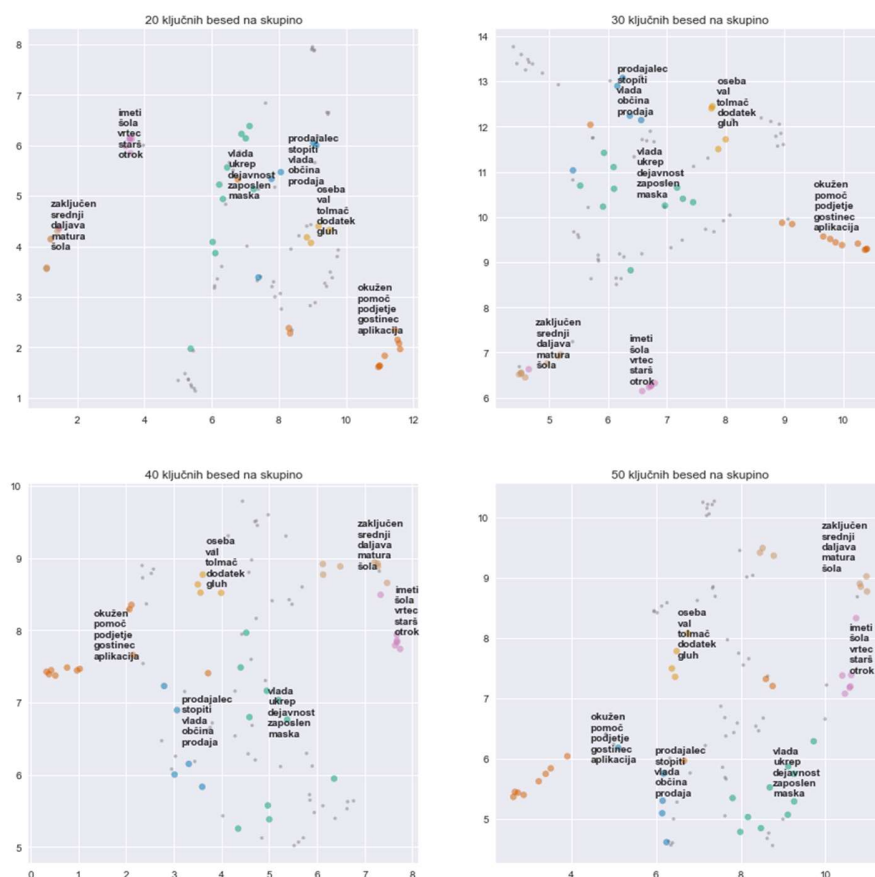
2.1.10. Odkrivanje skupin in izris kart dokumentov na podlagi besed specifičnih za skupine

Dimenzionalnost vektorskih predstavitev dokumentov lahko zmanjšamo na 2, kar nam omogoča prikaz dvodimenzionalne karte dokumentov, na kateri vsaka točka predstavlja dokument. Poleg tega lahko v dvodimenzionalnem prostoru odkrijemo skupine podobnih dokumentov in vsako skupino na karti obarvamo z različno barvo. To nam omogoča dober vpogled v celotno množico dokumentov.

Iz besedil 100 predlogov vladi, ki vsebujejo najmanj 50 besed, s pomočjo TF-IDF ustvarimo vektorske vložitve dokumentov, ki jih nato gručimo v skupinah in za vsako skupino izpišemo

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco

pet ključnih besed pridobljenih prav tako z metodo TF-IDF. Za primerjavo izrišemo štiri dokumentne karte, ki se razlikujejo po tem, koliko ključnih besed za vsako skupino je bilo vključenih v vložitve.



Slika 7: Odkrivanje skupin in izris kart dokumentov na podlagi besed specifičnih za skupine

Povezava do kode: https://github.com/biolab/text-semantics/blob/main/examples/03_05_document_maps_specific.ipynb

2.1.11. Iskanje značilk z uporabo vložitev fastText

Uporaba vložitev fastText omogoča enostavno izračunavanje razdalj in podobnosti med dokumenti in besedami. Kot je bilo prikazano v prejšnjih poglavjih, lahko za vsak dokument identificiramo specifične besede, ki so mu najbolj podobne. Vendar, če je beseda podobna tudi drugim dokumentom, ni za noben dokument specifična. Zaradi tega pri določanju

specifičnih besed upoštevamo tudi podobnost te besede z drugimi dokumenti in na ta način najdemo besede, ki so bližje ciljnemu dokumentu.

V tem primeru so bile za množico zadnjih 100 predlogov vladi, ki vključujejo najmanj 50 besed, izračunane najbolj specifične besede. Uporabljena sta bila dva pristopa: prvi je upošteval vse besede v množici dokumentov kot kandidate za specifično besedo dokumenta, drugi pa le besede znotraj ciljnega dokumenta. Predvideva se, da bodo besede iz korpusa dale boljše rezultate, vendar je tak izračun časovno zahtevnejši.

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_01_specific_words_with_embeddings.ipynb

2.1.12. Iskanje značilk z uporabo obogatitve besed

V prejšnjem primeru so bile specifične besede poiskane s pomočjo vložitev dokumentov, v tem primeru pa je uporabljena metoda obogatitev besed (ang. Word enrichment).

Obogatitev besed izračuna p-vrednost za vsako od besed v množici vseh besed. Nižja p-vrednost pomeni večjo verjetnost, da je beseda značilna za izbrani dokument. Na ta način se določijo značilke, na podlagi teh pa značilne besede z metodo obogatitev besed. V prvem koraku se izračunajo p-vrednosti in FDR (popravljen p-vrednost) za vsako besedo iz množice vseh besed za podani dokument. Nato funkcija vrne zgolj besede, ki imajo p-vrednosti manjše od 1. Te besede so značilke izbranega dokumenta.

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_02_specific_words_with_enrichment.ipynb

2.1.13. Iskanje značilk z uporabo transformacije TF-IDF

V tem primeru se specifične besede v dokumentu določijo z uporabo transformacije TF-IDF, ki uteži besede glede na njihovo pogostost v besedilu. Besede, ki močno zaznamujejo manjšo množico dokumentov, bodo tako imele večjo težo kot take, ki so vseprisotne v korpusu.

Iskanje značilk z uporabo TF-IDF je prikazano na primeru dokumenta »Sprememba stopnje DDV za stanovanjske nepremičnine«. Pet najvišje uteženih besed z metodo TF-IDF prikazuje Tabela 4.

Tabela 4: Pet najpogostejših besed in uteži za primer »Sprememba stopnje DDV za stanovanjske nepremičnine«

Beseda	Utež
površina	0.56
stanovanje	0.37
obdavčen	0.34
cenzus	0.23
250m2	0.23

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_03_specific_words_with_tfidf.ipynb

2.1.14. Iskanje značilk z uporabo metod na grafih besed

Tokrat so značilke dokumenta določene z uporabo metod teorije grafov. Uporabljeni sta metodi TextRank in RAKE. Metodi zgradita graf sopojavitev besed ter na podlagi grafa točkujeta besede in fraze.

Znova na primeru dokumenta »Sprememba stopnje DDV za stanovanjske nepremičnine« s pomočjo metod TextRank in RAKE pridobimo značilke. Rezultati so prikazani na Tabela 5.

Tabela 5: Metode na grafih besed: TextRank in RAKE

Word	TextRank	Score	Word	RAKE	Score
0	obdavčen	0.389836	0	hiše	4.0
1	obdavčitev	0.320819	1	posledično	4.0

2	stanovanje	0.320819	2	površina	4.0
3	površina	0.301328	3	stanovanja	1.0
4	cenzus	0.246738	4	absurdno	1.0

Povezava do kode:

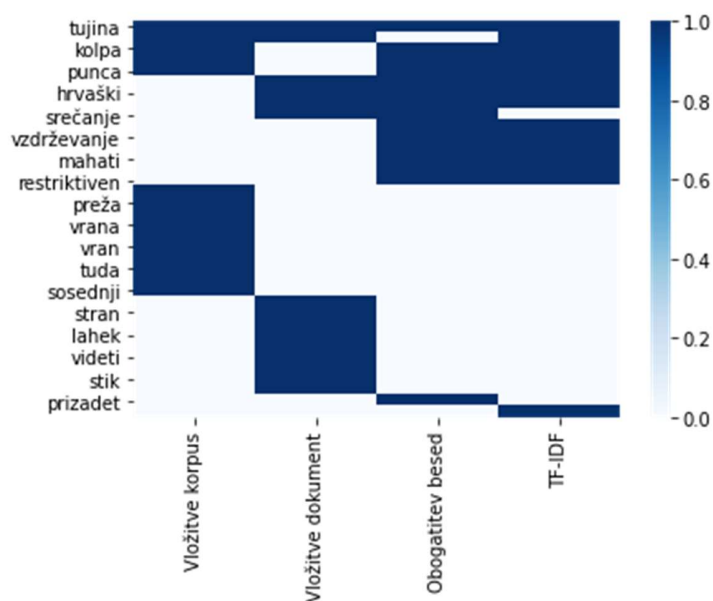
https://github.com/biolab/text-semantics/blob/main/examples/04_04_specific_words_graph_base.ipynb

Primerjava pristopov za specifične besede

Za dokument »Videvanja s partnerjem iz tujine med epidemijo« je narejena primerjalna analiza 15 najbolj specifičnih besed.

- **Vložitve korpusa:** meja, kolpa, regija, val, preža, migracija, tujina, vrana, povezava, vran, ina, punca, tuda, okolica, sosednji
- **Vložitve dokumenta:** meja, tujina, izjema, stran, živ, sam, lahek, dovolj, videti, hrvaški, možen, stik, prehajanje, srečanje, velik
- **Obogatitev besed:** kolpa, mina, vzdrževanje, regija, partner, mahati, razdvojenost, punca, tujina, prehajanje, restriktiven, hrvaški, živ, prizadet, srečanje
- **TF-IDF:** meja, razdvojenost, punca, mahati, kolpa, mina, partner, vzdrževanje, regija, restriktiven, živ, prehajanje, hrvaški, tujina, prepoznati

Specifične besede so prikazane na Slika 8.



Slika 8: Primerjalna analiza različnih pristopov za izbor specifičnih besed

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_05_specific_words_comparison.ipynb

2.1.15. Primerjava pristopov za luščenje značilk na anotiranih besedilih iz revije Prispevki za novejšo zgodovino

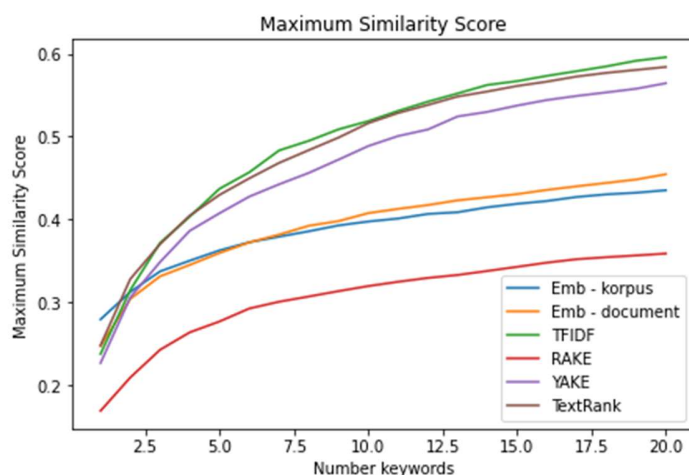
Primerjani so pristopi za iskanje ključnih besed iz besedil revije Prispevki za novejšo zgodovino, ki imajo označene ključne besede. Ključne besede vsakega članka so določili avtorji posameznega članka.

Izvedena je bila primerjalna analiza med petimi metodami za iskanje ključnih besed, in sicer:

- [TF-IDF](#)
- [Metodi z vložitvami:](#)
 - vseh besed v korpusu ali
 - samo besed v posameznih dokumentih
- [RAKE](#)
- [Yake!](#)
- [TextRank](#)

Z vsako metodo je pridobljen seznam ključnih besed, ki je razvrščen po pomembnosti. Povprečna maksimalna podobnost je izračunana za različno število izbranih najbolj pomembnih ključnih besed. Grafi prikazujejo povprečno maksimalno podobnost za poljubno število ključnih besed v intervalu med 1 in 20. Na ta način je prikazana uspešnost metode glede na izbrano število ključnih besed.

Na Slika 9 je narisana graf, ki prikazuje vrednost povprečja podobnosti v odvisnosti od števila izbranih najboljših ključnih besed.



Slika 9: Primerjava pristopov za luščenje ključnih besed na anotiranih besedilih iz revije »Prispevki za novejšo zgodovino«

Rezultati so podobni, kot bi jih dobili, če bi bile za primerjavo med metodami uporabljene mere natančnost (»precision«), priklic (»recall«) ali F1.

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_06b_specific_words_comparison_ctch_with_max_similarity.ipynb

2.1.16. Primerjava pristopov za luščenje značilk na označenih besedilih iz revije Elektrotehniški vestnik

Primerjani so pristopi za luščenje ključnih besed iz besedil revije Elektrotehniški vestnik, ki imajo označene ključne besede. Ključne besede vsakega članka so določili avtorji posameznega članka.

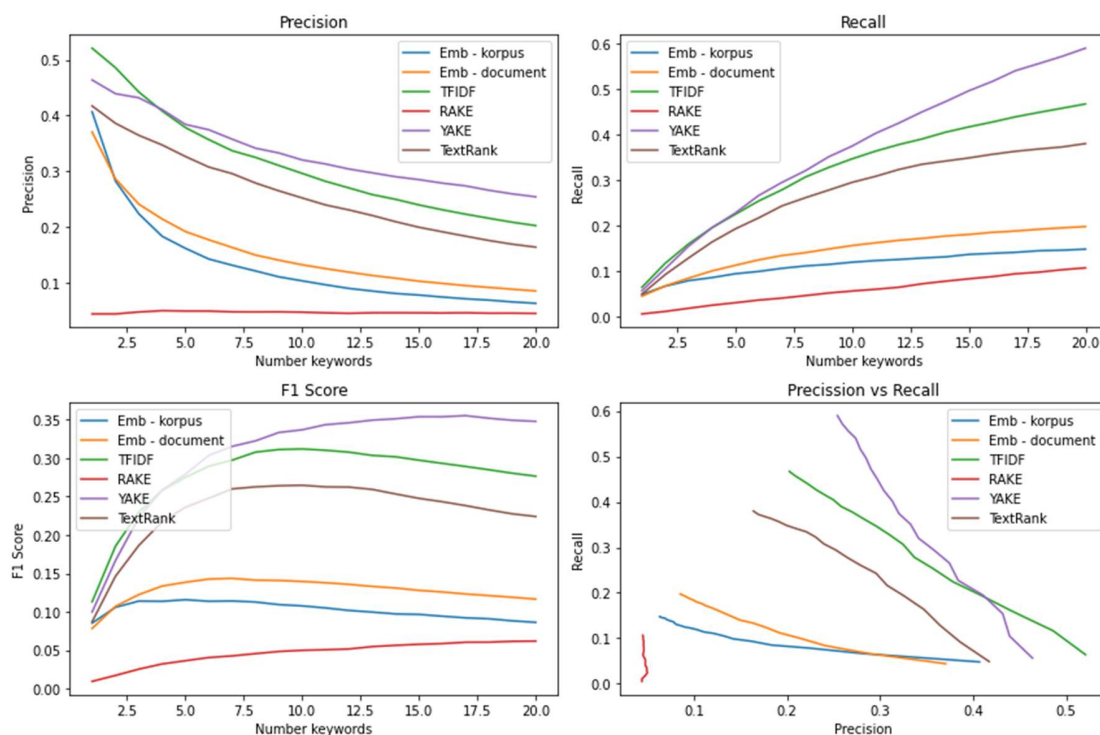
Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco

Izvedena je bila primerjalna analiza med petimi metodami za luščenje ključnih besed, in sicer:

- [TF-IDF](#)
- [Metodi z vložitvami:](#)
 - vseh besed v korpusu ali
 - samo besed v posameznih dokumentih
- [RAKE](#)
- [Yake!](#)
- [TextRank](#)

Za vsako metodo je izračunana povprečna natančnost, priklic ter mera F1.

Rezultati so prikazani na Slika 10. Iz grafov lahko sklepamo, da se na primeru člankov iz revije Elektrotehniški vestnik najboljše obnese metodi TF-IDF in YAKE!, sledi pa metoda TextRank. Podobno se obnese metodi z vložitvami, najslabše pa se obnese metoda RAKE.



Slika 10: Primerjava pristopov za luščenje ključnih besed na anotiranih besedilih iz revije »Elektrotehniški vestnik«

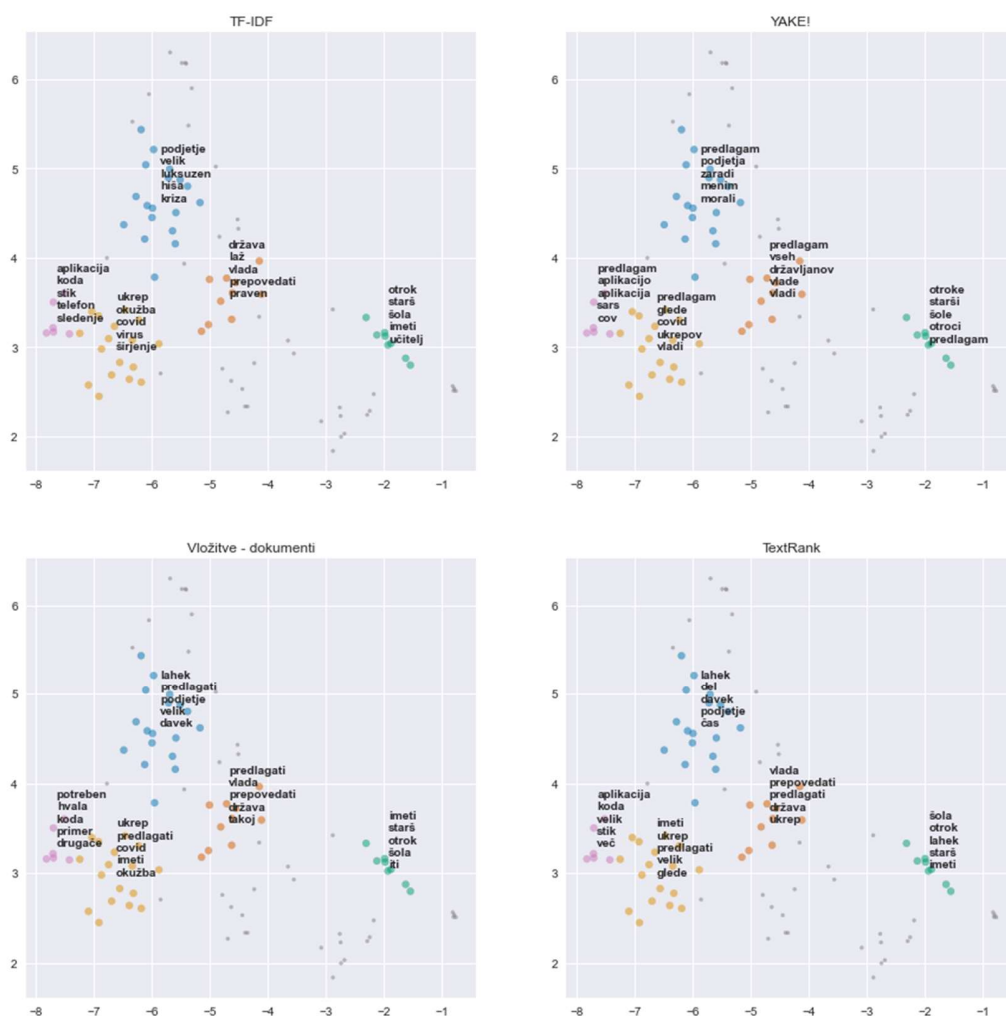
Povezava do kode:

<https://github.com/biolab/text->

[semantic/blob/main/examples/04_07_specific_words_comparison_el_vestnik.ipynb](https://github.com/biolab/text-semantic/blob/main/examples/04_07_specific_words_comparison_el_vestnik.ipynb)

2.1.17. Odkrivanje skupin in izris kart dokumentov iz revije Elektrotehniški vestnik

Dokumenti iz prejšnjega razdelka so za vsako metodo prikazani v dvodimenzionalnem prostoru ter razporejeni v skupine. Pri opisu skupin so upoštevane le tiste ključne besede, ki se najbolj pogosto pojavijo pri dokumentih v skupini. Rezultati so prikazani na Slika 11.



Slika 11: Razlaga izraznih kart dokumentov

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_08_document_maps_explanation.ipynb

2.1.18. Primerjava pristopov za luščenje značilk na primeru korpusa Schutz 2008

V tem razdelku je predstavljena primerjava pristopov za luščenje ključnih besed iz nabora besedil Schutz 2008. Korpus sestavlja 1.231 člankov s področja medicine (PubMed Central), pri čemer so ključne besede posameznega članka določili avtorji člankov.

Izvedena je bila primerjalna analiza med petimi metodami za luščenje ključnih besed, in sicer:

- [TF-IDF](#)
- [Metodi z vložitvami](#): samo na besedah v dokumentih
- [RAKE](#)
- [Yake!](#)
- [TextRank](#)

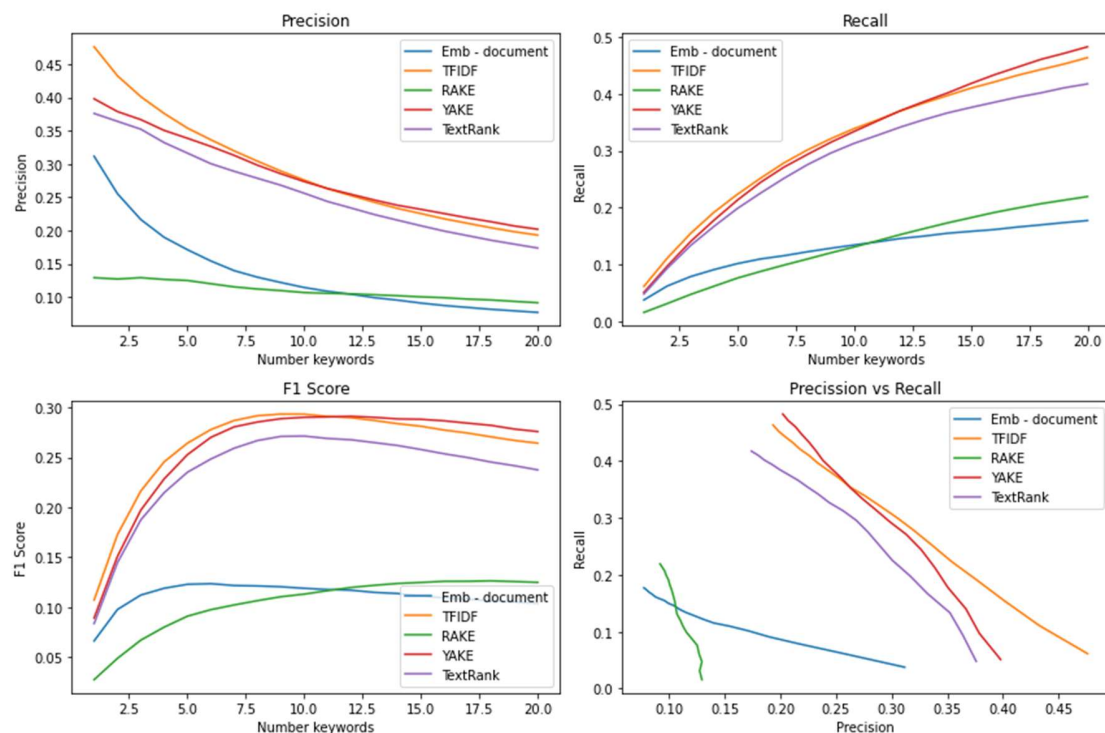
Za vsako metodo je izračunana povprečna natančnost, priklic ter mera F1.

Z vsako metodo so pridobili seznam ključnih besed, ki je razvrščen po pomembnosti. Vse tri mere so izračunane za število ključnih besed v intervalu med 1 in 20. Na ta način se vidi, kako uspešna je metoda glede na izbrano število ključnih besed.

Za vsako od mer je izrisan graf, ki prikazuje vrednost mere v odvisnosti od števila izbranih najboljših ključnih besed. Četrty graf prikazuje natančnost in priklic. V tem grafu ima metoda krivuljo iz večih točk. Vsaka od točk predstavlja natančnost in priklic za različno število izbranih ključnih besed. Metoda s krivuljo bližje zgornjemu desnemu kotu je boljša.

Iz grafov na Slika 12 lahko sklepamo, da se na primeru člankov podatkovnega nabora Schutz 2008 najboljše obnese metodi TF-IDF in YAKE!. Sledi TextRank. Najslabše se obnese metoda RAKE.

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 12: Primerjava pristopov za luščenje ključnih besed na izhodiščnih podatkih Schutz 2008

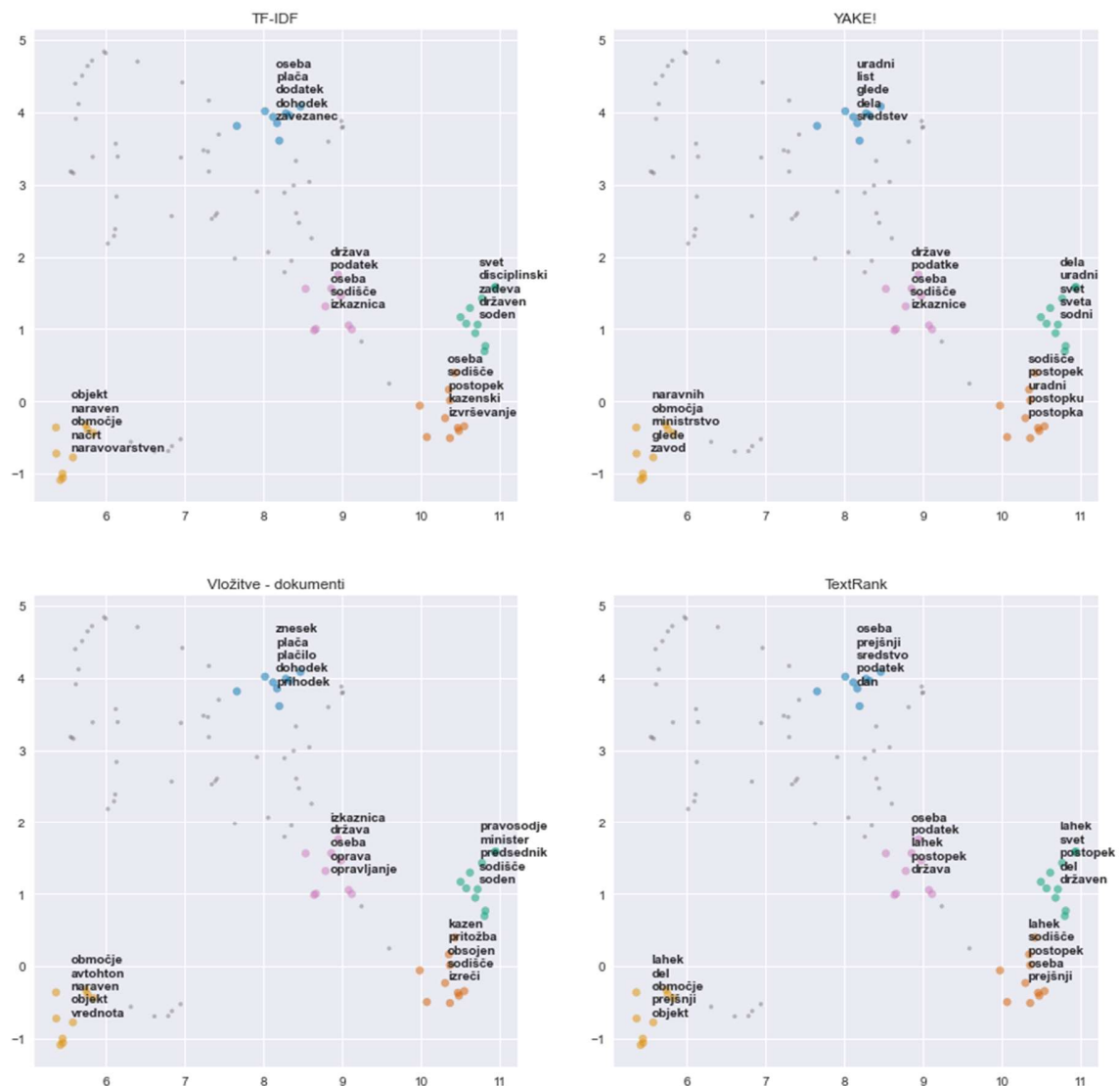
Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_08_specific_words_comparison_schutz2008.ipynb

2.1.19. Odkrivanje skupin in izris kart dokumentov

Dokumenti iz prejšnjega razdelka so za vsako metodo prikazani v dvodimenzionalnem prostoru ter razporejeni v skupine. Pri opisu skupin so upoštevane le tiste ključne besede, ki se najbolj pogosto pojavijo pri dokumentih v skupini. Rezultati so prikazani na Slika 13.

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 13: Razlaga izraznih kart dokumentov

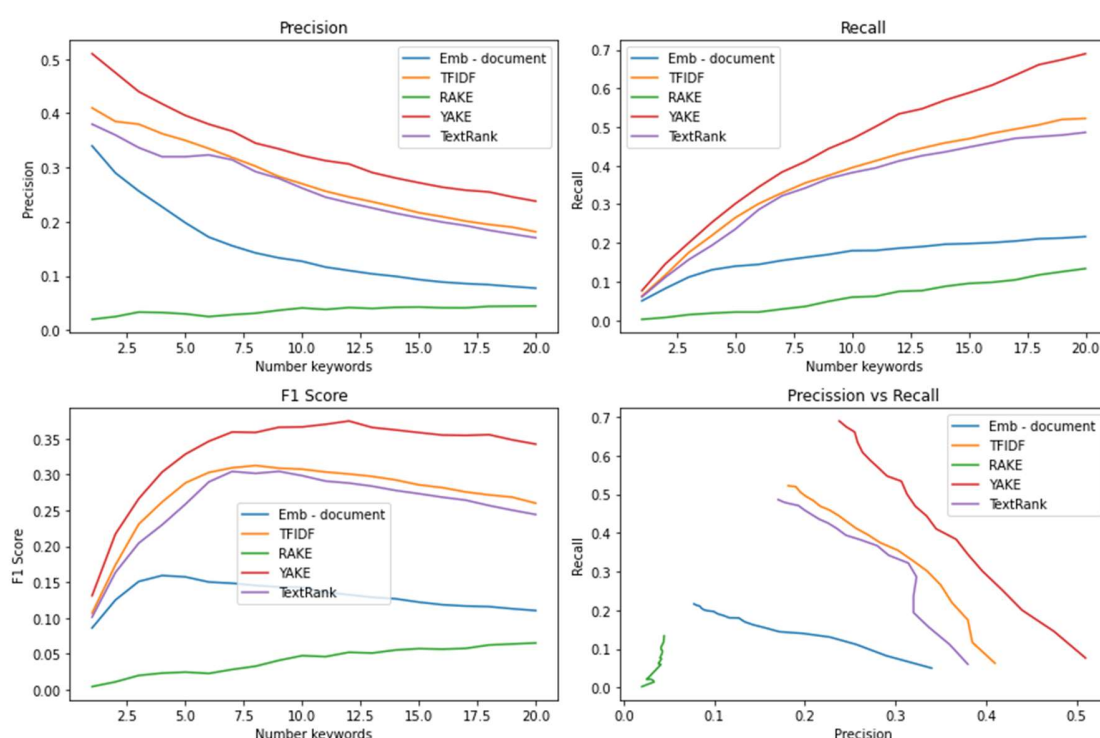
Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_08b_document_maps_explanation-laws.ipynb

2.1.20. Primerjava pristopov za luščenje značilnk na primeru korpusa SemEval

V tem razdelku je predstavljena primerjava pristopov za luščenje ključnih besed iz nabora besedil SemEval, ki vsebuje 244 člankov o računalništvu s portala ACM, pri čemer ključne besede podajo avtorji člankov, z vključitvijo funkcionalnosti transformacij in brez nje.

Rezultati brez vključitev funkcionalnosti transformacij so prikazani na Slika 14.

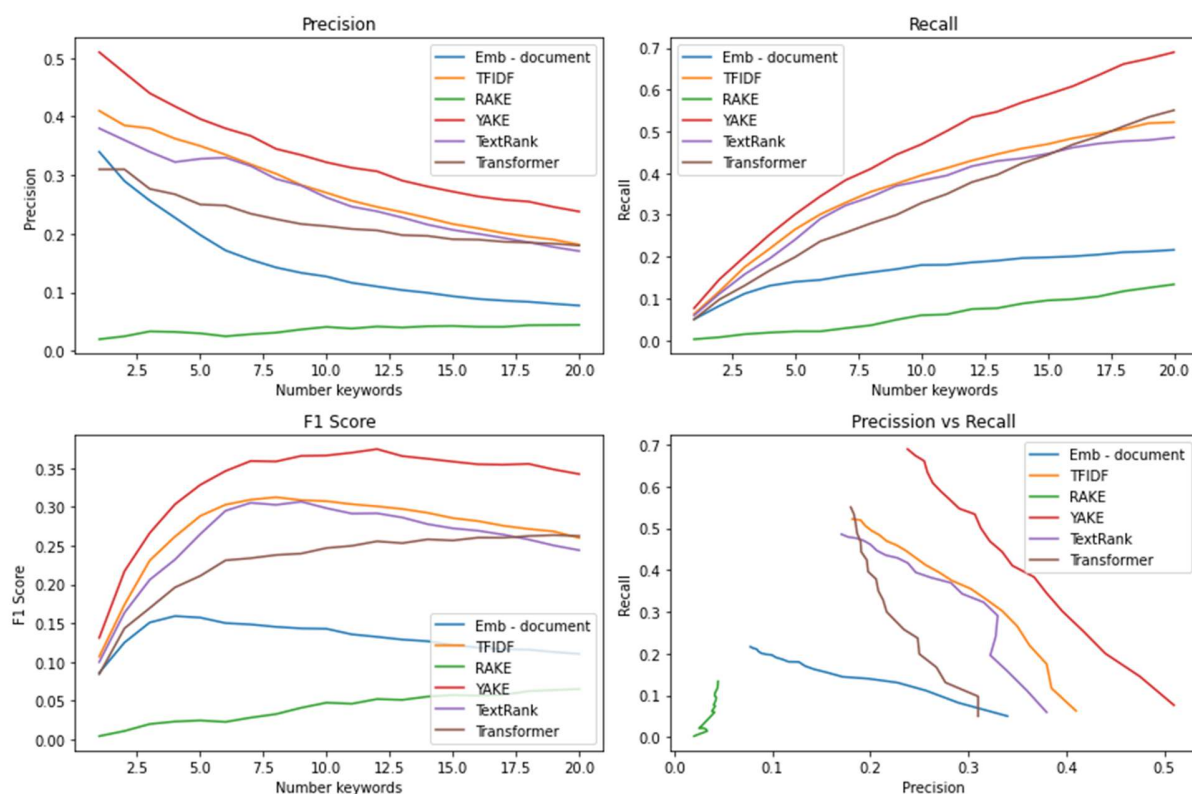


Slika 14: Primerjava pristopov za luščenje ključnih besed na izhodiščnih podatkih SemEval (brez transformacij)

Uporabljena je bila še metoda RoBERTa, ki spada med skupino transformacijskih metod (»transformers«), ki temeljijo na vektorskih vložitvah dokumentov in besed. Ta skupina metod je trenutno ena izmed najbolj uspešnih metod pri različnih aplikacijah na področju obdelave naravnega jezika.

Rezultati z vključitvijo transofrmerja RoBERTa so prikazani na Slika 15.

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 15: Primerjava pristopov za luščenje ključnih besed na izhodiščnih podatkih SemEval (s transformejem RoBERTa)

Iz grafov lahko sklepamo, da se na primeru člankov podatkovne zbirke SemEval najbolj obnese metoda YAKE!. Sledita TF-IDF in Text Rank. Razlika med YAKE in TF-IDF je na tem korpusu večja kot pri korpusu Schutz2008. RoBERTa deluje boljše kot fastText, vendar pa ne boljše kot YAKE, TextRank in TF-IDF. Prav tako je opazno, da so besede, ki jih odkrijeta metodi na podlagi vektorskih vložitev, take, da opisujejo bolj širše področje, na katero se dokument nanaša, kar je v določenih primerih zaželeno.

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_09_specific_words_comparison_semeval.ipynb

https://github.com/biolab/text-semantics/blob/main/examples/04_09b_specific_words_comparison_semeval_including_transformers.ipynb

2.1.21. Primerjava pristopov za luščenje značilk na primeru povzetkov člankov, ki vsebujejo besedo "longevity"

V tem razdelku je predstavljena primerjava pristopov za luščenje ključnih besed iz nabora povzetkov člankov zbirke PubMed, ki vsebujejo ključno besedo "Longevity", pri čemer so ključne besede posameznega članka določili avtorji člankov.

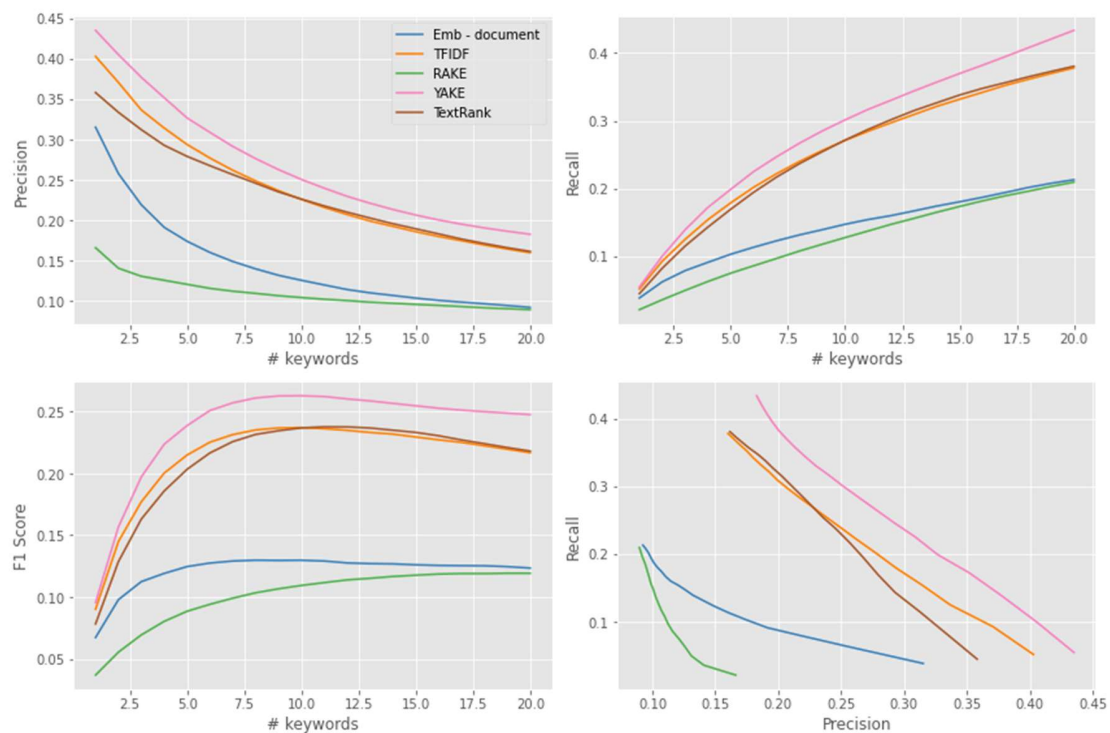
Izvedena je bila primerjalna analiza med 5 pristopi za luščenje ključnih besed brez lematizacije, in sicer:

- [TF-IDF](#)
- [Metodi z vložitvami](#): samo na besedah v dokumentih
- [RAKE](#)
- [Yake!](#)
- [TextRank](#)

Izračunane ključne besede so za vsak članek primerjane s ključnimi besedami, ki so jih označili avtorji članka. Za vsako metodo je izračunana povprečna natančnost, priklic ter mera F1.

Rezultati iskanja ključnih besed na povzetkih člankov brez lematizacije so prikazani na Slika 16. Za vsako od mer je izrisan graf, ki prikazuje vrednost mere v odvisnosti od števila izbranih najboljših ključnih besed. Četrty graf prikazuje natančnost in priklic. V tem grafu ima metoda krivuljo iz večih točk. Vsaka od točk predstavlja natančnost in priklic za različno število izbranih ključnih besed. Metoda s krivuljo bližje zgornjemu desnemu kotu je boljša.

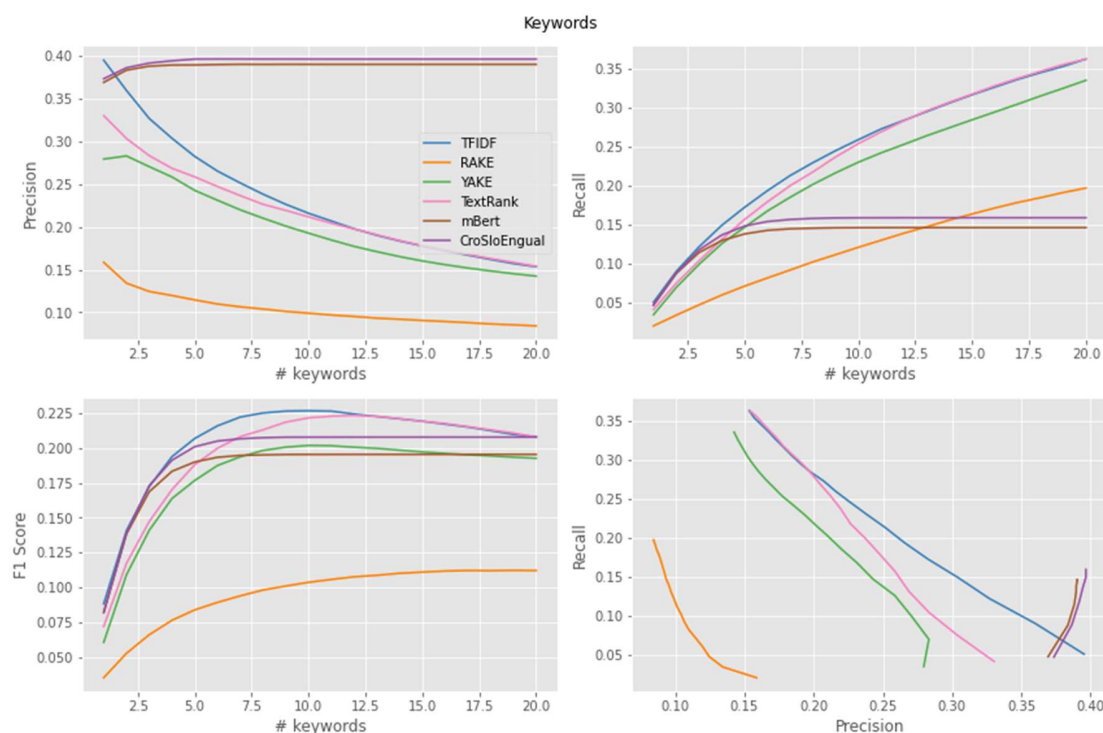
Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 16: Primerjava pristopov za luščenje ključnih besed na povzetkih člankov s ključno besedo "Longevity" (brez lematizacije)

Rezultati iskanja ključnih besed na povzetkih člankov s ključno besedo »Longevity« z uporabo lematizacije so prikazani na Slika 17.

Iz grafov je mogoče sklepati, da se na primeru člankov najbolj obnese metoda YAKE!. Sledita TF-IDF in Text Rank. Razlika med YAKE in TF-IDF je na tem korpusu večja kot pri korpusu Schutz2008.



Slika 17: Primerjava pristopov za luščenje ključnih besed na povzetkih člankov s ključno besedo "Longevity" (z lematizacijo)

Povezava do kode:

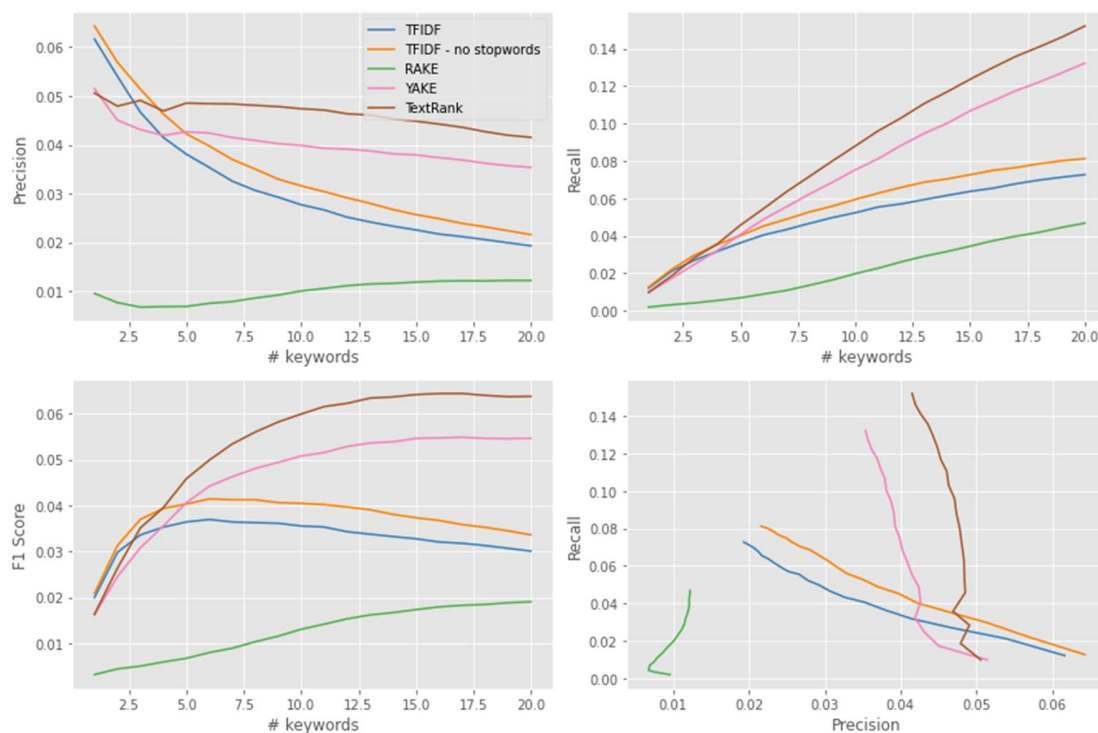
https://github.com/biolab/text-semantics/blob/main/examples/04_10_specific_words_comparison_longevity.ipynb
https://github.com/biolab/text-semantics/blob/main/examples/04_10_specific_words_comparison_longevity_lematizer.ipynb

2.1.22. Primerjava nelematiziranih pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Longevity" - not lemmatized

V tem zvezku predstavljamo primerjavo nelematiziranih pristopov za luščenje ključnih besed iz nabora povzetkov člankov s ključno besedo "Longevity" v zbirki PubMed.

Na Slika 18 je izrisan po en graf za vsako od mer - graf, ki prikazuje vrednost mere v odvisnosti od števila izbranih najboljših ključnih besed. Četrty graf prikazuje natančnost in priklic na

enem grafu. V tem grafu ima metoda krivuljo iz večih točk. Vsaka od točk predstavlja natančnost in priklic za različno število izbranih ključnih besed. Metoda, katere krivulja je bližje zgornjemu desnemu kotu, je boljša.



Slika 18: Primerjava nelematiziranih pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Longevity" - not lemmatized

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_11_keyphrases_comparison_longevity.ipynb

2.1.23. Primerjava pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Covid-19"

V tem razdelku je predstavljena primerjava pristopov za določanje ključnih besed iz nabora povzetkov člankov zbirke PubMed, ki vsebujejo ključno besedo "Covid-19", pri čemer so ključne besede posameznega članka določili avtorji člankov.

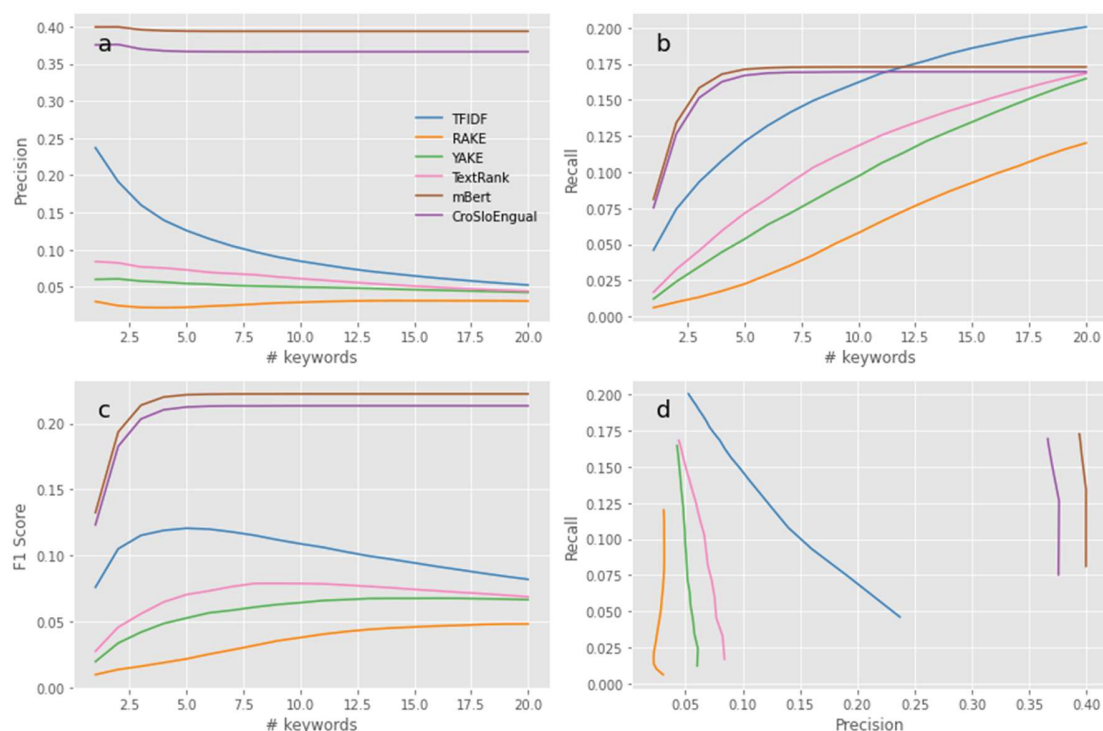
Izvedena je bila primerjalna analiza med 5 pristopi za iskanje ključnih besed brez lematizacije, in sicer:

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco

- [TF-IDF](#)
- [RAKE](#)
- [Yake!](#)
- [TextRank](#)
- [mBERT](#)
- [CroSloEngual](#)

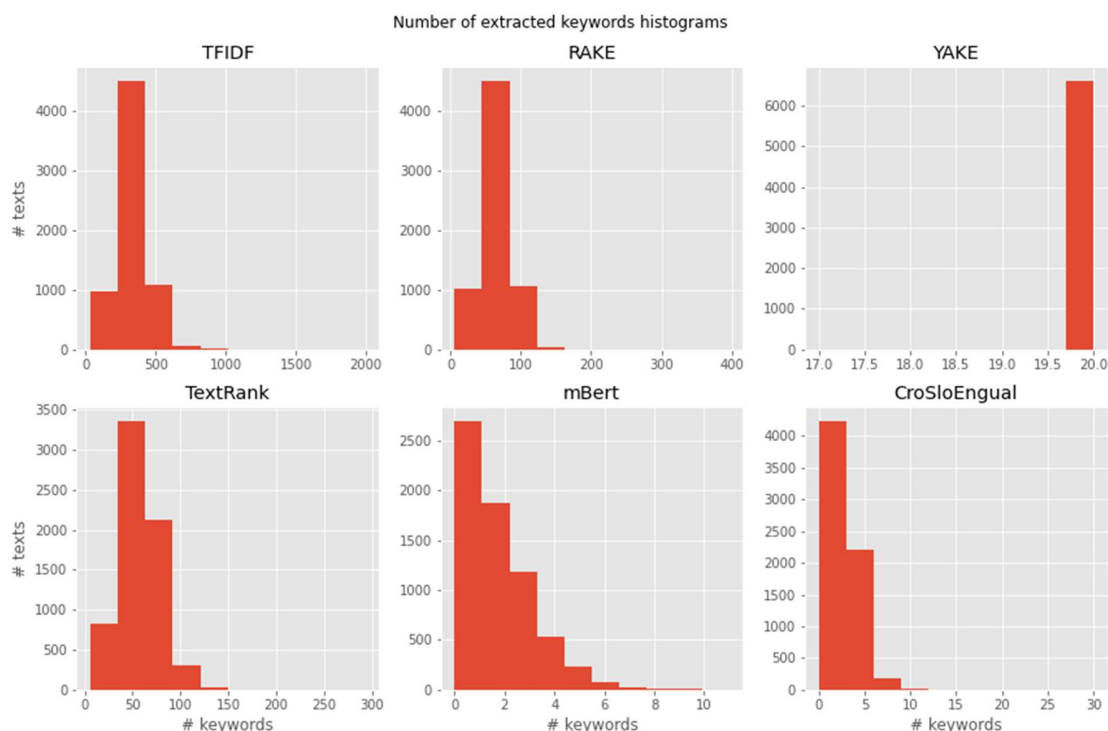
Izračunane ključne besede so za vsak članek primerjane s ključnimi besedami, ki so jih označili avtorji članka. Za vsako metodo je izračunana povprečna natančnost, priklic ter mera F1.

Rezultati iskanja ključnih besed na povzetkih člankov brez lematizacije so prikazani na Slika 169. Za vsako od mer je izrisan graf, ki prikazuje vrednost mere v odvisnosti od števila izbranih najboljših ključnih besed. Četrti graf prikazuje natančnost in priklic. V tem grafu ima metoda krivuljo iz več točk. Vsaka od točk predstavlja natančnost in priklic za različno število izbranih ključnih besed. Metoda s krivuljo bližje zgornjemu desnemu kotu je boljša.



Slika 19: Primerjava pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Covid-19"

Histogrami na Slika 20 prikazujejo število določenih ključnih besed za besedila povezana s Covid-19.



Slika 20 Histogrami za prikaz izluščenih ključnih besed iz število besedil povezanih na Covid 19

Prav tako je podanih nekoliko različnih primerov besedil s tematiko Covid 19, za katere je bila izvedena primerjava ključnih besed članka, ki jih je določil avtor, ter ključnih besed, ki jih je določil algoritem.

Naslov originalnega besedila: "Assessment of antiphospholipid antibodies and calprotectin as biomarkers for discriminating mild from severe COVID-19."

Originalno besedilo: "To explore the association of thrombo-inflammatory biomarkers with severity in coronavirus disease (COVID-19), we measured antiphospholipid antibodies (aPL) and calprotectin in sera of COVID-19 patients. Anticardiolipin antibodies (aCL) and anti- β 2-glycoprotein I antibodies were measured using enzyme-linked immunosorbent assay (ELISA) and multiplex flow immunoassay (MFIA) in hospitalized COVID-19 patients (N = 105) and healthy controls (N = 38). Anti-phosphatidylserine/prothrombin antibodies, calprotectin, and C-reactive protein (CRP) levels were also measured. We assessed the potential correlation between calprotectin levels and various laboratory parameters that were measured during the hospitalization period. After stratifying COVID-19 patients into two groups by their oxygenation status or acute respiratory distress syndrome presentation, the discriminatory performance of each biomarker was evaluated. A high proportion of COVID-19 patients (29.5%, 31/105) had low aCL IgM titers that were detectable by ELISA but mostly

below the detection limit of MFIA. Calprotectin levels in severe groups of COVID-19 were significantly higher than those in non-severe groups, while CRP levels revealed no significant differences. Serum calprotectin levels showed strong to moderate degree of correlation with other routinely used parameters including peak levels of CRP, ferritin, procalcitonin, BUN, and neutrophil-to-lymphocyte ratio, but a negative correlation with minimal lymphocyte count and CD4+ T cells. The discriminatory performance was highest for calprotectin in discriminating severe groups of COVID-19. Serum calprotectin levels were significantly elevated in severe COVID-19 cases. The prevalence of clinically significant aPL did not differ. The link between calprotectin and inflammatory pathway in COVID-19 may help improve the management and outcomes of COVID-19 patients."

Ključne besede samega avtorja: "COVID-19, anticardiolipin antibodies, antiphospholipid antibodies, calprotectin, severity"

TFIDF: calprotectin, calprotectin level, serum calprotectin level, severe group covid19, serum calprotectin, discriminatory performance, mfia, level, acl, apl

RAKE: link immunosorbent assay, multiplex flow immunoassay, various laboratory parameter, minimal lymphocyte count, measure use enzyme, discriminating severe group, clinically significant apl, measure antiphospholipid antibody, crp level reveal, serum calprotectin level

YAKE: measure antiphospholipid antibody, coronavirus disease, calprotectin, explore the association, association of thrombo-inflammatory, severity in coronavirus, calprotectin level, level, patient, antibody

TextRank: N, healthy control, COVID-19 patient, hospitalize COVID-19 patient, calprotectin level, Serum calprotectin level, severe COVID-19 case, non-severe group, COVID-19, severe group

mBert: calprotectin, covid-19

CroSloEngual: calprotectin, coronavirus, covid-19

Več primerov si lahko pogledate v zvezku na spodnji povezavi.

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_11_keyphrases_comparison_longevity_lematizer-covid19.ipynb

Primerjava pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Longevity"

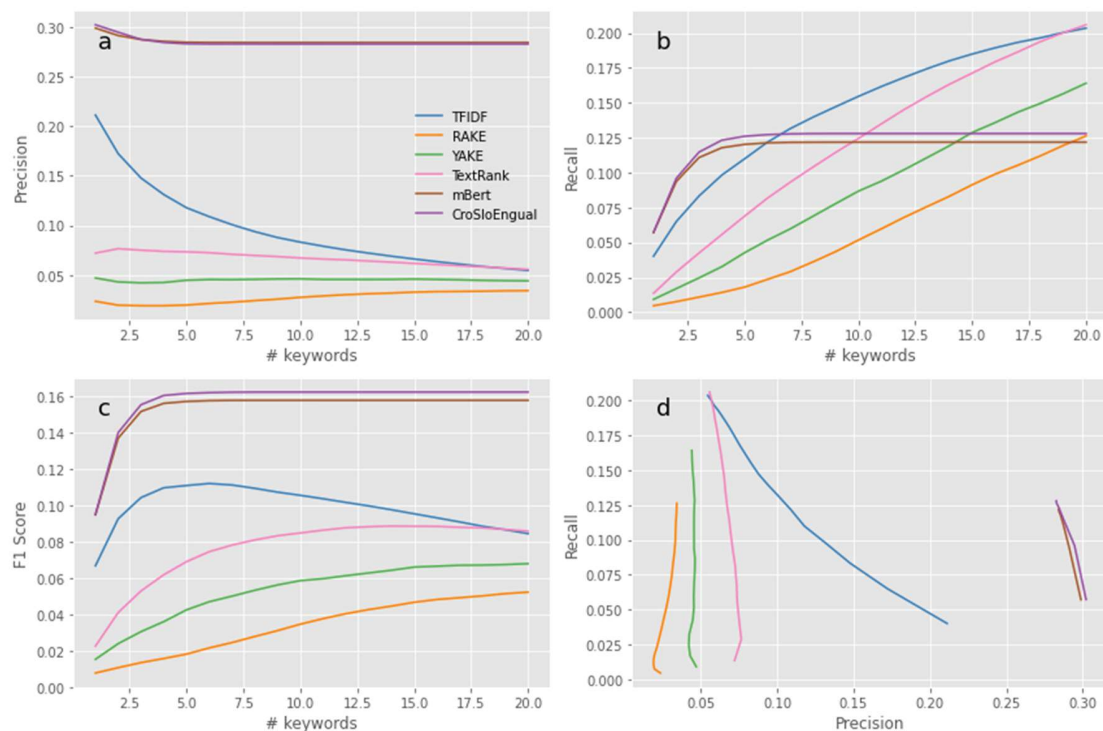
V tem zvezku predstavljamo primerjavo pristopov za luščenje ključnih besed iz nabora povzetkov člankov s ključno besedo "Longevity" v zbirki PubMed.

Izvedena je bila primerjalna analiza med 5 pristopi za luščenje ključnih fraz s ključno besedo »Longevity«, in sicer:

- [TF-IDF](#)
- [RAKE](#)
- [Yake!](#)
- [TextRank](#)
- [mBERT](#)
- [CroSloEngual](#)

Na Slika 21 je izrisan po en graf za vsako od mer - graf, ki prikazuje vrednost mere v odvisnosti od števila izbranih najboljših ključnih besed. Četrty graf prikazuje natančnost in priklic na enem grafu. V tem grafu ima metoda krivuljo iz več točk. Vsaka od točk predstavlja natančnost in priklic za različno število izbranih ključnih besed. Metoda, katere krivulja je bližje zgornjemu desnemu kotu, je boljša.

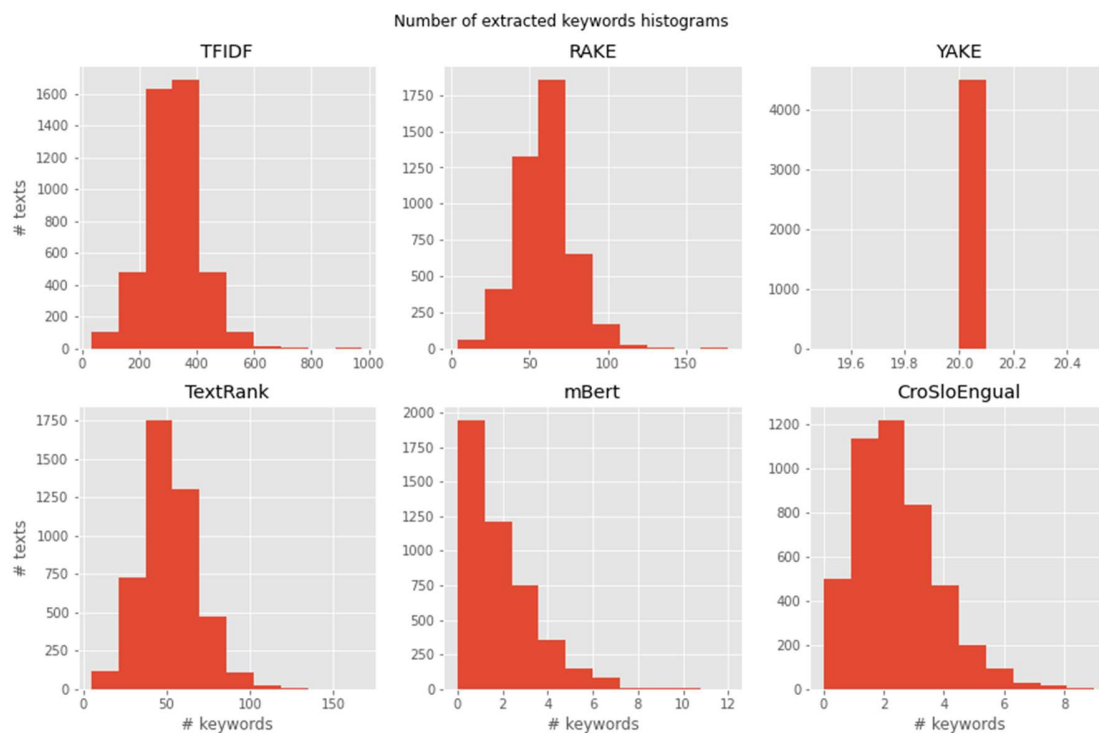
Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 21: Primerjava pristopov za luščenje ključnih fraz na povzetkih člankov s ključno besedo "Longevity"

Na histogramih na Slika 22 je prikazano število izluščenih ključnih besed za besedila, povezana s ključno besedo »Longevity«.

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 22: Histogrami za prikaz izluščenih ključnih besed iz besedil, povezanih s ključno besedo "Longevity"

Prav tako so prikazani različni primeri besedil v zvezi s ključno besedo »Longevity«, za katere primerjamo ključne besede avtorja in ključne besede algoritme za luščenje.

Naslov originalnega besedila: The frailty index outperforms DNA methylation age and its derivatives as an indicator of biological age.

Originalno besedilo: The measurement of biological age as opposed to chronological age is important to allow the study of factors that are responsible for the heterogeneity in the decline in health and function ability among individuals during aging. Various measures of biological aging have been proposed. Frailty indices based on health deficits in diverse body systems have been well studied, and we have documented the use of a frailty index (FI34) composed of 34 health items, for measuring biological age. A different approach is based on leukocyte DNA methylation. It has been termed DNA methylation age, and derivatives of this metric called age acceleration difference and age acceleration residual have also been employed. Any useful measure of biological age must predict survival better than chronological age does. Meta-analyses indicate that age acceleration difference and age acceleration residual are significant predictors of mortality, qualifying them as indicators of biological age. In this article, we compared the measures based on DNA methylation with FI34. Using a well-studied cohort, we assessed the efficiency of these measures side by side in predicting

mortality. In the presence of chronological age as a covariate, FI34 was a significant predictor of mortality, whereas none of the DNA methylation age-based metrics were. The outperformance of FI34 over DNA methylation age measures was apparent when FI34 and each of the DNA methylation age measures were used together as explanatory variables, along with chronological age: FI34 remained significant but the DNA methylation measures did not. These results indicate that FI34 is a robust predictor of biological age, while these DNA methylation measures are largely a statistical reflection of the passage of chronological time.

Ključne besede samega avtorja: Aging, Biological age, DNA methylation, Frailty, Mortality

TFIDF: fi34, dna methylation, methylation, biological age, dna, age, age acceleration, measure, chronological, acceleration

RAKE: diverse body system, leukocyte dna methylation, age acceleration residual, age acceleration difference, dna methylation age, dna methylation measure, measure biological age, frailty index base, fi34 remain significant, dna methylation

YAKE: DNA methylation, DNA methylation age, age, biological age, DNA, DNA methylation measure, methylation, age acceleration, methylation age, biological

TextRank: DNA methylation age, chronological age, biological age, age acceleration residual, age acceleration difference, DNA methylation, leukocyte DNA methylation, biological age, chronological time, significant predictor

mBert: dna methylation, frailty index, age acceleration, dna methylation age

CroSloEngual: biological age, chronological age, dna methylation, frailty

Več primerov si pogledate v zvezku na spodnji povezavi.

Povezava do kode:

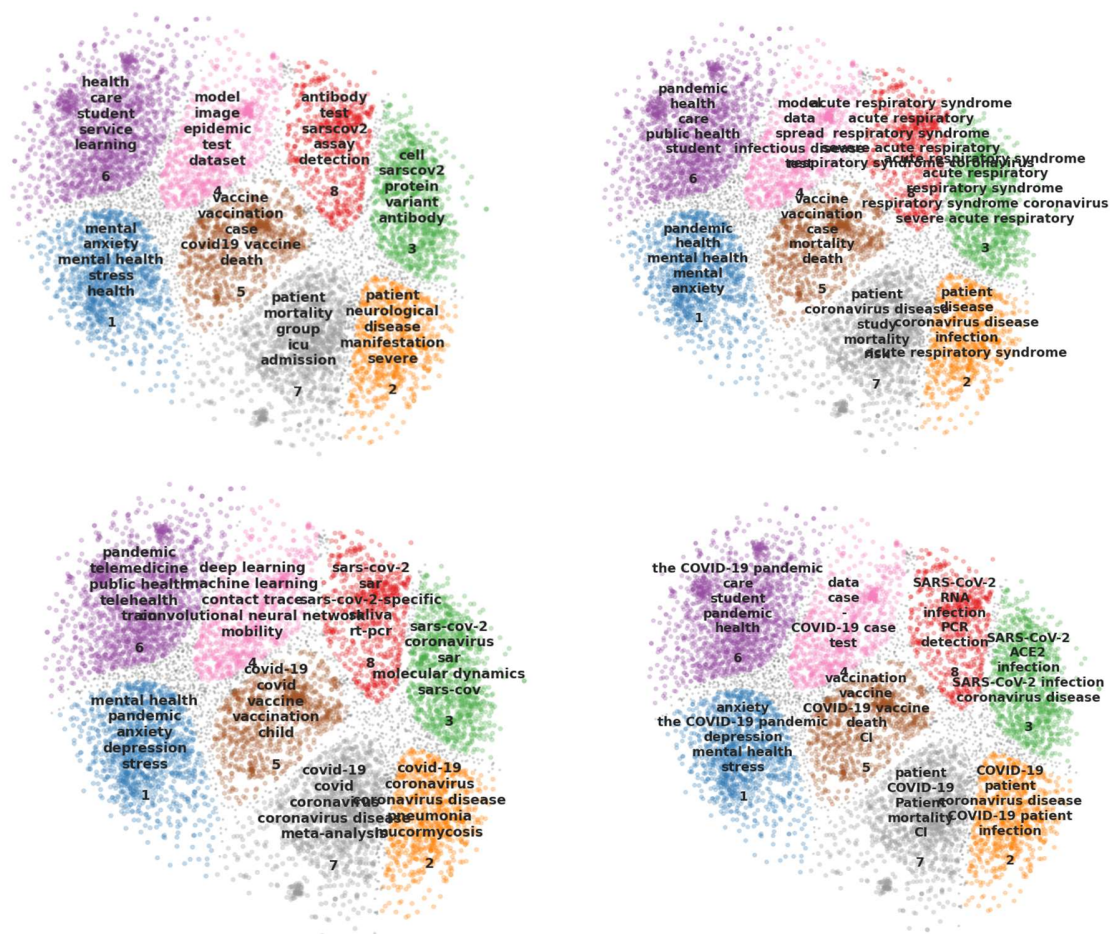
https://github.com/biolab/text-semantics/blob/main/examples/04_11_keyphrases_comparison_longevity_lematizer.ipynb

2.1.24. Vizualizacija gruč vložitev ključnih fraz AIIM člankov o longevity in Covid-19

V tem razdelku je predstavljena primerjava pristopov za vizualizacije gruč vložitev ključnih fraz AIIM člankov o Covid-19 in Longevity. Vložitve so narejene s pomočjo metode t-SNE. Za gručenje po ključnih fraz so uporabili metode: TF-IDF, YEKE, Transformers in TextRank

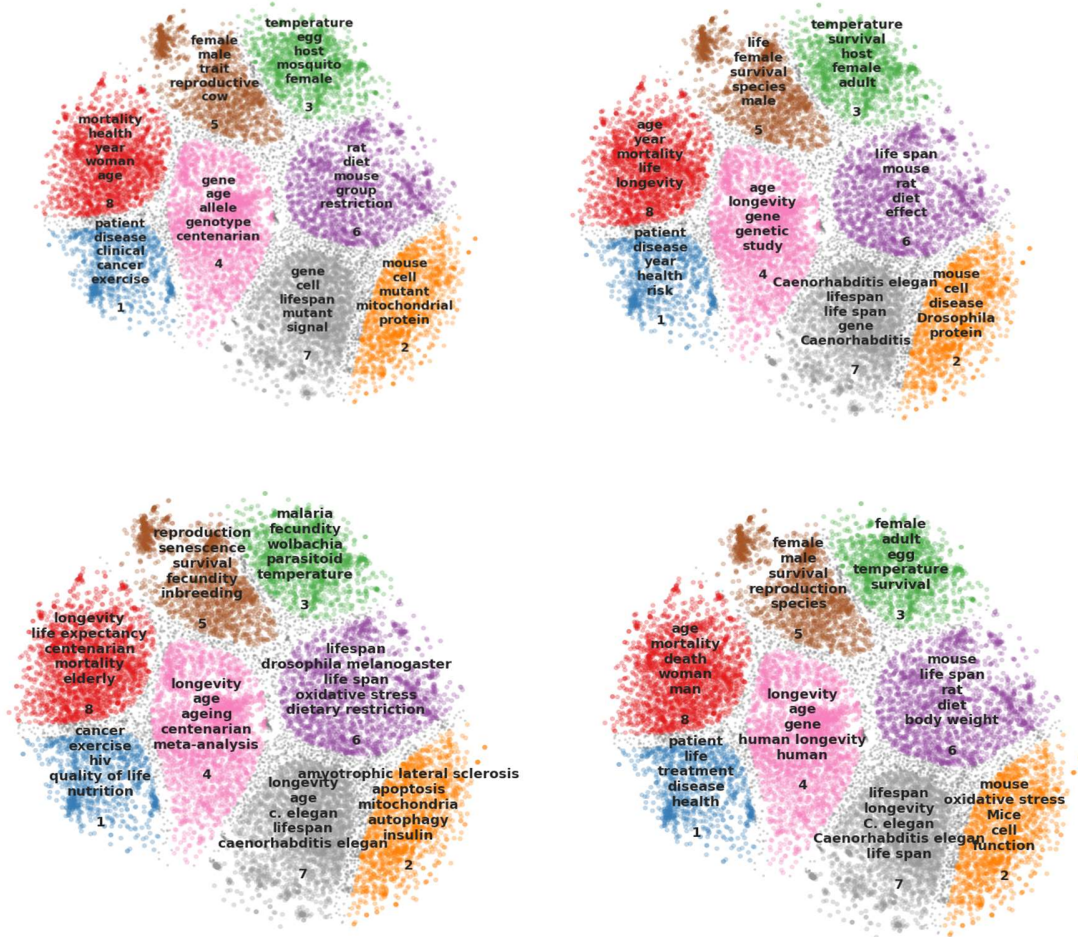
Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco

Na Slika 23 je prikazana vizualizacija gruč AIIM člankov v zvezi s Covid 19, na Slika 24 pa je prikazana vizualizacija gruč AIIM člankov v o Longevity.



Slika 23: Gruče ključnih fraz AIIM člankov o Covid 19. Zgoraj levo je TF-IDF, zgoraj desno je YAKE, spodaj levo je BERT transformer in spodaj desno je TextRank

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 24: Gruče ključnih fraz AIIM člankov o Longevity. Zgoraj levo je TF-IDF, zgoraj desno je YAKE, spodaj levo je BERT transformer in spodaj desno je TextRank

Povezava do kode:

https://github.com/biolab/text-semantics/blob/main/examples/04_11_longevity_visualizations_aiim_covid.ipynb

https://github.com/biolab/text-semantics/blob/main/examples/04_11_longevity_visualizations_aiim_phrases.ipynb

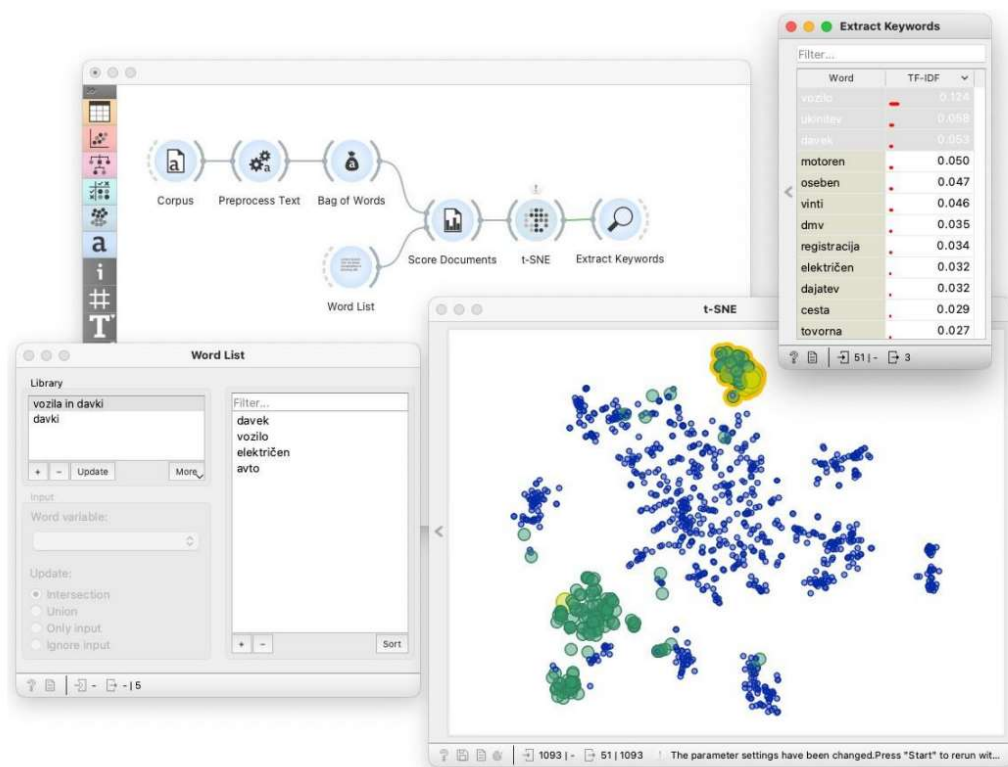
https://github.com/biolab/text-semantics/blob/main/examples/04_11_longevity_visualizations_aiim_words.ipynb

2.1.25. Primer v programskem paketu Orange

Vsi zgoraj popisani primeri uporabe v pilotnem projektu so implementirani v programskem paketu Orange. Najbolj pogosti gradniki, ki se uporabljajo na področju semantične in drugih naprednih analiz besedil, so:

- Corpus - prebere dokumente s spleta ali datotečnega sistema,
- Preprocess Text – predobdelava,
- Bag of Words – predobdelava,
- Score Documents – ocena podobnosti dokumentov v zbirki,
- Word List - seznam pojmov,
- t-SNE - vizualizacija podobnosti med dokumenti,
- Extract Keywords - določanje značilnih besed množici dokumentov.

Na Slika 25 je prikazan potek dela za luščenje ključnih besed v besedilih sistema Predlogi Vladi RS, narejen v orodju Orange.



Slika 25: Potek dela v Orange. Primer uporabe: Določanje ključnih besed besedilom iz sistema Predlagaj Vladi

2.1.25.1. Corpus

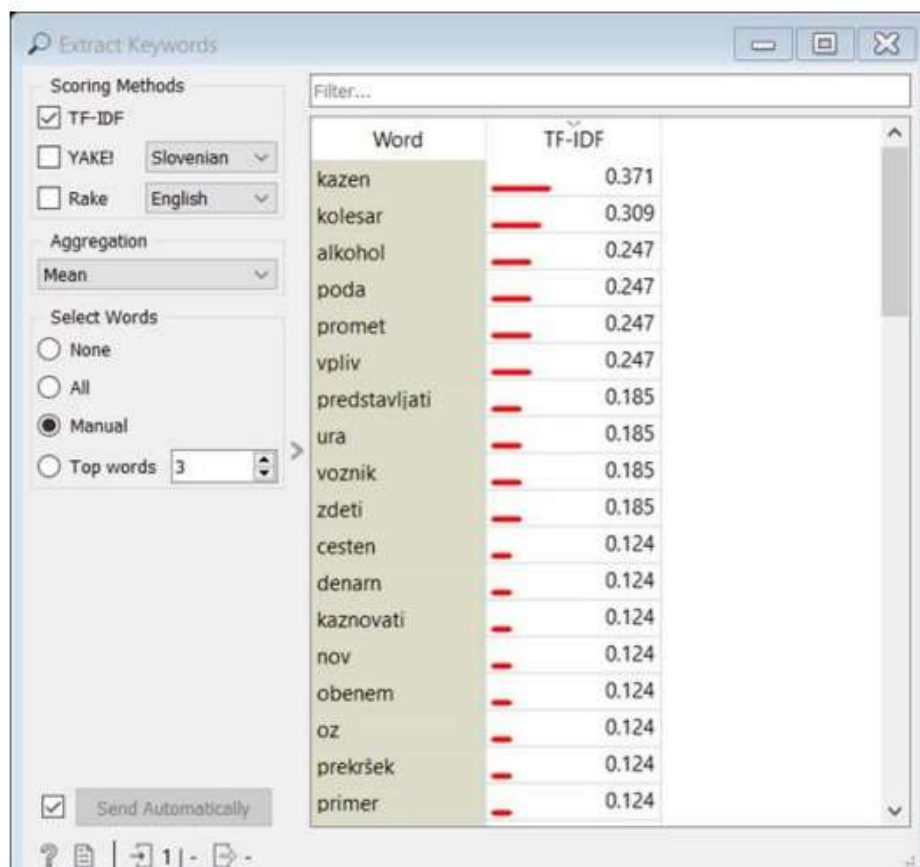
V gradniku Corpus iz poljubnega URL naslova ali iz izbrane mape datotečnega sistema naložimo datoteko z besedili in označimo, katere oznake dokumenta predstavljajo besedilo, s čimer so besedila pripravljena za predobdelavo. Da dobimo ustrezno korpus datoteko je treba najprej besedila ustrezno pretvoriti v golo besedilo.

2.1.25.2. Extract Keywords

Gradnik vsebuje metode točkovanja oz. ocenjevanja pomena besede za dokument. Gradnik vsebuje naslednje metode:

- **TF-IDF** - numerična statistika, ki meri pomembnost posamezne besede v dokumentu
- **YAKE7!** - metoda izvlečenja besed
- **Rake** - izvlečenje ključnih besed z razčlenjevanjem besedila v matriko

Na Slika 26 je prikazan primer ocenjevanja ključnih besed s pomočjo metode TF-IDF.



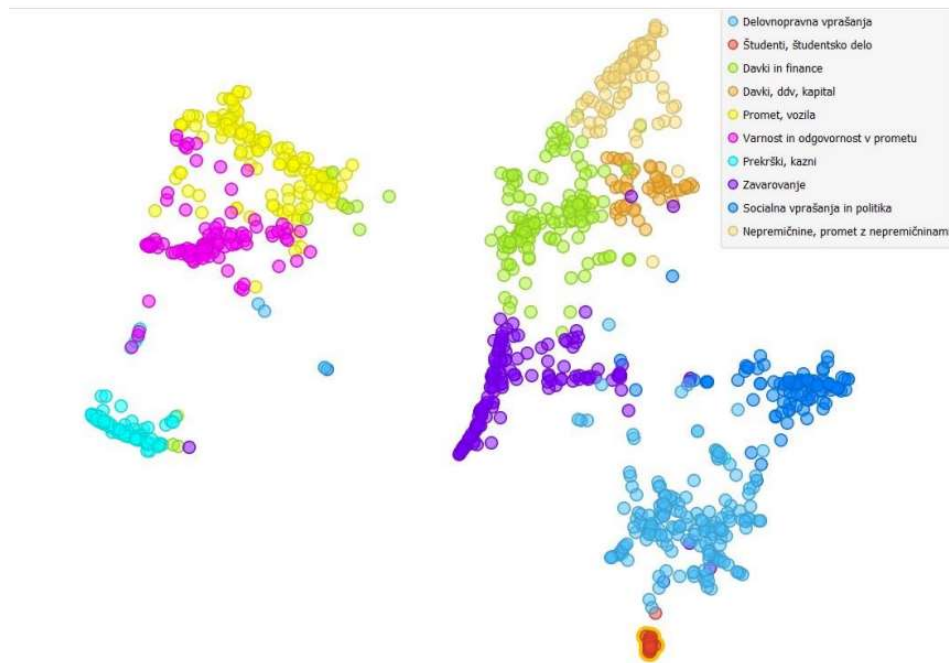
Slika 26: Ocenjevanja ključnih besed s pomočjo metode TF-IDF

2.1.25.3. Karta besedil - prikaz sorodnosti med besedili

Besedila, ki spadajo v isto skupino, vsebujejo podobne vsebine. Če med besedili iščemo tista, ki govorijo o neki vsebini, zadošča, da pregledamo le besedila v ustrezni skupini in tako močno skrčimo število in obseg besedil, ki bi jih sicer moral uporabnik v celoti natančno pregledati. Med izbranimi se je smiselno osredotočiti na tista besedila, ki so na karti narisana bolj skupaj. Na ta način lahko poiščemo besedila, ki govorijo o podobni vsebini kot neko določeno besedilo. Poiskati moramo le, v katero skupino sodi iskano besedilo.

V primeru na Slika 27 je uporabljen vzorčni nabor besedil iz javne zbirke "Predlagam vladi", ki na dan **15. 9. 2021** vsebuje 11.471 dokumentov oz. predlogov državljanov in drugih subjektov ter **3.528** odzivov nanje. Zraven je bil dodan vzorec **353 zakonskih besedil** kot vir črpanja možnih podlag za odgovore na vprašanja oziroma problematiko iz predlogov.

Zemljevid besedil dokumentov s Predlogi vladi RS s prikazom skupin je prikazan na Slika 27.



Slika 27: Zemljevid besedil dokumentov s predlogi vladi RS s prikazom skupin.

3. Procesi, vloge (uporabnikov) in primeri uporabe

Pri zajemu zahtev in primerov uporabnikov orodja SEMANT so bili identificirani naslednji morebitni uporabniki:

- Ministrstvo za digitalno preobrazbo, Direktorat za podporo uporabnikov
- Ministrstvo za javno upravo, Direktorat za kakovost, Sektor za kakovost predpisov in javne uprave
- Služba vlade za zakonodajo, Sektor za evropske zadeve in informatizacijo zakonodajnih postopkov
- Ministrstvo za digitalno preobrazbo, Direktorat za razvoj digitalnih rešitev in podatkovno ekonomijo

Povzete splošne ugotovitve:

- Direktorat za podporo uporabnikov nima neposredne povezave z vsebino, saj večinoma skrbijo za posredovanje vprašanj pristojnim ministrstvom. V primeru neposrednih odgovorov na uporabniška vprašanja pa Direktorat odgovarja na podlagi uradnih podatkov, ki so dosegljivi neposredno iz uradnih virov, zato v tem primeru semantični analizator ni primerno orodje.
- Sektor za kakovost predpisov in javne uprave bi semantični analizator vključil v lastno aplikacijo, uporabljal pa bi besedila iz vseh virov, tako notranjih kot zunanjih.
- Služba vlade za zakonodajo bi semantični analizator tudi uporabljala v okviru lastne aplikacije, vendar le za besedila iz svojih virov – zbirke predpisov.
- Direktorat za razvoj digitalnih rešitev in podatkovno ekonomijo bi semantični analizator uporabljal v povezavi s pripravo preverjenih ontologij v svojem lastnem sistemu.

3.1. MDP, Direktorat za podporo uporabnikov, eUprava

Na Ministrstvu za digitalno preobrazbo je vzpostavljen enotni kontaktni center (EKC), ki je namenjen vsem državljanom in uslužbencem državne uprave. Pri izvajanju podpore uporabnikom gre za dvosmerno komunikacijo med uporabniki/strankami in državo po telefonu in elektronskih poteh (elektronska pošta, spletni obrazci).

Državna uprava prek Enotnega kontaktnega centra (EKC) in aplikacije Maximo sprejema in rešuje težave uporabnikov, jim zagotavlja kakovostne informacije o organizaciji in poslovanju državnih organov, upravnih in drugih storitvah ter jim nudi tehnično pomoč pri uporabi elektronskih storitev državne uprave. Sprejema tudi njihove pobude in predloge za izboljšanje poslovanja, na katere se odziva s povratnimi informacijami.

Nov način izvajanja sistema za podporo upravljanju omogoča hitro in učinkovito spremljanje poslovnih procesov, komuniciranje z uporabniki se je izboljšalo, skrajšal se je čas reševanja težav, hkrati pa se učinkoviteje spremlja izpolnjevanje uporabniških zahtev.

Primeri pomoči:

- pomoč pri uporabi e-storitev državne uprave za državljane,
- pomoč pri vsebinskih vprašanjih – podatki iz uradnih strani državne uprave,
- urejanje SIPASS računov,
- pomoč pri nameščanju državnih digitalnih potrdil.

Pomoč poteka preko telefona, spletnih obrazcev in elektronske pošte. Najbolj časovno zahtevno je reševanje tehničnih vprašanj, ko je potrebno diagnosticirati vzrok napake. Pogosto stranka ne dovoli povezave na njen računalnik in je težavo možno rešiti samo z vodenjem uporabnika preko telefona.

EKC bi v prihodnje semantični analizator uporabljali pri zbirki vprašanj in odgovorov (MAXIMO), ki jih uporabniki pošiljajo preko spletnih obrazcev eUprave in elektronske pošte.

Za njihovo delo potrebujejo spletne vsebine v dejanskem trenutku, zato trenutno nimajo potrebe po uporabi semantičnega analizatorja.

Tabela 6 Viri Enotnega kontaktnega centra

Ime vira	Naslov vira
Izvajanje enotne informacijske podpore	https://www.gov.si teme/informatika-v-drzavni-upravi/
Enotni kontaktni center	https://www.gov.si/drzavni-organi/ministrstva/ministrstvo-za-digitalno-preobrazbo/o-ministrstvu-za-digitalno-preobrazbo/urad-za-podporo-uporabnikom/enotni-kontakt-center/

3.2. Ministrstvo za javno upravo, Direktorat za kakovost, Sektor za kakovost predpisov in javne uprave

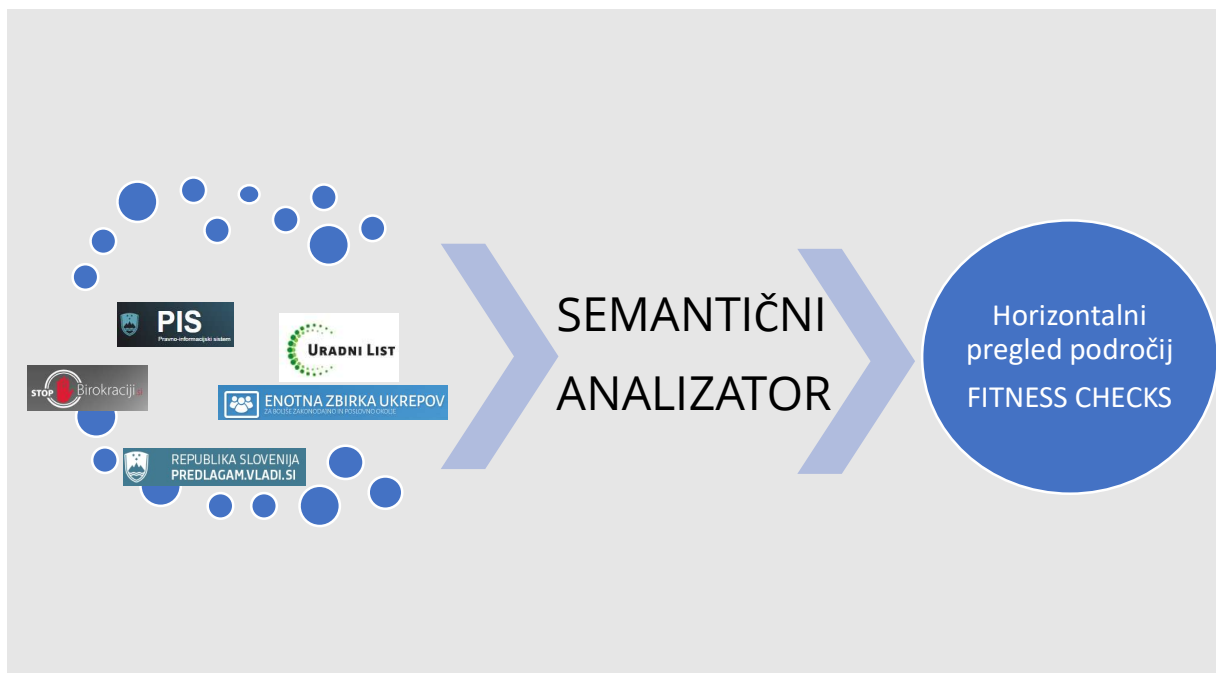
Na Direktoratu za kakovost se opravljajo naloge na področju systemske ureditve razvojnega načrtovanja, boljše zakonodaje, kakovosti in inovativnosti v javni upravi, vzpostavitve in upravljanja sistema usposabljanja, izpopolnjevanja in dviga kompetenc javnih uslužbencev ter systemskega urejanja splošnega upravnega postopka, systemskega urejanja upravnih taks in upravnega poslovanja.

Znotraj Direktorata v Sektorju za kakovost predpisov in javne uprave (v nadaljevanju Sektor) stremijo k pripravi boljših predpisov, ki so pomembni za doseganje kakovostnega zakonodajnega okolja. Ob zavedanju pomembnosti vpeljevanja inovativnih pristopov v organe javne uprave spodbujajo ustvarjalnost, agilne pristope, vključevanje deležnikov, sooblikovanje storitev in preizkušanje storitev z deležniki ter redno spremljajo uporabniško izkušnjo.

Njihovo poslanstvo je poenostaviti življenje državljanom in poslovanje podjetjem ter s tem prispevati k znižanju stroškov pri poslovanju z državo in odpravljanju administrativnih ovir. Zavedajo se, da le zakonodaja, ki je najmanj obremenjujoča, izboljšuje konkurenčnost gospodarstva in odprtost trga, izboljšuje standard državljanov in povečuje transparentnost. S tem namenom izvajajo različne dejavnosti tako s področja preprečevanja nastajanja kot odprave administrativnih ovir in priprave bolj kakovostnih predpisov. Pri tem sodelujejo z državljanji in podjetji ter drugimi državnimi organi.

So skrbniki portala STOP Birokraciji (SB), prek katerega prejemajo pobude za odpravo administrativnih ovir ali izboljšanje predpisov. Pobude proučijo, preverijo njihovo izvedljivost in usklajujejo aktivnosti za realizacijo. Del nalog je tudi skrb nad realizacijo sprejetih pobud, ki se preoblikujejo v ukrepe z jasno določenimi cilji, roki izvedbe in nosilci, združenih v aplikaciji Enotna zbirka ukrepov za boljše zakonodajno in poslovno okolje.

Primer uporabe SEMANT-a za Sektor je prikazan na Slika 28.



Slika 28: Primer uporabe Semanta za Sektor

Tabela 7: Viri Direktorata za kakovost

Ime vira	Naslov vira
Direktorat za kakovost	https://www.gov.si/drzavni-organi/ministrstva/ministrstvo-za-javno-upravo/o-ministrstvu/direktorat-za-kakovost/
Sektor za kakovost predpisov in javne uprave	https://www.gov.si/drzavni-organi/ministrstva/ministrstvo-za-javno-upravo/o-ministrstvu/direktorat-za-kakovost/sektor-za-kakovost-predpisov-in-javne-uprave/
Stop birokraciji	https://www.stopbirokraciji.gov.si/domov

3.2.1. Primer uporabe

Na spletni strani STOP Birokraciji se odda pobuda, ki se jo umesti v neko področje ter določi pristojni organ za njeno izvedbo. Na javni strani je trenutno 828 pobud, obstaja pa zaledni sistem, v katerem je preko 1.500 pobud, ki niso javnega značaja (osebni podatki).



Administrator v zalednem sistemu pregleda pobudo, jo posreduje pristojnemu organu v pripravo odziva in odloči, katere pobude objavi (vsebino lahko tudi prilagodi, vendar le z namenom, da se pobuda popolnoma anonimizira). Pri določitvi relevantne zakonodaje, vezane na pobudo, in pristojnega organa, ki mora pripraviti odgovor, se trenutno uporablja baza PIS RS (Organi, odgovorni za pripravo tega predpisa), ki bi jo v tem koraku nadomestila uporaba orodja SEMANT. Sočasno bi navedeno orodje preverilo med bazama že prejetih pobud na SB in Predlagamvladi.si, s čimer bi se optimiziral proces priprave odgovora pristojnega organa, saj bi se jih ob posredovanju novo prejete pobude posredovali tudi predhodni odgovori z njihove strani na podobne pobude. V bazi Enotne zbirke ukrepov pa bi orodje lahko preverilo ali se na temo pobude že realizira določen ukrep in se bi v tem primeru s tem le seznanilo pobudnika.

Po prejemu odziva s strani pristojnega organa se k pobudi objavi tudi odgovor. O vsem je sočasno obveščen tudi pobudnik. V kolikor je pobuda sprejeta, se sproži preoblikovanje vsebine h ukrepom (Enotna zbirka ukrepov). Potrebno je določiti pristojni organ za njeno izvedbo, saj bo ta organ prevzel vodenje, rok za realizacijo ter izvedbo potrebnih aktivnosti za uresničitev ukrepa.

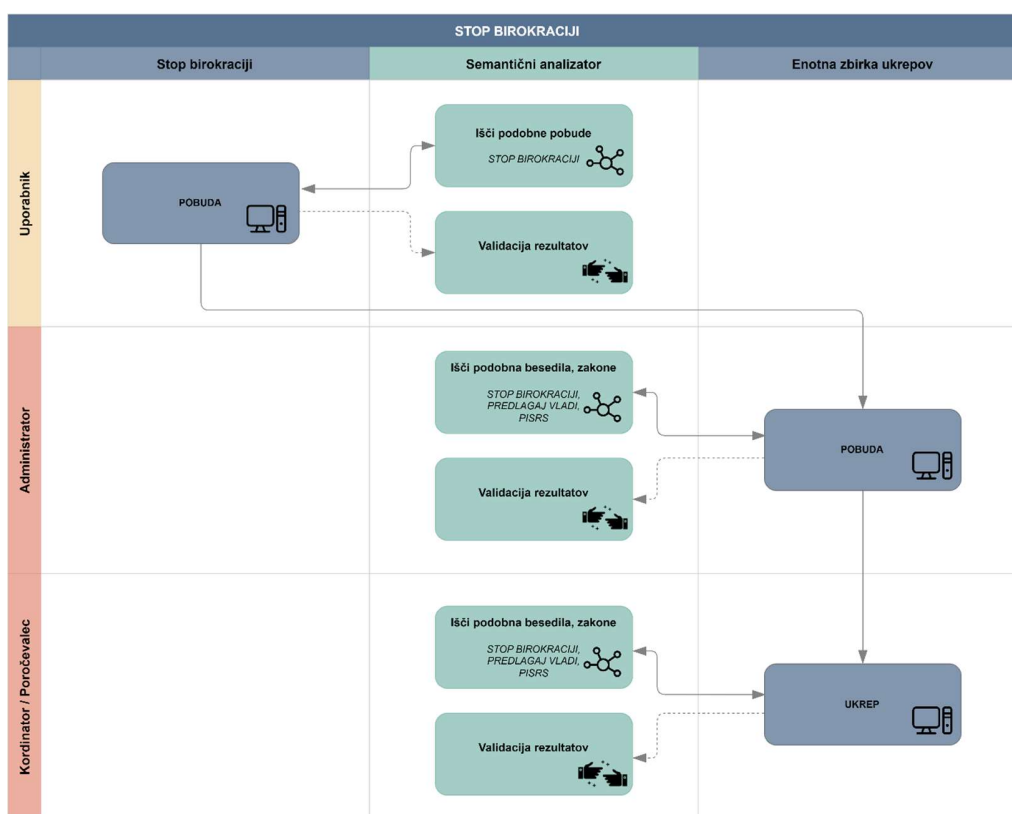
Nekatera iskanja v orodju SEMANT je potrebno omejiti samo znotraj enega sistema. Orodje mora omogočiti izbiro med različnimi viri, SEMANT pa naj bo integriran v že obstoječe sisteme (Stop birokraciji, Enotna zbirka ukrepov, PIS RS baza, Predlagamvladi.si). Primer uporabe aplikacije SEMANT v sistemu STOP Birokraciji je prikazan na Slika 29.

Uporabnik: Med procesom oddaje pobude na spletni strani STOP birokraciji se naj uporabnikom s pomočjo orodja SEMANT prikažejo povezave na obstoječe pobude z enako ali podobno vsebino. Tako bi se lahko zmanjšal obseg pobud.

Administrator: Pred sprejemom pobude mora administrator vedno preveriti, če podobna pobuda že obstaja. Za dodelitev izvedbe pobude pa mora ugotoviti, kateri organ je ustrezen za njeno izvedbo. V tem primeru je potrebna izvedba treh različnih vrst iskanja:

- Iskanje pobud znotraj virov STOP birokraciji in Predlagamvladi.si,
- Iskanje ukrepov znotraj vira Enotna zbirka ukrepov ter
- iskanje po zakonih in podzakonskih aktih (ter ustreznih njihovih metapodatkih) znotraj zbirke podatkov PISRS.

Koordinatorji / poročevalci: Ostali uporabniki sistema so različni koordinatorji in poročevalci. V aplikaciji Enotna zbirka ukrepov bi bilo orodje SEMANT v pomoč koordinatorju in poročevalcu pri njunih nalogah, in sicer pregledu pobud in ukrepov v njihovi pristojnosti.



Slika 29: Primer uporabe semantičnega analizatorja v sistemu STOP birokraciji

3.3. Služba vlade za zakonodajo

Služba vlade za zakonodajo skrbi za skladnost predpisov z zakoni in ustavo, za notranjo skladnost predpisov, upoštevanje nomotehničnih pravil pri pripravi predpisov ter končno za to, da so predpisi ljudem razumljivi in tudi v praksi učinkoviti.

Pripravljalce predpisov zato s posebno dovtetnostjo usmerjajo k uresničevanju temeljnih pravnih načel in ustaljenih pravil pisanja predpisov, jim pomagajo, kako predloge predpisov pravilno umestiti in uskladiti z ostalimi slovenskimi predpisi ter pravnim redom Evropske unije, opozarjajo pa tudi na težave in posledice, ki lahko nastopijo po uveljavitvi ne dovolj dobro pripravljenih predpisov.

Po poslovníku vlade morajo biti predlogi splošnih aktov in aktov poslovanja vlade vedno predhodno usklajeni s Službo vlade za zakonodajo. Samo če po pridobitvi mnenj zaprosenih ministrstev in vladnih služb uskladitve ni mogoče doseči, ali če predhodnega usklajevanja zaradi nujnosti postopka ni bilo mogoče izvesti, se lahko v obravnavo vladi predložijo tudi neusklajena gradiva.

Služba vlade za zakonodajo ima za pripravo mnenja 14 dni časa. Če se v postopku medresorskega usklajevanja gradivo predlagatelja spremeni, pa je treba mnenje Službe vlade za zakonodajo pridobiti ponovno. Ponovno mnenje mora biti podano najpozneje v petih delovnih dneh po prejemu spremenjenega gradiva.

3.3.1. Primer uporabe

MOPED (modularno ogrodje za pripravo elektronskih dokumentov) je tehnološko napredno orodje za vodenje podatkov o predpisih in za pripravo e-dokumentov v postopku sprejemanja predpisov.

Služba vlade za zakonodajo (SVZ) upravlja z aplikacijo MOPED, ki je namenjena pripravljavcem predpisov, predvsem tistim v okviru posameznih ministrstev. Predpis je skupen naziv za zakone, podzakonske akte, novele in podobno.

SEMANT želijo uporabljati v okviru lastnega sistema, kasneje pa tudi kot samostojno aplikacijo. SVZ vidi glavno prednost SEMANT-a v pomoči pri želji po uskladitvi izrazov in dikcij v sami zakonodaji – znotraj posameznega predpisa, kakor tudi na splošno v vseh predpisih.

Pomembnejši izrazi se običajno pojavljajo v posameznem členu o opredelitvi pojmov in so največkrat v navednicah. Ti izrazi se znotraj istega predpisa že nekako poenoteno uporabljajo, ni pa pregleda nad različnimi predpisi ter splošnega poenotenja teh izrazov. Želja SVZ-ja je, da bi v sklopu celotne zakonodaje uporabljali bolj poenotene izraze in dikcije.

Zaključki:

- Integracija orodja SEMANT v orodje MOPED bi z iskanjem po vsebinah ter predlogih sorodnih vsebin omogočila takojšnjo pomoč ob pripravi besedil.
- Uporaba orodja SEMANT za pripravo vira podatkov za poslovno inteligenco za pregled stanja različnih dikcij (povezava s sistemom Skrinja).

Tabela 8 Viri Službe vlade za zakonodajo

Ime vira	Naslov vira
Konferenca DSI 2021	https://dsi2021.dsi-konferenca.si/uploads/zbornik/pdf/ABDSI21_paper_13.pdf
O Službi vlade za zakonodajo	https://www.gov.si/drzavni-organi/vladne-sluzbe/sluzba-vlade-za-zakonodajo/o-sluzbi/
MOPED	https://www.gov.si/assets/vladne-sluzbe/SVZ/37c52320ea/Sistem-MOPED.pdf

3.4. MDP, Direktorat za razvoj digitalnih rešitev in podatkovno ekonomijo

Ministrstvo za digitalno preobrazbo spremlja in analizira stanje digitalne preobrazbe in informacijske družbe na državni ravni. Pristojno je za delovno področje informacijske družbe, elektronskih komunikacij, digitalne vključenosti, digitalnih kompetenc, podatkovne ekonomije, upravljanja z informacijsko-komunikacijskimi sistemi in zagotavljanja elektronskih storitev javne uprave. V sodelovanju s pristojnimi ministrstvi in vladnimi službami pripravlja, usklajuje in izvaja državne ukrepe in projekte na področju informacijske

družbe in digitalne preobrazbe gospodarstva, javne uprave, zdravstva, pravosodja, kmetijstva in izobraževanja ter drugih področjih.

Razvoj digitalne uprave, uvajanje poslovne inteligence in obdelav podatkov velikega obsega, objava odprtih podatkov, sočasno zagotavljanje vgrajene zasebnosti, zagotavljanje skladnosti po GDPR in vgrajena informacijska varnost so vse aktivnosti, ki temeljijo na podatkih. Vendar samo zbiranje podatkov ni dovolj, pomembno je tudi pravilno in enako razumevanje pomena podatkov pri vseh uporabnikih. S pristopi, ki jih opredeljujejo in uvajajo, želijo tudi preseči trenutno situacijo in ponuditi platforme ter orodja za učinkovit pregled nad obstoječimi podatkovnimi viri (predvsem registri), njihovimi strukturami in enostavno gradnjo učinkovito povezanih podatkovnih modelov. Ob tem želijo hkrati tudi postaviti skupne standarde na celi vertikalni področja semantične interoperabilnosti. Vertikalo razumejo od ontologij na vrhu, preko metapodatkovnih slovarjev, jedrnih modelov in logičnih domenskih modelov, do najnižjega nivoja, ki ga predstavljajo fizični podatkovni modeli - sestavni deli informacijskih rešitev. S pomočjo orodij in postopkov, opredeljenih v strategiji, prednostno rešujejo:

- enkratni zapis oz. načelo samo enkrat (ang. "Once Only Principle"),
- učinkovitejše in standardizirano načrtovanje modelov informacijskih rešitev (pri novih in reinžiniringu obstoječih),
- večjo stopnjo zanesljivosti in kakovosti podatkov, ki se izmenjujejo med sistemi (pravilo ene resnice – enako razumevanje pomena podatkov pri vseh uporabnikih).

Tabela 9 Viri Ministrstva za digitalno preobrazbo

Ime vira	Naslov vira
O Ministrstvu za digitalno preobrazbo	https://www.gov.si/drzavni-organi/ministrstva/ministrstvo-za-digitalno-preobrazbo/o-ministrstvu-za-digitalno-preobrazbo/
Strategija upravljanja semantične interoperabilnosti	https://nio.gov.si/nio/asset/strategija+upravljanja+semanticne+interoperabilnosti?lang=sl%20

3.4.1. PzSI - Platforma za semantično interoperabilnost

Za namen zagotavljanja semantične interoperabilnosti in principa enkratnega zapisa je v slovenski javni upravi trenutno vzpostavljen sistem, ki ga sestavljajo naslednji moduli:

- Orodje za upravljanje s terminologijami; omogoča gradnjo, upravljanje, objavo in vizualizacijo nadzorovanih besednjakov.
- Orodje za upravljanje s podatkovnimi modeli; omogoča gradnjo, upravljanje, objavo in vizualizacijo jedrnih podatkovnih modelov.
- Orodje za upravljanje s šifranti; omogoča gradnjo, upravljanje in objavo šifrantov.
- Orodje za komentiranje; omogoča komentiranje za vse tri zgornje rešitve.
- Orodje za uporabnike; omogoča upravljanje s pravicami uporabnikov.

Sistem omogoča:

- oblikovanje enotnega repozitorija definicij temeljnih pojmov,
- oblikovanje enotnega repozitorija šifrantov,
- doseganje višje stopnje standardizacije in interoperabilnosti na podatkovnem sloju,
- znižanje visoke stopnje heterogenosti informacijskih sistemov,
- definiranje razmerij med registri in evidencami javne uprave,
- učinkovitejše zagotavljanje medresorskih storitev,
- učinkovitejšo uporabo informacijskih virov,
- doslednost pri izkazovanju in razumevanju podatkov,
- zagotavljanje principa enkratnega zapisa,
- učinkovitejšo strojno berljivost podatkov ter
- razvoj naprednih aplikativnih rešitev s področja umetne inteligence in upravljanja z znanjem.

3.4.2. Primer uporabe

Osnovni primer uporabe je, da MDP najprej izbere ontologijo, nato pa izbere enega ali več pojmov iz nje in začne iskati ustrezne vsebine v podatkovnih zbirkah (npr. podatkovna zbirka zakonodaje), kjer išče definicije in sorodne pojme.

Naprednejši primer uporabe je avtomatsko generiranje ontologij.

SEMANT naj služi kot orodje za pomoč pri izgradnji ontologij, vendar pa naj v prvi fazi ne ustvarja ontologij samodejno. Pomoč pri izgradnji ontologij pomeni natančnost in potrditev s strani "avtoritete," da so informacije pravilne. Pri tem procesu ima ključno odločevalno moč urednik.

V orodje SEMANT bi morali kot enega izmed virov podatkov integrirati vse obstoječe ontologije. Prva uporaba orodja SEMANT bi lahko bila za vnose pojmov iz platforme PzSI. Podatki bi se vnašali v obliki RDF datoteke, ki vsebuje vse podatke ontologije. Zaželeno je, da SEMANT omogoča prikaz RDF strukture.

Iskanje sorodnih pojmov:

- Uporabnik določi pojem ali izraz za iskanje (znotraj orodja za terminologije).
- Uporabnik izbere zbirke znotraj katerih želi izvesti iskanje.
- SEMANT uporabi ontologijo za identifikacijo sorodnih pojmov.

Iskanje v dokumentih:

- SEMANT išče dokumente, ki vsebujejo izbrani pojem ali sorodne pojme.

Filtriranje rezultatov:

- SEMANT filtrira dokumente ter tako prikaže samo relevantne dele zakonodaje.

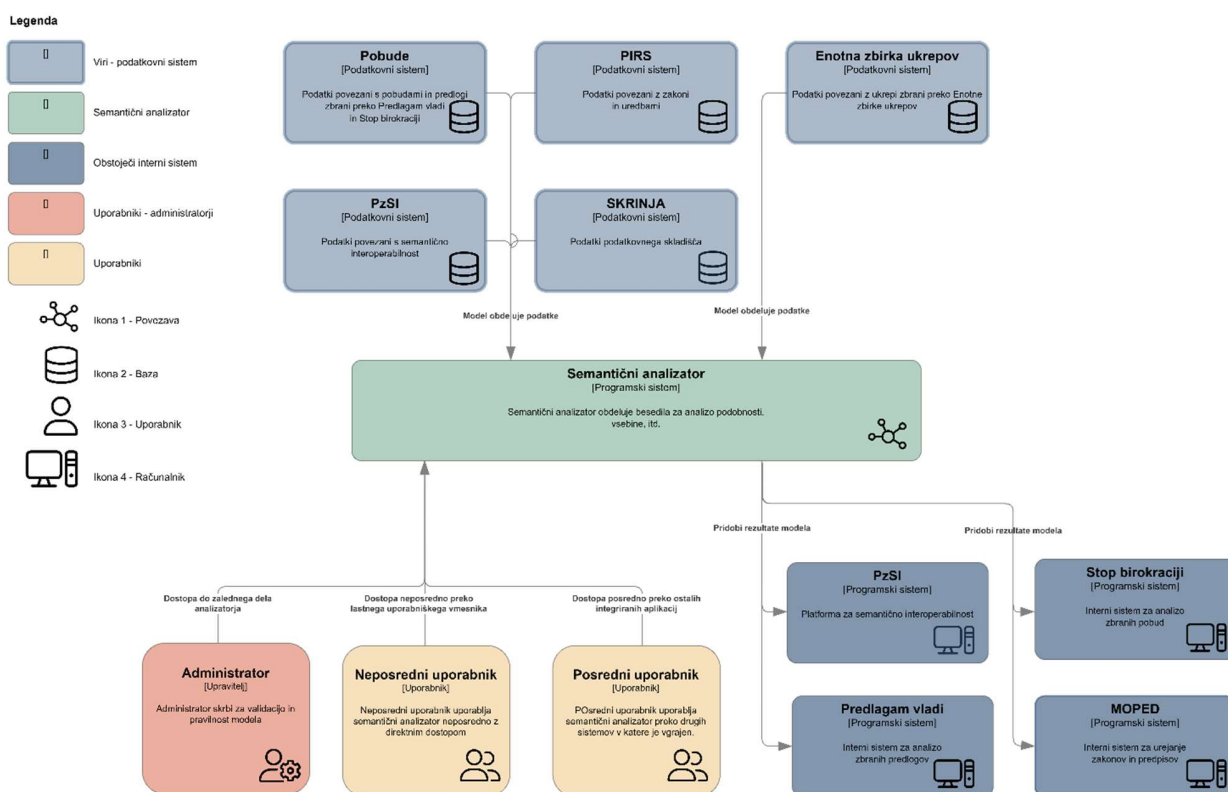
Pregled relevantnih delov zakonodaje:

- Uporabnik pregleduje relevantne dele dokumentov, ki vsebujejo izbrani pojem.
- Uporabnik glasuje o primernosti vsebine.

4. Logična arhitektura

4.1. Integracija aplikacije »Semantični analizator« z drugimi zunanji aplikacijami

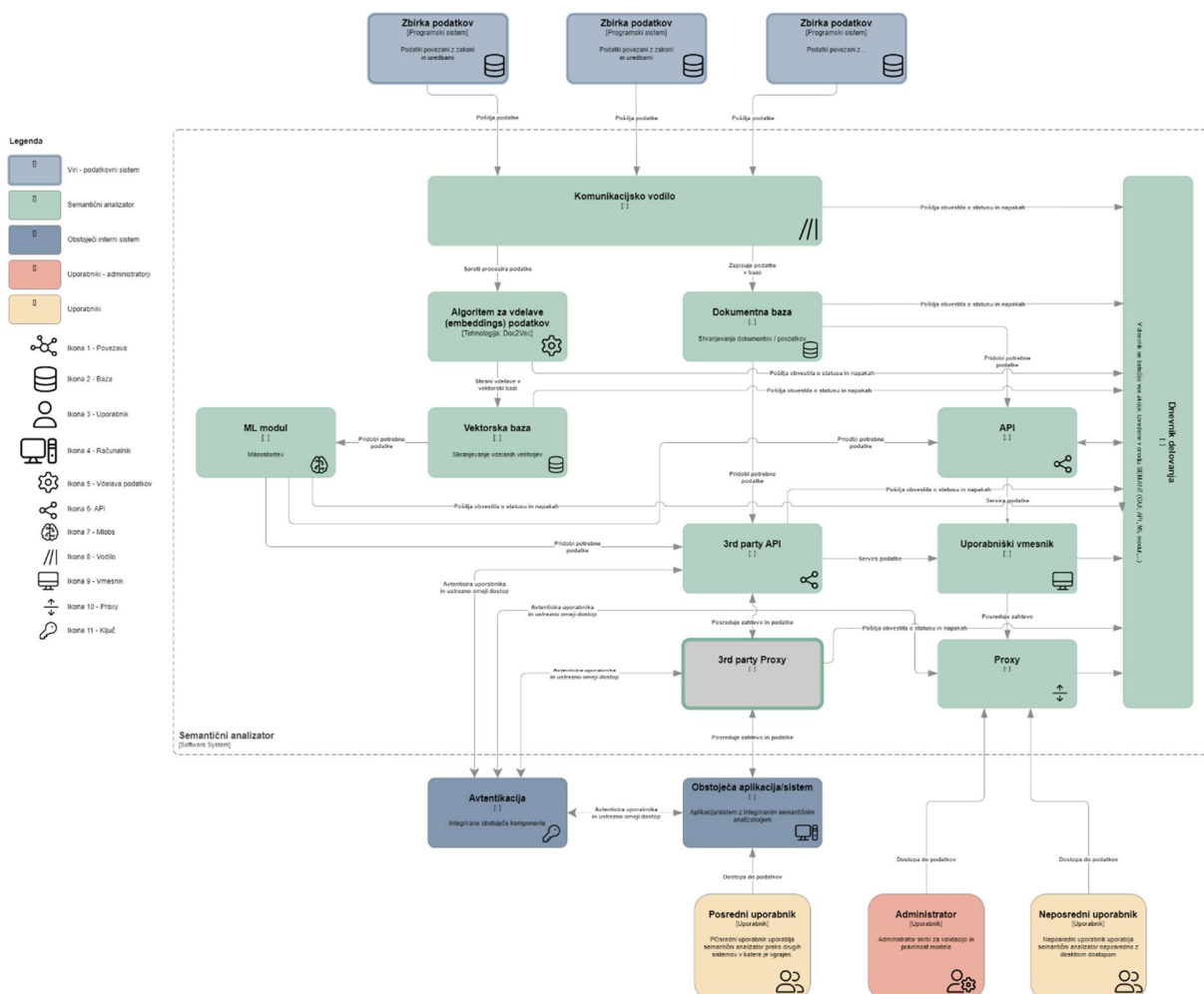
Integracija aplikacije SEMANT z drugimi zunanji aplikacijami je prikazana na diagramu na Slika 30.



Slika 30: Visokonivojska logična arhitektura

4.2. Arhitektura aplikacije »Semantični analizator«

Arhitektura aplikacije »Semantični analizator« (SEMANT) je prikazana na Slika 31.



Slika 31: Podrobna logična arhitektura aplikacije SEMANT

4.2.1. Komunikacijsko vodilo

Preko komunikacijskega vodila poteka izmenjava podatkov. Po eni strani kot posrednik skrbi, da gradniki sistema niso odvisni od ostalih gradnikov in zunanjih sistemov, zato v primeru težav pri pridobivanju podatkov iz zunanjih sistemov ali shranjevanju v podatkovno bazo ti problemi ne vplivajo na delovanje ostalih gradnikov. Po drugi strani pa omogoča fleksibilnost pri širitvah platforme z drugimi podatkovnimi viri ali dodatnimi odjemalci podatkov.

Ker ima osrednji gradnik pomembno vlogo v celotnem sistemu je ključno, da je komunikacijsko vodilo postavljeno na čim bolj zanesljiv in robusten način, s pomočjo delovnih kopij, ki omogočajo nemoteno delovanje sistema tudi v primeru težav. V skladu s priporočili so predvidene tri delovne kopije in uspešen zapis na vsaj dveh od njih, preden je sporočilo obravnavano kot sprejeto.

Komunikacijsko vodilo je sicer širok pojem, ki vključuje različne sisteme in principe izmenjave podatkov znotraj sistema in med sistemi. V našem primeru se osredotočamo na dogodkovna komunikacijska vodila, ki so namenjena izmenjavi pretočnih podatkov.

Nekaj odprtokodnih rešitev, ki so na voljo, je sledečih:

1. *Apache Kafka*

Apache Kafka je porazdeljena platforma za pretakanje podatkov in obdelavo dogodkov, ki je zasnovana za visoko stopnjevalnost, vzdržljivost in visok pretok podatkov. Deluje po principu objav in naročanja (ang. *publish / subscribe*). Uporablja se za zajemanje, shranjevanje in obdelavo velikih količin podatkov v realnem času. Kafka se pogosto uporablja v aplikacijah za analitiko, sledenje dogodkom in gradnjo mikrorstitev.

2. *Apache Pulsar*

Apache Pulsar je razširjena platforma za objavo in pretakanje dogodkov, zasnovana za visoko pretočnost, nizko zakasnitev in odlično stopnjevalnost. Ponuja podporo za večnajemniški model (ang. *multi-tenancy*) in zagotavlja obstojnost podatkov.

3. *RabbitMQ*

RabbitMQ je priljubljen odprtokodni posrednik sporočil, ki podpira različne vzorce sporočanja, vključno z objavo-naročanjem (ang. *publish-subscribe*) in čakanjem na sporočila. Slovi po enostavni uporabi in obsežni podpori skupnosti.

4. *Apache ActiveMQ*

Apache ActiveMQ je odprtokodni posrednik sporočil, ki podpira Java Message Service (JMS) API. Pogosto se uporablja v aplikacijah, ki temeljijo na programskem jeziku Java ter ponuja funkcionalnosti, kot sta združevanje in visoka razpoložljivost.

5. *NATS*

NATS je lahek in visoko zmogljiv sistem za sporočanje, ki se osredotoča na preprostost in hitrost. Primeren je za arhitekture mikrorstitev in aplikacije za internet stvari (IoT).

6. *Apache RocketMQ*

Apache RocketMQ je porazdeljena platforma za sporočanje in pretakanje. Ponuja doslednost (ang. *strong consistency*) in visoko pretočnost ter se običajno uporablja v oblčnih storitvah in aplikacijah za elektronsko poslovanje.

7. *Redis Streams*

Redis, sicer priljubljena shramba v pomnilniku, ponuja tudi podporo za pretakanje podatkov. Zato je primeren za pretakanje dogodkov z nizko zakasnitvijo in visoko pretočnostjo.

4.2.2. Algoritem za vložitve (angl. *embeddings*) podatkov

Vložitve (angl. *Embeddings*) v kontekstu strojnega učenja pomenijo predstavitev podatkov v vektorski obliki. Ta tehnologija se najpogosteje uporablja pri obdelavi besed ali besednih nizov, vendar je možno ustvariti vložitve za različne vrste podatkov. Temeljno vodilo pri tej tehnologiji je, da podobne informacije ali entitete v izvirnem prostoru ostanejo podobne tudi v vektorski obliki (vektorskem prostoru). To pomeni, da bodo podobne besede (glede na kontekst ali pomen) imele podobne vektorske predstavitve. **Word2Vec** je verjetno najbolj znan algoritem za ustvarjanje vložitev besed, ki napoveduje besedo na podlagi njenega okoliškega konteksta (ali obratno) in pri tem ustvarja vektorske predstavitve besed, ki dobro ujamejo semantične in sintaktične odnose med besedami. Po drugi strani je **Doc2Vec** (ali **Paragraph2Vec**) razširitev ideje Word2Vec, ki ne predstavlja le posameznih besed v vektorskem prostoru, ampak celotne dokumente ali stavke. Distributed Memory (DM) in Bag of Words (BoW) sta najbolj znana pristopa Doc2Vec algoritma.

Z algoritmom Doc2Vec je mogoče pridobiti vektorsko predstavitev katerega koli dokumenta, ki je bil vključen v učni nabor, ali katerih koli algoritmu še neznanih dokumentov. Te vložitve omogočajo, da besedila obravnavamo kot vektorske objekte, kar olajša številne naloge kot so iskanje podobnosti ali uporabo različnih pristopov strojnega učenja, kot sta klasifikacija in gručenje.

Vektorske predstavitve dokumentov oz. vložitve se shranjujejo v večdimenzijsko oz. vektorsko zbirko podatkov, ki je opisana v podpoglavju 4.2.3.2.

4.2.3. Podatkovne shrambe

4.2.3.1. Dokumentna baza

Za shranjevanje celotnih ali delnih dokumentov (povzetkov) za uporabo znotraj semantičnega analizatorja je priporočljiva uporaba nestrukturirane dokumentne baze. Nestrukturirane dokumentne baze so idealne za shranjevanje in obdelavo velikih količin nestrukturiranih podatkov, kot so tekstovni dokumenti, ker omogočajo fleksibilno shranjevanje podatkov brez predhodno določene sheme.

MongoDB je ena izmed najbolj priljubljenih odprtokodnih dokumentnih baz, ki omogoča shranjevanje podatkov v obliki JSON. To omogoča enostavno shranjevanje kompleksnih podatkovnih struktur, kot so dokumenti, povzetki in semantične analize, ter omogoča hitro iskanje in indeksiranje teh podatkov. MongoDB podpira raznolike operacije nad dokumenti, vključno z iskanjem po besedilu, kar je ključnega pomena za semantični analizator. MongoDB je primeren za aplikacije, ki zahtevajo visoko razpoložljivost in razširljivost, saj omogoča distribuirano shranjevanje podatkov preko več strežnikov.

DynamoDB je popolnoma upravljana NoSQL baza podatkov, ki jo ponuja Amazon Web Services (AWS). Čeprav ni odprtokodna, je DynamoDB močno razširljiva in zagotavlja hitro in predvidljivo zmogljivost z minimalnim zamikom. DynamoDB omogoča shranjevanje in pridobivanje katerih koli količin podatkov ter služi kot robustna rešitev za aplikacije, ki potrebujejo visoko zmogljivost in zanesljivost. DynamoDB podpira fleksibilne sheme in dinamične poizvedbe, kar je koristno za dinamično obdelavo in analizo dokumentov v semantični analizator.

Pri izbiri dokumentne baze za shranjevanje in obdelavo dokumentov za semantični analizator je pomembno upoštevati več dejavnikov, kot so:

- *Razširljivost*: Sposobnost baze, da se prilagodi rasti podatkov in zahtevam po zmogljivosti.
- *Zmogljivost*: Hitrost in učinkovitost pri shranjevanju, iskanju in pridobivanju dokumentov.
- *Varnost*: Možnosti za zagotavljanje varnosti podatkov in nadzora dostopa.
- *Stroški*: Stroški vzpostavitve in vzdrževanja baze, vključno z morebitnimi stroški gostovanja v oblaku.

- *Podpora in skupnost*: Razpoložljivost dokumentacije, orodij in skupnosti za podporo razvoju in odpravljanju težav.

Za semantični analizator, ki bo zahteval obdelavo velikih količin tekstovnih dokumentov, je ključnega pomena izbira primerne dokumentne baze, ki omogoča hitro iskanje, indeksiranje in analizo podatkov, ob hkratnem zagotavljanju potrebne razširljivosti in zmogljivosti. MongoDB in DynamoDB predstavljata dve močni opciji, ki lahko zadostita tem zahtevam, odvisno od specifičnih potreb projekta.

4.2.3.2. Vektorska baza

Vektorska baza je specializirana baza podatkov zasnovana za učinkovito hranjenje in iskanje vektorskih podatkov. V kontekstu besednih vložitev se uporablja za hranjenje vektorskih predstavitev, ki so pridobljene iz algoritma za vložitve. Glavna značilnost vektorskih baz je, da omogočajo visoko učinkovito iskanje podobnih vsebin - torej iskanje najbližjih vektorjev v bazi glede na dano merilo podobnosti (najpogosteje kosinusna podobnost).

Klasična (ne vektorska) baza podatkov sprejme poizvedbo in vrne podatke, ki natančno ustrezajo vsem pravilom in omejitvam v tej poizvedbi. Ključna razlikovalna značilnost vektorske baze je, da rezultati poizvedb niso natančno ujemanje z vprašanjem. Namesto tega, uporabljajoč določeno merilo podobnosti, vektorska baza vrne vložitve, ki so podobne poizvedbi. Z uporabo vektorske baze se implementirajo ključni uporabniški primeri, kjer je potrebno uporabniku posredovati podatke, ki so podobni določenim podatkom – dokumente, ki so podobni določenemu dokumentu.

Osnovne značilnosti vektorske baze so:

- **Indeksiranje** - vektorska baza podatkov indeksira vektorje z različnimi algoritmi (PQ, LSH ali HNSW). Ta korak preslika vektorje v podatkovno strukturo, ki bo omogočila hitrejša iskanje.
- **Poizvedovanje** - vektorska baza podatkov primerja indeksirani vektor poizvedbe z indeksiranimi vektorji v naboru podatkov in poišče najbližje sosede (uporabi metriko podobnosti, ki jo uporablja ta indeks)
- **Naknadna obdelava** (angl. post-processing) - V nekaterih primerih vektorska baza podatkov pridobi najbližje sosede iz nabora podatkov in jih naknadno obdelava, da vrne končne rezultate. Ta korak lahko vključuje ponovno razvrščanje najbližjih sosedov z uporabo drugačne mere podobnosti.

Vektorska baza omogoča:

- shranjevanje vložitve dokumentov v več dimenzijski vektorski obliki skupaj z metapodatki o dokumentih,
- shranjevanje obstoječih ontologij,
- iskanje podobnih vektorjev (oz. dokumentov z najbolj podobnimi vsebinami),
- zajem podatkov za učenje ML modelov,
- beleženje povratne informacije po validaciji s strani končnega uporabnika: Ali je bil vrnjen rezultat ustrezen ali ne?
- Poizvedbe za ustvarjanje povratne informacije končnemu uporabniku (klic do storitev za izdelavo napovedi).

Za shranjevanje in upravljanje vektorskih podatkov v kontekstu semantičnega analizatorja, ki zahteva učinkovito iskanje in primerjanje visoko-dimenzijskih vektorskih prostorov, obstajajo odprtokodne rešitve vektorskih baz, ki so posebej optimizirane za tovrstne naloge. Tukaj je nekaj predlogov:

1. **Milvus:** Milvus je odprtokodna vektorska podatkovna baza za shranjevanje in iskanje vektorskih podatkov v velikem merilu. Zasnovana je za visoko zmogljivo iskanje podobnosti v realnem času in podpira različne algoritme indeksiranja, kar omogoča hitro in učinkovito obdelavo vektorskih poizvedb.
2. **Faiss** (Facebook AI Similarity Search): Razvila ga je ekipa Facebook AI in je knjižnica za učinkovito iskanje podobnosti med vektorskimi podatki. Faiss je zasnovan za obdelavo velikih naborov podatkov in omogoča hitro iskanje najbližjih sosedov v visokodimenzijskih prostorih.
3. **Vectara:** Vectara je še ena močna odprtokodna rešitev za vektorsko iskanje, ki omogoča enostavno upravljanje in iskanje v visokodimenzijskih vektorskih prostorih. Primerna je za različne aplikacije, vključno s semantičnim iskanjem in analizo dokumentov.
4. **Weaviate:** Weaviate je odprtokodna vektorska iskalna baza, ki podpira grafične in tekstovne poizvedbe. Njena arhitektura omogoča enostavno shranjevanje, iskanje in povezovanje kompleksnih vektorskih podatkov, kar je idealno za aplikacije, ki zahtevajo semantično analizo in razumevanje naravnega jezika.

5. ChromaDB: ChromaDB je odprtokodna vektorska baza zasnovana posebej za aplikacije, ki uporabljajo velike jezikovne modele (LLM). Ponuja enostaven vtičnik za uporabo API in impresivno zmogljivost, kar jo naredi odlično izbiro za različne aplikacije, ki uporabljajo vložitve. ChromaDB omogoča učinkovito shranjevanje in poizvedovanje po podatkih o vložitvah in razširja zmogljivosti tradicionalnih relacijskih baz na vložitve. ChromaDB tako predstavlja močno orodje za razvoj aplikacij, ki zahtevajo napredno obdelavo in analizo jezikovnih podatkov, kar je še posebej relevantno za razvoj semantičnega analizatorja. Z njeno uporabo lahko razvijalci semantičnega analizatorja učinkovito shranjujejo, upravljajo in poizvedujejo po velikih količinah vektorskih podatkov, kar omogoča hitro in natančno iskanje ter analizo semantično bogatih dokumentov.

Vsaka od zgoraj navedenih rešitev ima svoje prednosti in omejitve, zato je priporočljivo, da se skrbno in podrobno preuči ter ovrednoti vsako možnost glede na specifične zahteve projekta, kot so zmogljivost, stroški, razpoložljivost virov ter potrebna tehnična podpora. Pred implementacijo je prav tako priporočljivo opraviti nekaj preizkusov zmogljivosti ali pilotnih projektov, da se lahko zagotovi optimalno delovanje izbrane rešitve.

4.2.3.3. Sinhronizacija podatkov v bazah

V obeh bazah so shranjeni enaki dokumenti, le na drug način. V dokumentni bazi so shranjeni besedilni dokumenti ter njihovi metapodatki, v vektorski pa vdelani dokumenti in njihovi metapodatki. Vsak dokument ima v vsaki bazi unikaten ID, v metapodatkih o dokumentih pa se v vsaki bazi nahaja informacija o ID-ju istega dokumenta v drugi bazi. Na ta način sta bazi povezljivi med seboj.

Podmnožica metapodatkov o dokumentih je zaradi hitrejšega delovanja sistema shranjena v obeh bazah hkrati. Spremembe te podmnožice metapodatkov v eni bazi se morajo zato zavesti tudi v drugi bazi. Zaradi tega je v ML modulu predvidena funkcionalnost sinhronizacije podatkov, ki preko API vmesnikov do obeh baz podatke samodejno sinhronizira.

4.2.4. Aplikacije

4.2.4.1. Uporabniški vmesnik

Uporabniški vmesnik je preprosta spletna aplikacija, preko katere bo administrator lahko pregledoval in potrjeval predloge modela. Uporabniki z neposrednim dostopom bodo prek spletne aplikacije lahko uporabljali vse funkcionalnosti semantičnega analizatorja. Bolj podrobno so funkcionalnosti opisane v poglavju 5.

Vmesnika REST API in 3rd party REST API

Semantični analizator uporablja REST API vmesnik za dostop do rezultatov strojnega modela in drugih podatkov oz. dokumentov v sklopu funkcionalnosti semantičnega analizatorja.

Preko 3rd party REST API vmesnika pa storitve semantičnega analizatorja upodabljajo drugi sistemi in na ta način integrirajo SEMANT v svoje funkcionalnosti.

4.2.4.2. Posrednik (Proxy)

Posrednik za dostop do uporabniškega vmesnika in 3rd party API-ja (lahko je tudi enoten).

4.2.5. Avtentikacija

Predvidena je integracija z obstoječim sistemom za avtentikacijo v okviru gradnika SI-PASS in po potrebi informacijske rešitve KeyCloak.

4.2.6. Dnevnik delovanja

V dnevnik se zapisujejo obvestila o akcijah vseh gradnikov SEMANT. Obvestila o akcijah so tudi opozorila, napake ter vse akcije, ki jih je uporabnik zahteval bodisi preko uporabniškega vmesnika bodisi preko integracije v druge sisteme, in tako služijo tudi kot revizijska sled. Prav tako se v dnevnik odlagajo obvestila o delovanju ML modula.

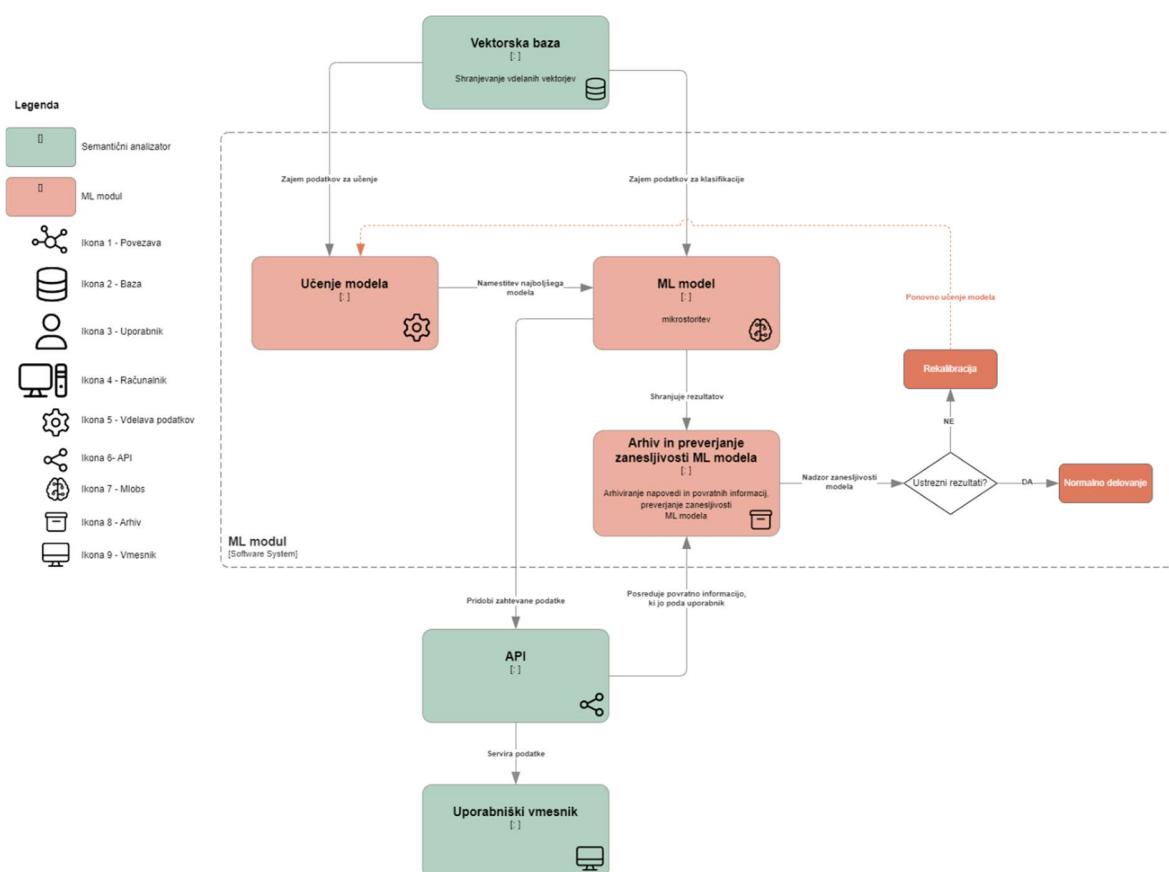
Uporabniška vmesnika za vsebinskega administratorja ter ML inženirja iz dnevnika delovanja pridobivata podatke za prikaz informacij o stanju in delovanju sistema.

Primer odprtokodne rešitve je **Prometheus** ([povezava do uradne spletne strani](#)).

4.2.7. ML modul: Opis procesa MLOps

Na Slika 32 je prikazan diagram procesa MLOps, ki je sestavljen iz naslednjih gradnikov, in sicer:

- zajem podatkov,
- učenje ML modela,
- namestitev ML modela,
- arhiviranje in nadzor modelov,
- rekalkibracija (ponovna nastavitev).



Slika 32: Proces MLOps

4.2.7.1. Zajem podatkov

Zajem podatkov za učenje se izvaja preko poizvedbe v vektorski bazi. Glaven podatek v učni množici bo vektorska reprezentacija oz. vložitev dokumenta. Proces zajema učnih podatkov je proces, ki se izvaja občasno. Na začetku se izvede prvo učenje ML modela, potem pa pri

vsaki rekaliibraciji ML modela. Rekaliibracija ML modela je avtomatizirana in se opravi glede na trenutno poslabšanje modela, ki ga meri izbrana metrika kvalitete ML modela.

Zajem podatkov za klasifikacije je stalni proces, ki se izvaja vsakič, ko je v vektorski bazi na voljo nov dokument, ki ga je potrebno uvrstiti v predhodno določene/označene skupine dokumentov.

4.2.7.2. Učenje modela

ML modul omogoča različne tipe učenja modela, in sicer:

- **Klasifikacija neoznačenih dokumentov** se izvaja v primeru neobstoja predhodno označenih dokumentov. Tukaj se je najbolj pogosto uporablja metoda TF-IDF. Ko je v dokumentni bazi na voljo nov dokument, se samodejno zažene ML model za gručenje dokumentov, ki dokument uvrsti v ustrezno skupino. Uporabnik preko uporabniškega vmesnika lahko poda povratno informacijo o pravilni klasifikaciji dokumenta. Informacija bo zapisana kot eden izmed metapodatkov vektorske reprezentacije dokumenta. Obstaja tudi možnost, da bo dokument oz. vektorska reprezentacija dokumenta tako različen od obstoječih dokumentov, da ga ne bo možno smiselno uvrstiti v eno izmed obstoječih dokumentnih gruč. V tem primeru bo uporabnik dokument ročno uvrstil v novo dokumentno gručo. Rekaliibracija ML modela nenadzorovanega učenja se izvaja bodisi takrat, ko se ugotovi, da je uvrščanje novih dokumentov nesmiselno, bodisi po predhodno nastavljenem urniku.
- **Nadzorovano učenje** upošteva povratno informacijo s strani uporabnika kot edino resnico – je bil dokument uvrščen v pravilno gručo in če ne, v katero gručo bi moral biti uvrščen? Ta proces predstavlja natančnejšo kalibracijo predhodnega procesa klasifikacije neoznačenih dokumentov in izboljšanje zanesljivosti procesa klasifikacije.

4.2.7.3. Namestitev naučenega ML modela

Naučeni modeli iz prejšnjega odstavka se namestijo na strežniški infrastrukturi naročnika kot mikrororitve. Vsak model vsebuje informacijo o njegovi različici. Različice predstavljajo naravna števila, ki se v MLOps procesu povečujejo. Za namestitev ML modelov kot mikrororitve je potrebna uporaba tehnologije zabojnikov oz. kontejnerjev ("containers"). Najbolj priljubljeno orodje, ki to tehnologijo uporablja, je Docker. Za lažjo spremljanje in upravljanje z več soodvisnimi ML modeli (več Docker zabojniki), je za MLOps proces potrebna

tehnologija naprednega spremljanja ter upravljanja z vsemi nameščenimi zabojniki. Za to tehnologijo predlagamo uporabo okolja Kubernetes.

Prav tako priporočamo MLflow (www.mlflow.org) - odprtokodno ogrodje za namestitev, verzioniranje in registriranje ML modelov.

4.2.7.4. Arhiviranje in preverjanje zanesljivosti modela

Modul za arhiviranje in preverjanje zanesljivosti modela je ključen gradnik MLOps procesa. Ta modul bo ponujal vmesnike za sledenje in beleženje modelskih parametrov, kot so koeficienti pozornosti ter modelskih metrik, kot je napaka oz. zmedenost modela, različic programskih kod, artefaktov modela ter beleženja izvajanja oz. vseh klicev do mikrorstitev modela. Spremljanje usposabljanja in uvajanja ML modela je ključnega pomena za preprečevanje zanašanja oz. stranpota modela in za zagotovitev, da model še naprej natančno odraža celotne podatke v sistemu. Zato se bo skupaj z uporabnikom določila meja vrednosti metrike zanesljivosti ML modela. Ko se bo ugotovilo, da model dokumentov v gruče ne uvršča po pričakovanju, se bo ponovno sprožil proces učenja modela (rekalibracija), ki je opisan v naslednjem odstavku.

4.2.7.5. Rekalkibracija

Postopek rekalkibracije se sproži takrat, ko se ugotovi, da ML model za uvrščanje dokumentov v dokumentne gruče dokumente v gruče ne uvršča več z zadovoljivo natančnostjo. Za spremljanje zanesljivosti bo v orodje SEMANT vključena funkcionalnost grafičnih prikazov zgodovine delovanja ML modela za klasifikacijo. Orodje SEMANT lahko ponuja dve možnosti proženja rekalkibracije, in sicer:

- **Samodejno** – v primeru presežanja predhodno določene spodnje meje dovoljenega odstopanja metrike zanesljivosti ali z uporabo določenega urnika. Spodnja meja odstopanja ter čas in dan samodejnega proženja opravila se uskladi med inženirjem strojnega učenja (zadolžen za vzdrževanje in konfiguriranje ML modula) ter vsebinskimi uporabniki orodja SEMANT,
- **Ročno** – preko grafičnega vmesnika se poda možnost ročnega proženja rekalkibracijskega procesa s strani vsebinskega uporabnika po uskladitvi z inženirjem strojnega učenja.

Pri rekaliibraciji ML modela za klasifikacijo dokumentov po podobnosti vsebine se iz metapodatkov dokumentov odstrani vsa zgodovina predhodnih uvrstitev vseh dokumentov, po rekaliibraciji modela pa izboljššan model ponovno uvrsti dokumente v dokumentne gruče. Vsebinski uporabnik ima nato možnost uvrstitev preveriti ter podati povratno informacijo o kakovosti uvrstitve dokumentov v gruče.

5. Opredelitev uporabniških vmesnikov

Večina procesov orodja SEMANT za svoje delovanje ne potrebuje interakcije z uporabnikom, saj je največja dodana vrednost orodja integracija v obstoječe sisteme s pomočjo REST vmesnika. Potrebno pa je razviti uporabniški vmesnik za vnos novih dokumentov, iskanje podobnih dokumentov ter napredno poenotenje besedil, vmesnik za upravljanje z ML modulom ter vmesnik za nadzor orodja.

Vsi uporabniški vmesniki naj bodo implementirani v obliki spletne aplikacije z ustreznimi zalednimi moduli. Ključne uporabnikove odločitve ter akcije se sproti beležijo v zaledni dnevniški sistem. Tako je zagotovljena revizijska sled.

5.1. Uporabniški vmesnik za vnos novih dokumentov, pridobivanje in validacijo podobnih dokumentov ter napredno poenotenje besedil

Uporabniški vmesnik za vnos novih dokumentov ter iskanje podobnih dokumentov je namenjen vsebinskim uporabnikom. Vmesnik podpira naslednje procese:

- avtentikacijo uporabnika preko eksternih orodij (SI-PASS, KeyCloak)
- vnos novega dokumenta v orodje SEMANT ter validacijo vnosa novega dokumenta v dokumentno gručo (proces opisan v 8.1)
- pridobivanje, pregled ter validacijo podobnih vsebin (proces opisan v 8.2)
- napredno poenotenje besedil (proces opisan v 8.3)

Ob prijavi v uporabniški vmesnik se s pomočjo zalednih (backoffice) modulov ogrođa SEMANT uporabnik najprej avtenticira. Glede na uporabniško vlogo vsebinskega uporabnika se uporabniku prikažejo funkcionalnosti vmesnika, do katerih lahko dostopa.

5.1.1. Vnos dokumentov in validacija uvrstitve dokumenta v gručo

Pri vnosu novega dokumenta v orodje SEMANT uporabnik preko vmesnika izbere dokument, dostopen na uporabnikovi napravi. Uporabnik ima možnost predogleda dokumenta, ki ga je naložil. Po predogledu uporabnik potrdi ali zavrne uvoz dokumenta v orodje. Nazadnje

prejme po uvozu uporabnik povratno sporočilo. V primeru neuspešnega uvoza uporabnik dobi sporočilo o napaki, v primeru uspešnega uvoza pa tudi informacijo o nazivu dokumentne gruče, v katero se je dokument ob uvozu uvrstil.

Ta uporabniški vmesnik podpira tudi pregled dokumentov po gručah ter validacijo uvrstitve dokumentov v gručo. Funkcionalnost validacije uvrstitve dokumenta v gručo ni na voljo vsakemu vsebinskemu uporabniku, ampak tistim v ustreznih uporabniških vlogah. Uporabnik dokument, ki ga želi validirati, poišče glede na ključne besede oz. ostale identifikatorje. Med seznamom najdenih dokumentov izbere ustreznega. Ta dokument se na vmesniku prikaže skupaj s podatkom, v katero dokumentno gručo je uvrščen. Uporabnik ima možnost uvrstiti dokument v eno izmed trenutno aktivnih dokumentnih gruč ter to odločitev shrani.

5.1.2. Pridobivanje, pregled ter validacija podobnih vsebin

Uporabnik najprej med obstoječimi dokumenti s pomočjo ključnih besed poišče izvorni dokument, za katerega želi pridobiti podobne vsebine. Celotni izvorni dokument se pred pridobivanjem podobnih vsebin uporabniku po potrebi tudi prikaže. Uporabnik potrdi, da je dokument pravilen, ter pred pošiljanjem zahteve po pridobitvi podobnih vsebin določi še, iz katerih virov javnoupornih sektorjev oz. podsektorjev se naj pridobijo podobne vsebine.

Po pridobitvi rezultatov poizvedbe se uporabniku prikažejo podobne vsebine, uporabnik pa ima možnosti validirati rezultate. Povratna informacija o kakovosti rezultatov poizvedbe se pošlje v vektorsko bazo, kjer se zapiše kot metapodatek v vektorski zbirki podatkov.

5.1.3. Napredno poenotenje vsebin

Ta uporabniški vmesnik uporabniku najprej omogoči vnos osnutka besedila, ki ga želi poenotiti z obstoječimi besedili v ogrodju SEMANT. Na vmesniku se prikaže vneseni dokument z označenimi ključnimi besedami, ki so kandidati za poenotenje. Za vsako besedo nato uporabnik izbere ustrezno sopomenko iz prikazanih možnosti ter potrdi izbiro. Po izbiri ustrezne sopomenke za vsako ključno besedo uporabnik potrdi spremembe v dokumentu, informacija o izbrani sopomenki pa se za vsako ključno besedo shrani kot metapodatek v vektorski bazi.

5.2. Uporabniški vmesnik za upravljanje z MLOps modulom

Uporabniški vmesnik za upravljanje z MLOps modulom je namenjen ML inženirjem. Uporabniški vmesnik ML inženirju omogoča:

- pregled zgodovine zagonov ter metrik modelov,
- pregled in spremembo nastavitev samodejne rekaliibracije modelov (nastavitve urnika rekaliibracije modelov, spodnjih mej modelskih metrik, ki prožijo samodejno rekaliibracijo modela),
- administracijo modelov (pregled modelskih verzij, možnost zamenjave aktivnega modela z drugo verzijo modela) ter
- ročno proženje rekaliibracije modelov.

5.3. Uporabniški vmesnik za nadzor orodja SEMANT

Uporabniški vmesnik za nadzor orodja SEMANT je namenjen vsebinskim administratorjem. Prvenstveno je namenjen spremljanju kakovosti modelov ter revizijske sledi dela vsebinskih uporabnikov z ogrođjem SEMANT.

Kakovost modelov je mogoče spremljati preko grafikonov, ki prikazujejo časovni razvoj posameznik metrik za spremljanje kakovosti modela, skupaj s prikazom vrednosti trenutnih spodnjih mej teh metrik, ki prožijo avtomatsko rekaliibracijo ter trenutni način avtomatske kalibracije (urnik, vrednosti spodnjih mej metrik kakovosti modela)

Vsebinski administrator ima samo možnost pregleda kakovosti modelov. Proženje rekaliibracije ter nastavitev parametrov opravi uporabnik v vlogi ML inženirja preko uporabniškega vmesnika za upravljanje z MLOps modulom.

Revizijsko sled dela vsebinskih uporabnikov vsebinski administrator pregleduje na nivoju akcij v ogrođju SEMANT. Zaradi velike količine teh podatkov je vsebinskemu administratorju na voljo filtriranje po časovnem oknu dogodkov, uporabnikih ter tipih dogodkov.

6. Fizična arhitektura

Fizična arhitektura informacijske rešitve je odvisna od zasnove informacijske rešitve, ki jo bo predlagal izvajalec. Predlagane komponente zasnovane informacijske rešitve mora izvajalec uskladiti z ekipo državnega računalniškega oblaka in državnega omrežja HKOM. Po uskladitvi se opredeli fizična arhitektura in vključi v dokument PZI.

7. Strojna in programska oprema

7.1. Specifikacija strojne opreme

Za vzpostavitev vektorske baze in učenje strojnih učnih modelov na veliko število dokumentov različnih sistemov javne uprave Republike Slovenije bi bila primerna uporaba grafične kartice NVIDIA L40S. Ta nudi neprekosljivo zmogljivost (izvajanja algoritmov) UI in grafike, prilagojeno za potrebe podatkovnih centrov. Grafična kartica L40S temelji na arhitekturi NVIDIA Ada Lovelace in zagotavlja pomembno izboljšanje zmogljivosti pri večopravilnosti, vključno z inferenco (sklepanje oz. rezultat modela na podlagi vnosnih podatkov) in učenjem velikih jezikovnih modelov (LLM) ter aplikacijami za grafiko in video.

NVIDIA L40S izstopa s svojimi jedri četrte generacije Tensor, ki podpirajo hitra preračunavanja strukturno redkih podatkov in optimiziran format TF32, kar zagotavlja povečanje zmogljivosti za usposabljanje (učenje) modelov UI in podatkovne znanosti. Njena jedra tretje generacije RT izboljšujejo zmogljivost sledenja žarkom, kar je koristno za naloge, ki zahtevajo visoko kakovostne vizualizacije. Za splošne računske naloge, kot so razvoj 3D modelov in simulacije računalniško podprtega inženiringa, se izkaže visoka učinkovitost pospešenega pretočnega računanja v enojni natančnosti njenih jeder CUDA.

Poleg tega grafika L40S podpira NVIDIA Transformer Engine, ki bistveno izboljšuje zmogljivost UI z inteligentnim pregledovanjem in samodejnim preklapljanjem med natančnostmi FP8 in FP16 v nevronskih mrežah arhitekture transformer. To omogoča hitrejšo zmogljivost UI in pospešuje tako usposabljanje kot inferenco.

Kar zadeva učinkovitost in varnost podatkovnega centra, je GPU L40S optimiziran za neprekinjeno delovanje in izpolnjuje najnovejše standarde z lastnostmi, kot sta zagon iz varnega vira in tehnologija korena zaupanja, kar dodaja dodatno plast varnosti za podatkovne centre.

Pri primerjavi L40S z NVIDIA H100 prvi nudi stroškovno učinkovito alternativo z nekoliko nižjo zmogljivostjo, a značilnimi prihranki v smislu stroškov. Prav tako je poročano, da je L40S lažje dostopna kot H100, kar bi lahko bilo koristno za pravočasno izvedbo projekta.

Ob upoštevanju teh dejavnikov bi NVIDIA L40S GPU lahko zagotovila potrebno zmogljivost za vse naloge UI in ML, z ravnovesjem med stroški in dostopnostjo, ki bi lahko ustrezalo

zahtevam projekta. Za najnovejše specifikacije in podrobnosti o zmogljivosti je treba preučiti uradne podatkovne liste in informativne liste, ki jih zagotavlja NVIDIA.

Specifikacija infrastrukture za **NVIDIA L40S 8xGPU 48GB, 1.5TB RAM, 2x Genoa Epyc CPU, 5x 1.9TB SSD-je, konfigurirani v RAID6 + 1 spare**, ki bi bila primerna za potrebe Semantičnega analizatorja, je prikazana na Slika 33.

NVIDIA High End PCIe Inference and Training GPU Options					
Model	A100 (80GB)	L40	L40S	H100 (80GB)	H100 NVL
GPU Architecture	Ampere	Ada Lovelace	Ada Lovelace	Hopper	Hopper
GPU Memory	80GB	48GB	48GB	80GB	188GB
Memory Bandwidth	1555 GB/s	864 GB/s	864 GB/s	2 TB/s	7.8 TB/s
CUDA Cores	6912	18176	18176	14592	14592
RT Cores		142	142		
RT Core Performance TFLOPS		209	209		
Tensor Cores	432	568	568	576	576
FP64 TFLOPS	9.7			26	68
FP64 Tensor Core TFLOPS	19.5			51	134
FP32 TFLOPS	19.5	90.5	91.6	51	134
TF32 Tensor Core TFLOPS (with Sparisty)	156 (312)	90.5 (181)	183 (366)	378 (756)	989 (1979)
BFLOAT16 Tensor Core TFLOPS (with Sparisty)	312 (624)	181.05 (362.1)	362.05 (733)	756 (1513)	1979 (3958)
FP16 Tensor Core TFLOPS (with Sparisty)	312 (624)	181.05 (362.1)	362.05 (733)	756 (1513)	1979 (3958)
FP8 Tensor Core TFLOPS (with Sparisty)		362 (724)	733 (1466)	1513 (3026)	3958 (7916)
Peak INT8 TOPS (with Sparisty)	624 (1248)	362 (724)	733 (1466)	1513 (3026)	3958 (7916)
Peak INT4 TOPS (with Sparisty)	1248 (2496)	724 (1448)	733 (1466)	1513 (3026)	3958 (7916)
NVLink	Yes			Yes	Yes
Display Ports		4x DP 1.4a	4x DP 1.4a		
Max Power Consumption (W)	300	300	350	350	800
Virtual GPU (vGPU) Software Support	Yes	Yes	Yes	Yes	Yes
vGPU Support	Yes	Yes	Yes	Yes	Yes
MIG Support	Yes	No	No	Yes	Yes
PCIe Generation	PCIe Gen4 x16	PCIe Gen4 x16	PCIe Gen4 x16	PCIe Gen5 x16	PCIe Gen5 x16

Slika 33: Specifikacija infrastrukture za NVIDIA L40S 8xGPU 48GB, 1.5TB RAM, 2x Genoa Epyc CPU, 5x 1.9TB SSD-je, konfigurirani v RAID6 + 1 spare

Ta specifikacija zagotavlja, da je NVIDIA L40S optimalna izbira za obdelavo velikega števila dokumentov z uporabo strojnega učenja in vektorskih baz v podatkovnih centrih, pri čemer ponuja izjemno zmogljivost za AI, grafiko in video aplikacije.

7.2. Specifikacija programske opreme

Pri razvoju semantičnega analizatorja, ki bo služil kot napredno orodje za obdelavo in analizo velikih količin tekstovnih podatkov iz različnih virov, je predlagana uporaba dveh ključnih tehnologij: Java Spring za zaledni sistem in React za čelni sistem. Ta kombinacija tehnologij

je izbrana zaradi svoje robustnosti, fleksibilnosti in široke podpore skupnosti, kar bo omogočilo razvoj zanesljive in učinkovite aplikacije.

Zaledni sistem z Java Spring: Java Spring predstavlja enega izmed vodilnih ogrodij za razvoj aplikacij v Javi, ki omogoča gradnjo visoko zmogljivih, varnih in lahko vzdržljivih spletnih aplikacij. S svojim obsežnim naborom modulov, vključno s Spring Boot, Spring Data, Spring Security in drugimi, ponuja vse potrebno za hitro in učinkovito implementacijo kompleksnih zalednih rešitev. Njegova glavna prednost je v avtomatizaciji konfiguracije in lažjem upravljanju odvisnosti, kar znatno pospeši razvojni proces. Uporaba Java Spring za zaledni sistem bo zagotovila trdno osnovo za obdelavo podatkov, avtentikacijo uporabnikov, upravljanje sej in interakcijo z bazami podatkov.

Čelni sistem z React: React, knjižnica za izgradnjo uporabniških vmesnikov, razvita s strani Facebooka, je znan po svoji hitrosti, fleksibilnosti in učinkovitosti pri gradnji dinamičnih in odzivnih spletnih aplikacij. React omogoča razvoj kompleksnih uporabniških vmesnikov s preprostim in deklarativnim načinom programiranja, ki izboljša produktivnost in olajša razvoj. Z uporabo komponentnega modela React omogoča ponovno uporabo kode, kar pripomore k hitrejšemu razvoju in enostavnejšemu vzdrževanju aplikacij. Integracija React za čelni sistem bo omogočila razvoj intuitivnega in privlačnega uporabniškega vmesnika, ki bo uporabnikom zagotavljal prijetno izkušnjo pri delu s semantičnim analizatorjem.

Integracija in delovanje: Integracija med Java Spring zalednim sistemom in React čelnim sistemom bo potekala preko REST API-jev, kar omogoča ločeno razvojno pot za obe komponenti ter zagotavlja fleksibilnost in možnost neodvisnega nadgrajevanja posameznih delov aplikacije. Ta pristop ne samo da olajša razvoj in testiranje, ampak tudi omogoča boljšo skalabilnost in varnost celotne aplikacije.

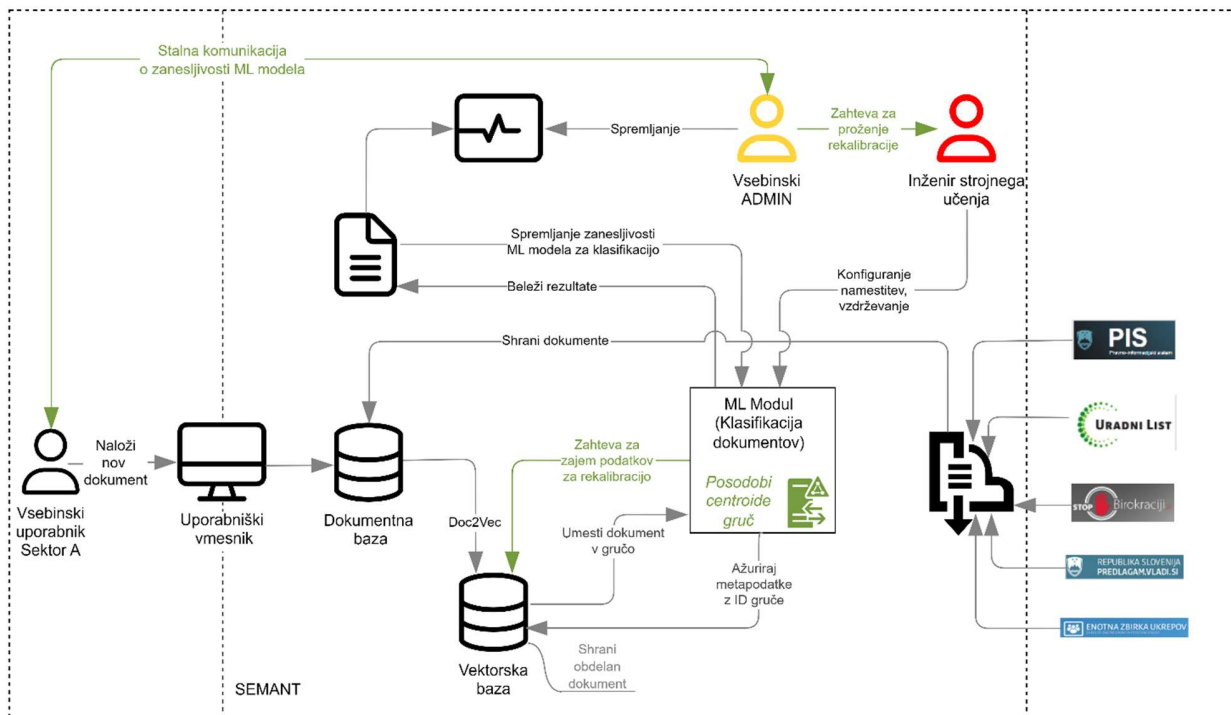
S tem pristopom k razvoju semantičnega analizatorja se lahko ustvari robustna, zanesljiva in uporabniku prijazna rešitev, ki bo služila kot močno orodje za analizo in obdelavo tekstovnih podatkov.

8. Upravljalški procesi

8.1. Proces za shranjevanje novih vsebin

Obstajata dva načina nalaganja oz. shranjevanja novih vsebin, in sicer:

- **Ročno** - vsebinski uporabnik ročno naloži dokument preko uporabniškega vmesnika aplikacije SEMANT.
- **Samodejno** - preko storitvenega oz. komunikacijskega vodila, v katerega bodo integrirani različni zunanji sistemi, ki vsebujejo vire besedil.



Slika 34: Proces za shranjevanje novih vsebin

Proces za shranjevanje novih vsebin je prikazan na diagramu na Slika 34.

Ko je dokument oz. besedilo bodisi preko ročnega ali samodejnega vnosa na voljo v orodju SEMANT, se najprej shrani v dokumentni bazi. ML modeli so modeli, ki za svoje izračune potrebujejo numerične podatke in ne besedila, zato je dokument vdelan v numerično obliko s pomočjo algoritma Doc2Vec in nato v tej obliki shranjen v vektorsko bazo.

Prva klasifikacija bo enaka, kot je bila določena že v pilotnem projektu, in sicer z nenadzorovanim ML. Klasifikacijo dokumenta se določi tako, da se najprej preberejo teksti posameznih dokumentov, nad katerimi bo izveden standardni proces podatkovnega čiščenja. To čiščenje vsebuje operacije, kot so na primer izločitev vejic, klicajev ali drugih ločil, ki ne vplivajo na podatkovno analizo. K tem spada tudi spreminjanje velikih črk v male. Čiščenje pripomore k čim manjšemu vektorskemu prostoru v vektorski bazi, kar pomeni večjo učinkovitost hrambe podatkov ter izvajanja poizvedb.

Ker že obstaja veliko knjižnic, ki so sicer namenjene dokumentom v angleškem jeziku, je ena izmed možnosti implementacije ta, da se najprej vsi dokumenti prevedejo v angleščino in nato obdelajo s temi knjižnicami. Drugi način prevajanje dokumentov iz slovenščine v angleščino je lahko z uporabo lokalno nameščenih (brez API klicev izven HKOM) odprtokodnih velikih jezikovnih modelov (angl. Large language models; preveriti bi bilo treba tudi model SloBERTa).

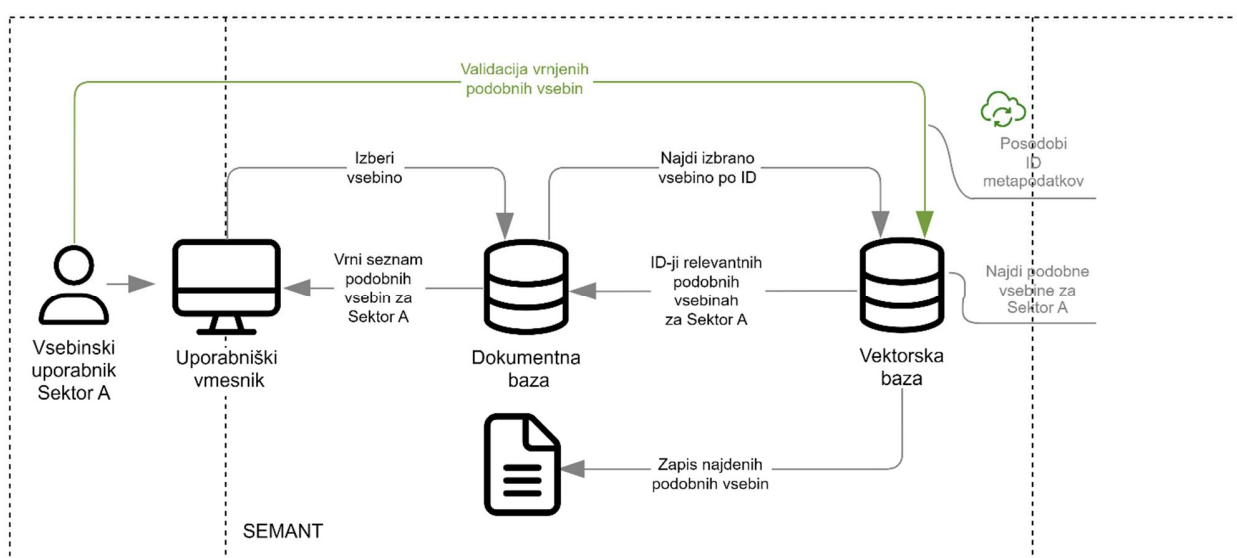
V prvih fazah bo za klasifikacijo teh dokumentov uporabljeno nenadzorovano učenje oz. metoda gručenja, ki bo na začetku samo prepis grupiranja dokumentov oz. določeno uvrščanje iz pilotnega projekta. V nadaljevanju procesa bodo modeli nenadzorovanega učenja delovali tako, da na podatkih izvajajo prevedbe v druge manjše dimenzije, ki se lahko vizualizirajo in se nato iščejo smiselni vzorci v točkah, ki so bližje ena drugi. Algoritem s pomočjo metriko za razdalje (kosinusna razdalja, evklidska razdalja ipd.) določi, kateri gruči je novi dokument najbližji, in ga ustrezno uvrsti. Po vsakem novem uvrščenem dokumentu se bodo centroidi gruči ponovno izračunali. Centroid gruče je najbolj reprezentativen predstavnik celotne gruče. To je točka znotraj gruče, ki je kumulativno najmanj oddaljena od vseh drugih točkah v tej gruči. Naslednja stopnja bo validacija novo uvrščenih dokumentov s strani vsebinskega uporabnika. S tem bo povratna informacija o pravilnosti uvrstitve dokumenta v gručo uporabljena v naslednjem ciklu učenja algoritma. Vsaka validacija pomeni bolj zanesljiv naslednji vrnjen rezultat s strani algoritma.

Vsak dokument bo imel svoj edinstveni identifikator, ki bo enak tako v dokumentni kot v vektorski bazi. Id iz dokumentne baze bo uporabljen za poizvedbe za prikaz dokumenta na uporabniškem vmesniku. V vektorski bazi bodo poleg Id dokumenta, kot primarnega ključa, shranjeni tudi metapodatki. V metapodatkih bo shranjena informacija o predhodno določeni

šifri oz. identifikatorju gruče, v katero sodi shranjen dokument, ter informacija o pravilnosti klasifikacije oz. povratni informaciji s strani vsebinskega uporabnika.

Vsako delovanje algoritma za gručenje oz. vsako uvrščanje novih prihajajočih dokumentov bo beleženo v zbirke za beleženje rezultatov (angl. Logs), ki so potem ključne za iskanje morebitnih nedoslednosti delovanja sistema ter ugotavljanje slabih napovedih ML modela za gručenje.

8.2. Proces za pridobivanje podobnih vsebin



Slika 35: Proces za pridobivanje novih vsebin

Preko uporabniškega vmesnika orodja SEMANT lahko vsebinski uporabnik izbere vsebino oz. besedilo, za katero si želi pridobiti seznam podobnih vsebin (Slika 35). Izbrano vsebino preko grafičnega vmesnika se po ID najde v vektorski bazi. Potem se bo z algoritemsko poizvedbo v bazi poiskalo podobne vsebine. Obstajata dva načina poizvedb, in sicer:

- **Splošno poizvedbo** - iskanje podobnosti po vseh vsebinah oz. po vsebinah iz različnih sektorjev javne uprave in
- **Namensko poizvedbo** - iskanje podobnosti po vsebinah določenega sektorja vsebinskega uporabnika.

Iskanje podobnih vsebin je proces, ki je zasnovan na algoritmu Approximate Nearest Neighbour Search (ANNS). V vektorski bazi se s pomočjo metrike za razdaljo (kosinus, evklidska, Manhattan ipd.) poiščejo najbližji »sosedni« vektorja vhodnega besedila. Slednji se vrnejo nazaj kot seznam identifikatorjev tj. Id-ji, ki se prek poizvedb v dokumentni bazi pokažejo na grafičnem vmesniku končnega vsebinskega uporabnika. Vsebinski uporabnik bo moral pogledati vrnjen seznam podobnih besedil in validirati pravilnost vrnjenih rezultatov (označi, katera besedila so pravilno izbrana in katera niso). Na podlagi povratne informacije oziroma validacije s strani vsebinskega uporabnika se bodo vsi identifikatorji podobnih vsebin v metapodatkih posodobili, da bi lahko pri nadaljnjih poizvedbah algoritem vrnil čim bolj pravilne zadetke.

Vsaka poizvedba in validacija se beleži v zbirko beleženja procesov aplikacije SEMANT.

8.3. Proces za napredno poenotenje oz. izbira enotnih besedil v Službi za zakonodajo

Besede iščemo po istem pomenu s pomočjo **Stematizacije/Lematizacije**.

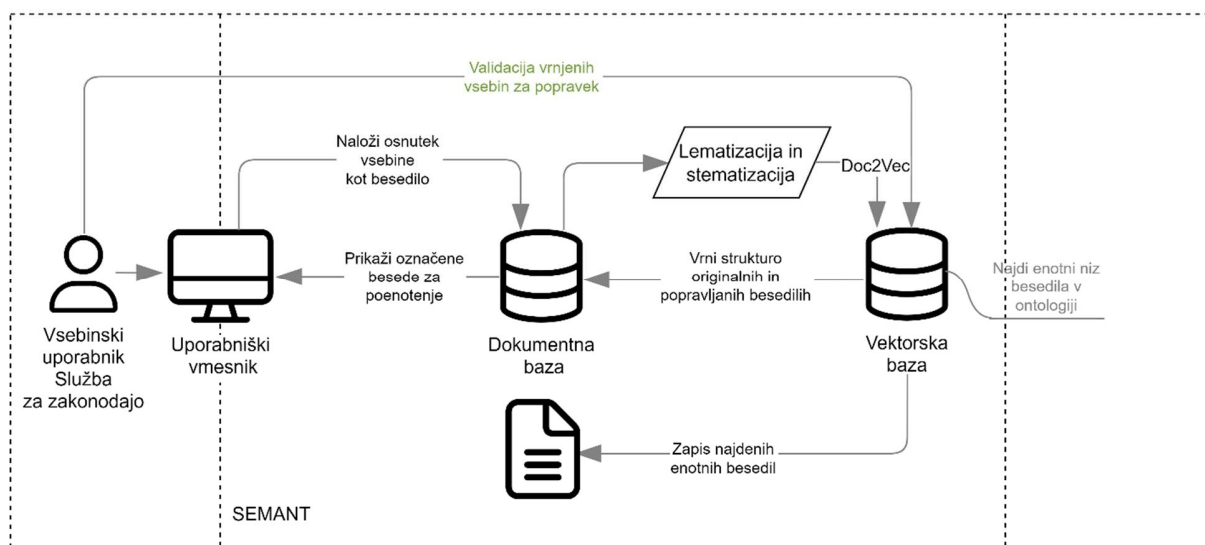
Kot pri vsaki tekstovni obdelavi bi tudi pri iskanju ključnih oziroma podobnih besed najprej poskrbeli za čiščenje teksta. Tokenizacija je naslednji korak, npr. besedna tokenizacija, na podlagi katere se lahko pripiše določeni besedi stavčni člen (POS označevanje). Korak, ki bi nato sledil, je uporaba lematizacije ali stematizacije. Stematizacija je bolj preprosta oblika združevanja besed, ki besedi vzame koren besede npr. korenje > koren, kar pa je v določenih primerih lahko dvoumno, saj so takšne pretvorbe kontekstno nepovezane. Lematizacija je v tem primeru boljša metoda, saj ne vzame korena besede ampak poskuša izveči enak pomen, na primer: dober > dobro ali rekel > reči. Na ta način bi lahko ugotovili podobne oziroma kontekstno enake besede, ki bi jih nato pokazali kot predloge uporabniku.

Besede vdelamo v vektorje s pomočjo **word2vec**:

Besede prevedemo v vektorski prostor, ki je že pred definiran. Vsaka nova beseda v novem dokumentu tako dobi svojo pozicijo v vektorskem prostoru. Vektorski prostor besed je postavljen tako, da imajo podobne besede podobno smer in velikost vektorja. Prednost tega

prostora je, da lahko poiščemo sorodne besede, kot so npr. avtocesta, cesta, pot, kolovoz... in s tem naše iskanje ključnih besed razširimo na druge kontekste v dokumentih.

Vse rezultate, ki jih dobimo s pomočjo modelov, lahko tako uporabimo na dva načina. Prvi je, da uporabimo besedo za iskanje po ostalih dokumentih in s tem lahko pregledamo kontekste, ki so podobni, in drugi, predlagamo besedo, ki je bolj primerna določenemu kontekstu. Proces za napredno poenotenje je prikazan na Slika 36.



Slika 36: Proces za napredno poenotenje

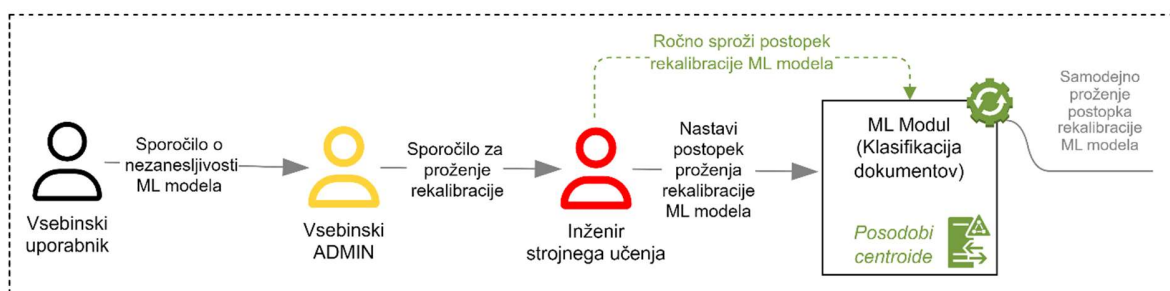
8.4. Proces za proženje postopka rekaliibracije ML modela

Zbirka beleženja rezultatov je ključnega pomena v procesu MLOps, ker se bodo vsi ML modeli nadzorovali preko vmesnika s prikazovanjem grafikona trenda metrike zanesljivosti modela.

Vsebinski uporabnik bo oseba, zadolžena za pregled, iskanje in nalaganje novih vsebin ter validacijo vrnjenih rezultatov uvrščanja. Če bo vsebinski uporabnik zaporedoma dobival nezanesljive rezultate uvrščanja, bo sporočil vsebinskemu administratorju.

Vsebinski administrator bo oseba, ki bo spremljala in nadzirala grafikon o zanesljivosti modela in ko se bo ugotovilo, da je vrednost metrike z zanesljivostjo pod predhodno določeno mejo, bo sporočil skrbniku ML modula tj., inženirju strojnega učenja, da sproži postopek rekaliibracije modela (Slika 37).

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



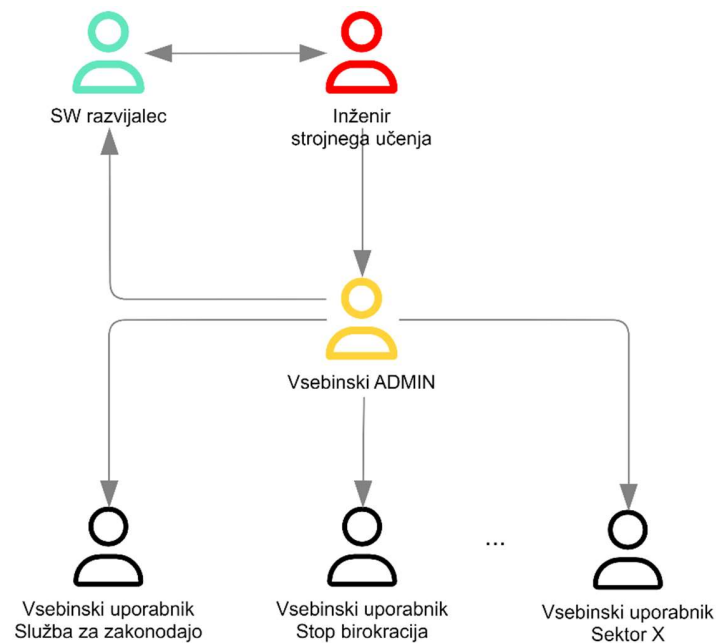
Slika 37: Proces za sproženje rekalkibracije ML modela

Rekalibracija modela je postopek, ki se odvija algoritemsko tj. kot del ML modula. Z rekalibracijo se ponastavljajo predhodne anotacije vseh dokumentov in se nato potem naredi ponovno gručenje dokumentov. Število različnih tipov dokumentov oz. klas, v katere sodijo dokumenti, je dinamično oz. se lahko spremeni pri vsaki rekalibraciji. Ko se bo rekalibracija izvedla, je vsebinski uporabnik dolžan ponovno pregledati in validirati gručen dokumente. Pričakovano je, da bo veliko predhodnih uvrstitev enakih kot v prejšnjem stanju, ampak je tudi pričakovano in ključno, da se dodeli nova uvrstitev, če so predhodno prišli dokumenti, ki niso bili pravilno uvrščeni oz. niso sodili v nobeno od predhodnih klas dokumentov.

8.5. Korespondenca med uporabniki in skrbniki sistema SEMANT

Zaradi brezhibne uporabe ter nadaljnjega vzdrževanja aplikacije predlagani sistem SEMANT potrebuje jasno in razumljivo korespondenco med vsemi uporabniki znotraj aplikacije. Na Slika 38 je prikazan diagram korespondence znotraj aplikacije SEMANT.

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco



Slika 38: Korepondenca med uporabniki in skrbniki sistema SEMANT

9. Nadaljnji razvoj informacijske rešitve (možni predlogi)

9.1. Proces za izdelavo povzetka dokumenta

Veliki jezikovni modeli (LLM), kot so npr. GPT-4, LLaMa2, Mistral 7B, Zephyr-7B se lahko uporabljajo za tekstovno povzemanje. Začne se z izvirnim besedilom, ki ga vsebinski uporabnik preko uporabniškega vmesnika naloži v aplikaciji SEMANT. Dokument se shrani v dokumentno bazo podatkov. Ker je besedilo lahko preobremenjeno z nepotrebnimi informacijami ali oblikovnimi elementi, se nadaljuje s predobdelavo besedila, ki vključuje čiščenje besedila nepotrebnih presledkov, ločil, posebnih znakov, spletnih povezav itd. Besedilo se lahko razdeli na manjše segmente ali odstavke (ang. Chunks), če je predolgo. Slednje se potem s pomočjo vložitvenega algoritma Doc2Vec pretvori v večdimenzijski numerični vektor, ki se potem shrani v vektorski bazi skupaj s svojimi metapodatki za nadaljnjo uporabo. Besedilo se preko prompt engine-a (vnosne oblike) prenese v model z določenim ukazom ali pozivom, na primer: »Povzemi naslednje besedilo:«.

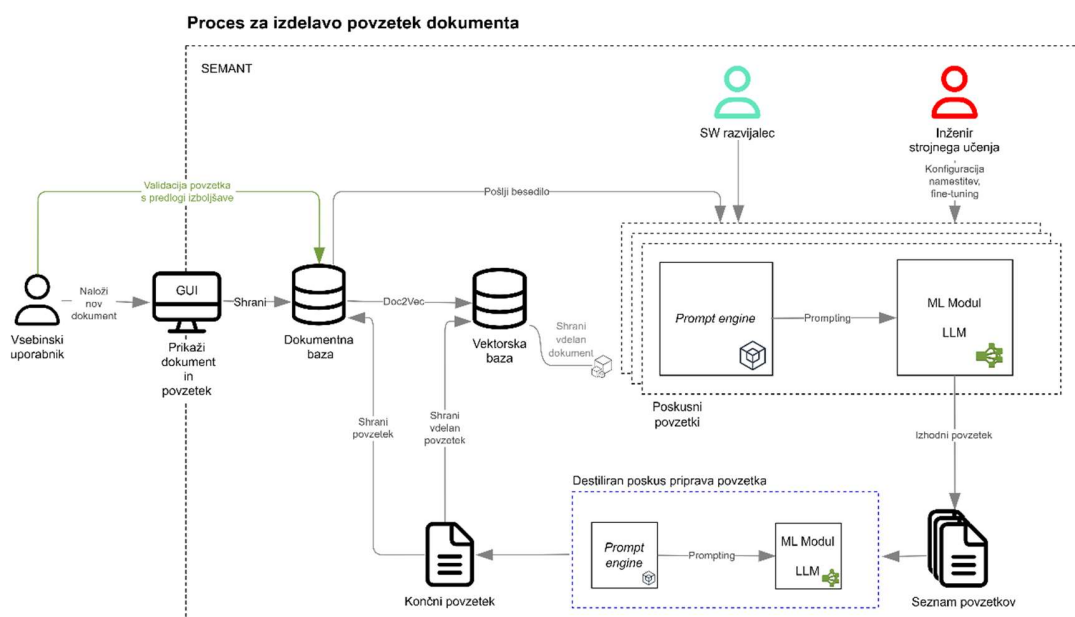
Ta postopek se ponovi večkrat, z različnimi pozivi. Model, kot je GPT-4, ne potrebuje posebnega usposabljanja za nalogo povzemanja, saj je že usposobljen na obsežnem naboru podatkov in se lahko prilagaja različnim nalogam. Ko prejme besedilo, model poskuša razumeti njegovo vsebino in izluščiti ključne informacije. Na podlagi svojega »razumevanja« izvirnega besedila model generira kratek povzetek. Ta povzetek je pogosto krajši od izvirnega besedila, vendar skuša ohraniti glavne informacije ali poudarke.

Povzetek se lahko še dodatno obdela, da se zagotovi kakovost, doslednost in berljivost. To vključuje preverjanje pravopisa, slovnic in stilskih elementov. V primeru več ukaznih pozivov s strani prompt engine-a, se na koncu naredi končni povzetek preko zadnjega »destilirane« poskusa.

Končni povzetek je nato predstavljen uporabniku in se prav tako shrani kot zasebno besedilo v dokumentni bazi s povezavo preko tujega ključa do originalnega dokumenta. Istočasno se naredi vložitev povzetka dokumenta in se nato shrani v vektorski bazi, podobno kot v dokumentni bazi s povezavo do vložitve originalnega dokumenta.

Koncept arhitekturne in funkcionalne zasnove informacijske rešitve za semantično obravnavo naravnega jezika z umetno inteligenco

LLM, kot GPT-4, so zelo močni modeli, vendar je vedno priporočljivo, da končni povzetek pregleda človek, da zagotovi točnost, ustreznost in kakovost informacij. Zaradi tega bo potrebno, da vsebinski uporabniki naredijo validacijo pravilnosti pridobljenega povzetka dokumenta, ki ga je na začetku naložil uporabnik. Diagram procesa dokumentnega povzemanja je opisan na Slika 39.



Slika 39: Diagram procesa za izdelavo povzetek dokumenta

9.2. Detekcija anomalij oz. kontradikcij

Kontradikcije se definira na nekaj primerih, katere nato s pomočjo uporabnikov označimo. Označimo tudi povedi, ki niso kontradiktorne. Po znanem načinu tako kot v prejšnjih korakih najprej besedila pretvorimo v vektorski prostor. Za nekaj primerov se naredi EDA analiza in se ugotovi, po katerih pravilih se kontradiktornost odraža v podatkih. Pomembna bo tudi določitev ključnih besed, ki se bodo uporabile kot referenca v povedi. Referenca bo določala, ali se določen kontekst zanika, omejuje ali enači. Primer ključne besede Delavec: *Delavec lahko dela več kot 40 ur tedensko,...* *Delavec ne sme delati več kot 40 ur tedensko...*

Prva metoda, da zaznamo kontradiktornost je, da se pogleda strukturo stavčnih členov in njihovo oddaljenost od ključne besede. Običajno se negacije v povedih odražajo s prislovi ali pomožnimi glagoli. V tem primeru je oddaljenost besede 'Delavec' od prislova 'lahko' samo eno besedo in enako je pri drugi povedi. S POS označevanjem tako lahko pridemo do pozicije besed posameznih stavčnih členov in tako ugotovimo, ali je na ključnih besedah oddaljenost pozitivnih in negativnih prislovov majhna. Tako lahko trdimo, da obstaja možnost, da so določene povedi ali stavki kontradiktorni. Seveda pa obstajajo tudi druga pravila, ki jih lahko upoštevamo pri iskanju kontradiktornih stavkov.

Druga metoda za zaznavanje kontradiktornosti je označevanje kontradiktornih in nekontradiktornih povedi, ki bi jih nato uporabili pri uporabi ML modelov z nadzorovanim učenjem. Seveda ta metoda zahteva veliko podatkov, ki jih v začetku ne bomo imeli, ali pa jih bo potrebno ročno označiti, zato si lahko kot pripomoček pomagamo s prvo metodo.

Result d.o.o., Celovška 182, 1000 Ljubljana

E: info@result.si

T: +386 01 542 17 80

result.si | result.eu