Delotoki za nadaljnje analize nestandardne slovenščine

Matej Martinc, Senja Pollak, Ana Zwitter Vitez

Izvleček

V zadnjih letih se je razmahnil razvoj platform, ki poenostavljajo znanstveno raziskovanje, povezujejo različna področja in omogočajo lažjo dostopnost metod in rezultatov širšemu krogu uporabnikov. V raziskavi, ki jo predstavljamo v tem poglavju, smo v platformo za vizualno programiranje ClowdFlows implementirali množico orodij (gradnikov) za obdelavo naravnega jezika, ki bodo jezikoslovcem in ostalim zainteresiranim uporabnikom omogočila lažjo in hitrejšo analizo besedil. Novi gradniki omogočajo obdelavo naravnega jezika, upravljanje korpusa ter končni prikaz statistik in analiz. Implementacijo novih gradnikov predstavimo na primeru dveh delotokov. Prvi je namenjen izdelavi novega korpusa iz tvitov, zbranih s pomočjo orodja *TweetCaT*, drugi pa analizi komentarjev Evrovizije. Razvita spletna orodja omogočajo gradnjo in obdelavo novih korpusov, razširjajo možnosti kvantitativnih analiz ter poenostavljajo zapletene postopke za obdelavo naravnega jezika.

Ključne besede: obdelava naravnega jezika, orodja za korpusno analizo, vizualno programiranje, ClowdFlows, nestandardna slovenščina

1 UVOD

Interdisciplinarnost je eden ključnih pogojev za izboljšanje kvalitete in kvantitete znanstvene produkcije. V zadnjem času je bilo veliko truda vloženega v razvoj platform za interdisciplinarno sodelovanje, glavni namen tovrstnih platform pa je poenostavitev in pospešitev znanstvenega raziskovanja in s tem lažja dostopnost metod posameznih področij širšemu krogu uporabnikov. V članku predstavimo implementacijo orodij za obdelavo naravnega jezika v platformo za vizualno programiranje z imenom *ClowdFlows* (Kranjc et al. 2012). *ClowdFlows* je spletna platforma za rudarjenje podatkov, namenjena poenostavitvi razvoja kompleksnih metod in procesov rudarjenja podatkov. Platforma omogoča uporabo metod tudi raziskovalcem s področja humanistike in družboslovja, ki bi jim bilo zaradi pomanjkljivega tehničnega znanja takšno raziskovanje sicer onemogočeno.

Zaradi navedenih razlogov je platforma *ClowdFlows* zasnovana za preprosto uporabo, dosegljiva preko spletnega brskalnika in ne potrebuje predhodne namestitve. Paradigma vizualnega programiranja poenostavi uporabo in izdelavo zapletenih procesov na upravljanje gradnikov (angl. *widgets*) s preprosto operacijo *primi-odloži* (angl. *drag-and-drop*). Gradniki so vizualno predstavljeni deli programov, ki imajo definirane vhodne in izhodne oblike podatkov, parametre pa lahko uporabnik ročno izbere. Te gradnike se na delovni površini sestavlja v delotoke (angl. *workflows*), ki jih lahko opišemo kot vizualne predstavitve znanstvenih procesov. Prednost platforme *ClowdFlows* je tudi, da omogoča preprosto deljenje in ponovljivost rezultatov ter ponovno uporabo delotokov (z objavo spletne strani delotoka). Platforma *ClowdFlows* je kolaborativne narave, saj lahko razvijalci v različnih programskih jezikih prispevajo svoje programe v obliki gradnikov, uporabniki pa lahko obstoječe gradnike inovativno povezujejo v nove delotoke.

Eno izmed področij z velikim številom uporabnikov statističnih in računalniških metod je gotovo jezikoslovje. Jezikoslovje je v zadnjih desetletjih doseglo izjemen napredek s pomočjo korpusnih metod raziskovanja avtentične jezikovne rabe, s katerimi lahko analiziramo kolokacije, besedotvorne posebnosti, specifično rabo ločil ipd. v eni ali več besedilnih zvrsteh. Ključno pomanjkljivost korpusnih analiz predstavlja dejstvo, da so v veliki meri odvisne od vključenosti jezikovnega gradiva v razpoložljiva orodja (Anthony 2013), kar jezikoslovcem pogosto onemogoča analizo najbolj relevantnega gradiva ali pa hitro odzivanje in preučevanje aktualnih družbenih procesov. Korpus Janes (Erjavec et al. 2018) tako ne omogoča vpogleda v besedila, povezana z družbenimi tematikami, ki so krojile diskurz na slovenskem spletu od leta 2017 dalje, kakršne so na primer migrantska problematika, spremembe družinske zakonodaje ali

reševanje morskih mejnih vprašanj. Po drugi strani orodja za gradnjo novih korpusov, kot so *SketchEngine* (Kilgarriff et al. 2004), *Wordsmith Tools* (Scott 1998) ali *AntConc* (Anthony 2014), zahtevajo sposobnost avtonomne priprave korpusa v zahtevanem formatu, vključno z izpeljavo postopkov lematizacije in jezikoslovnega označevanja, kar marsikateremu jezikoslovcu predstavlja nepremostljivo oviro. Delotoki lahko ustvarijo sinergijo med analitičnimi metodami jezikoslovja in računalništva ter skozi analizo različnih aktualnih tematik omogočijo boljše razumevanje družbenega dogajanja.

Za obdelavo slovenščine je že na voljo nekaj gradnikov in delotokov, ki omogočajo jezikovno označevanje besedil ter luščenje terminologije in definicij (Pollak et al. 2012a, 2012b). V poglavju predstavimo dvajset novih gradnikov, ki omogočajo sestavo številnih novih delotokov. Posebno pozornost namenimo gradnikom, ki omogočajo pridobivanje in obdelavo besedil z družbenih medijev in s tem korpusno podprto jezikovno analizo aktualnih družbenih tematik.

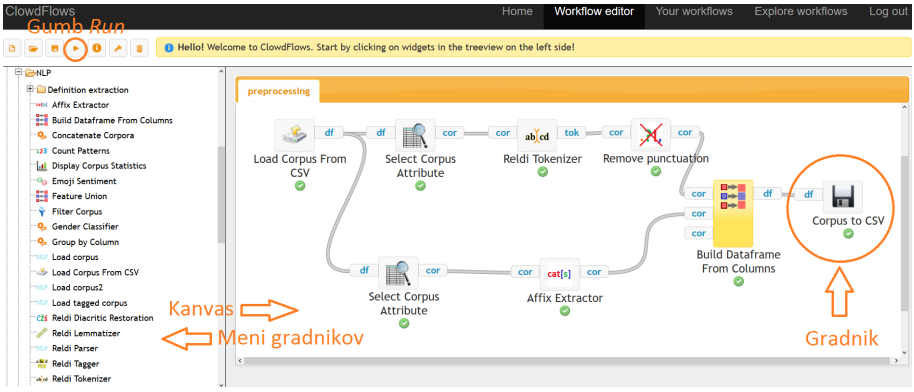
2 PLATFORMA CLOWDFLOWS

Kot je bilo na kratko omenjeno že v uvodu, je spletna platforma ClowdFlows namenjena gradnji, izvedbi in objavljanju interaktivnih delotokov za rudarjenje podatkov. Najpomembnejša značilnost platforme je grafični vmesnik, ki omogoča vizualno programiranje (Slika 1). Ta zajema izdelavo zapletenih delotokov s pomočjo tehnike *primi-odloži*. Uporabnik s klikom na gradnik v meniju na levi izbere želeni gradnik, ki se nato pojavi na delovni površini. Če kliknemo na izhod enega gradnika in nato na vhod drugega, se med gradnikoma vzpostavi povezava. Na ta način lahko hitro in učinkovito izdelujemo zapletene delotoke. Delotoke poženemo s klikom na ikono za pogon zgoraj levo ali s pritiskom na gumb *Run* v meniju, ki se pojavi ob desnem kliku na poljuben gradnik. V meniju je tudi gumb *Run only this*, ki namesto celega delotoka požene le izbrani gradnik. Če uporabnik želi pridobiti več informacij o posameznem gradniku, je v meniju tudi gumb *Help*, ki odpre okno s podrobno dokumentacijo o gradniku in opisom njegove uporabe.

Vizualno programiranje se je v zadnjem času uveljavilo kot učinkovit in priljubljen način za enostavno izvedbo zapletenih procesov, zato se je v preteklih letih pojavilo kar nekaj platform, ki so po funkcionalnosti podobne platformi ClowdFlows. Ena bolj znanih je platforma *RapidMiner* (Mierswa et al. 2006), znana tudi kot *YALE* (*Yet another Learning Environment*), ki jo avtorji opisujejo kot aplikacijo za strojno učenje in odkrivanje znanja v podatkovnih bazah. Dokaj znana je tudi platforma *KNIME* (*The Konstanz Information Miner*) (Berthold et al. 2009), ki jo lahko opišemo kot modularno okolje za izdelavo in interaktivno

izvajanje podatkovnih delotokov. Manj znane so platforme *ORANGE* (Demšar et al. 2004), *MyGrid*,[1] *Language Grid*[2] in *ARGO*.[3]

Platforma ClowdFlows se od zgoraj omenjenih platform razlikuje po posebnosti, da ne potrebuje namestitve na računalnik, saj jo lahko zaženemo v spletnem brskalniku. To poenostavi uporabo platforme in omogoča dostop do zgrajenih delotokov in gradnjo novih delotokov brez namestitve katerih koli programov. Še ena pomembna lastnost platforme *ClowdFlows* je njena odprtokodnost (njena izvorna koda je objavljena na platformi *GitHub*), kar pomeni, da lahko kdor koli v platformo dodaja nove gradnike. Dokumentacija o platformi in navodila za dodajanje novih gradnikov so dosegljivi na spletu.[4] Gradniki so vizualno predstavljeni deli programov, ki imajo definirane vhodne in izhodne oblike podatkov, parametre pa lahko uporabnik ročno izbere. Gradniki so lahko implementirani kot operacije spletnih servisov (angl. *web services*) ali pa lokalno v okolju *ClowdFlows*.



Slika 1: Prikaz grafičnega uporabniškega vmesnika platforme ClowdFlows. Na levi je meni gradnikov, na desni delovna površina za grajenje delotokov, levo zgoraj je gumb za zagon delotoka.

Platforma poleg gradnje novih delotokov omogoča tudi objavo delotokov, kar močno olajša ponovljivost rezultatov. Ponovljivost rezultatov je v znanosti temeljna zahteva, hkrati pa objava delotokov omogoča drugim uporabnikom, da že objavljeni delotok uporabijo na lastnih podatkih ali ga uporabijo kot osnutek za izdelavo lastnega delotoka s podobnimi značilnostmi. Na platformi *ClowdFlows* je trenutno objavljenih že več kot sto delotokov, ki so koristni tudi za učenje uporabe platforme.

1 http://www.mygrid.org.uk

2 http://langrid.org/

3 http://argo.nactem.ac.uk

4 http://clowdflows.readthedocs.io/en/latest/

ClowdFlows vsebuje množico gradnikov za podatkovno rudarjenje, pomanjkljiva pa je bila množica gradnikov za obdelavo naravnega jezika, ki bi jezikoslovcem omogočala kvantitativno analizo besedil in korpusov. V preteklosti je bila razvita posebna veja platforme *ClowdFlows*, poimenovana *TextFlows* (Perovšek et al. 2016), ki je specializirana predvsem za obdelavo naravnega jezika. Platforma *TextFlows* je trenutno namenjena predvsem obdelavi angleških besedil, prav tako pa ima predpisan izhodno/vhodni format gradnikov ADC (*Annotated Document Corpus*). Značilnost ADC-formata je izdelava novega nivoja anotacij za vsako operacijo obdelave naravnega jezika in shranjevanje teh nivojev skupaj z izvornim tekstom. Tak način zapisa omogoča, da so vse anotacije dosegljive na izhodu vsakega gradnika, kar izboljša povezljivost posameznih gradnikov. Po drugi strani je tak način zapisa spominsko potraten, tako da že relativno majhen korpus z okoli 100.000 pojavnicami in z nekaj nivoji anotacij, zapisan kot ADC-objekt, preseže mejo 1 GB, kar je zgornja meja za zapis objekta v podatkovno bazo PostGres[5], ki jo TextFlows uporablja za hranjenje podatkov. Zaradi omejitve procesiranja večjih korpusov in zaradi ciljnega jezika (slovenščine) smo se odločili, da nove gradnike za obdelavo naravnega jezika namesto v platformo *TextFlows*, ki bi bila tematsko bolj primerna, implementiramo v platformo *ClowdFlows*. Dodatni razlog za implementacijo gradnikov v platformo *ClowdFlows* je načrt avtorjev platforme *TextFlows*, da jo postopoma reintegrirajo v platformo *ClowdFlows* 3.0, novo in izboljšano različico.

V okviru raziskave smo tako v platformo *ClowdFlows* implementirali dvajset novih gradnikov za obdelavo naravnega jezika (ki jih predstavimo v tretjem razdelku) in kot primera uporabe implementirali dva funkcionalno različna delotoka za obdelavo naravnega jezika (predstavljena v četrtem razdelku).

3 GRADNIKI ZA KORPUSNO ANALIZO IN OBDELAVO SLOVENŠČINE

Glavni cilj novih gradnikov je obdelava slovenskega jezika, vendar je mnogo na novo implementiranih gradnikov splošno namenskih in omogoča podporo za več jezikov. Jezikoslovci po navadi za osnovno enoto obdelave uporabijo korpus, ki je sestavljen iz množice besedilnih dokumentov, zato smo ta princip upoštevali tudi pri naši implementaciji.

Obstaja množica različnih formatov za predstavitev korpusov, od zelo kompleksnih do precej preprostih. Pri načrtovanju implementacije smo se odločili, da mora biti izbrana predstavitev korpusa uporabniku prijazna in razumljiva, da

5 https://wiki.postgresql.org/wiki/FAQ#What_is_the_maximum_size_for_a_row.2C_a_table.2C_and_a_database.3F

mora omogočati preprosto razširitev in vključitev novih anotacij, hkrati pa mora upoštevati omejitve platforme, ki zaradi svoje spletne zasnove ne omogoča obdelave zelo velikih datotek. Zato smo se na koncu odločili za predstavitev korpusa v tabelarični obliki (angl. *dataframe*) (Slika 2), kjer vrstice predstavljajo posamezne dokumente, posamezni stolpci pa vsebujejo različne vrste anotacij (npr. besedilo, leme, oblikoskladenjske oznake ...).

Example	Sentiment	Subsentiment
Danes imajo naši turisti še zadnji dan turistovanja na davkoplačevalske stroške.	negative	cynicism
Jutri pa domov in davkoplačevalcem plačat račune.	negative	criticism
Potem bomo pa poslušali staro lajno...bila je lepa izkušnja, odziv publike je bil odličen, SLO smo dobro promovirali, dali smo vs	negative	cynicism
upam, da je šlo včeraj pri tinkari vse ok in da nas rtv spet ne zavaja kot nas je lani z naslovom...	positive	support
BO ... tokrat sem optimističen!	positive	positive exp
Tinkara je res profi in vsaka ji čast !!!	positive	support
Dejmo naši	positive	support
SREČNO !	positive	support
Držim pesti!	positive	support
Verjamem v finale in si finale tudi zaslužimo.	positive	positive exp
Pričakujem 12 točk iz Makedonije!	positive	positive exp

Slika 2: Primer korpusa v tabelarični obliki.

Tabelarična oblika je po svoji zasnovi podobna nekaterim že uveljavljenim formatom za predstavitev korpusa, je preprosta, razumljiva in računsko ter spominsko nezahtevna za obdelavo. Pri implementaciji takšne predstavitve smo si pomagali s programsko knjižnico *Pandas* (McKinney 2011), ki poleg drugih orodij vsebuje tudi razred *Dataframe*, ki predstavlja implementacijo tabelaričnega formata v programskem jeziku *Python*.

Na splošno lahko implementirane gradnike razdelimo v naslednje kategorije:

- gradniki za vnos korpusa,
- gradniki za obdelavo naravnega jezika,
- gradniki za upravljanje s korpusom,
- gradniki za izračun in prikaz statistik.

V naslednjih sekcijah, ki razvrščajo nove gradnike glede na funkcionalnost, podrobneje opišemo posamezne gradnike.

3.1 Gradniki za vnos korpusa

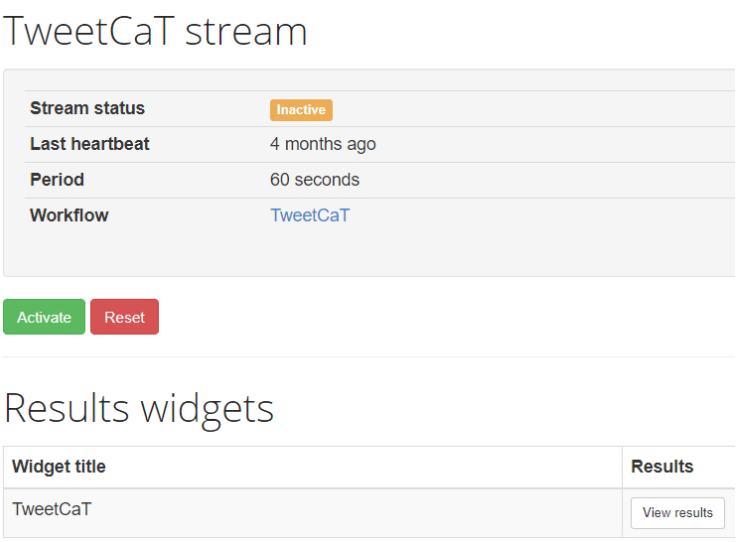
V platformo *ClowdFlows* smo implementirali dva gradnika za vnos korpusa:

- *Load Corpus From CSV:* Ta gradnik kot vhod sprejme datoteko s korpusom v formatu *comma-separated values* (CSV) in korpus pretvori v tabelarično obliko, ki je primerna za nadaljnjo obdelavo. Ta vrsta vnosa je primerna predvsem za jezikoslovce, ki večino svojih analiz izvajajo v programu Microsoft Excel, saj ta program omogoča transformacijo datotek iz formata Excel Workbook v format CSV. Gradnik uporabniku omogoča, da sam definira razločevalni znak (privzeto je ta v formatu CSV vejica), prva vrstica v dokumentu pa mora vsebovati imena stolpcev.
- *TweetCaT* (Ljubešić et al. 2014): Orodje za gradnjo korpusa z zajemom tvitov za jezike z relativno malo govorci, med katere sodi tudi slovenščina. Orodje omogoča zajem tvitov v realnem času (angl. *streaming*) s pomočjo vmesnika *Twitter API* in ima dva načina delovanja. V geografskem načinu (angl. *geographical search*) orodje zajema le tvite znotraj območja, ki je omejeno s štirimi geografskimi koordinatami. Uporabnik koordinate definira kot parametre gradnika v naslednjem zaporedju: minimalna geografska širina, minimalna geografska dolžina, maksimalna geografska širina, maksimalna geografska dolžina. V jezikovnem načinu (angl. *language search*) orodje zajema tvite na podlagi semenskih besed (angl. *seed words*) za posamezen jezik. Gre za besede, ki so specifične ali pogoste v določenem jeziku in jih lahko definira uporabnik ali pa uporabi privzeti seznam besed (za slovenščino so privzete besede definirane v besedilnem okencu parametra *seed words*, za hrvaščino, srbščino in bosanščino pa je množica privzetih besed podana v dokumentaciji gradnika). Z orodjem *TweetCaT* lahko uporabnik zgradi velike baze tvitov na hiter in učinkovit način, saj orodje nabere vse razpoložljive tvite posameznega avtorja, ki je objavil tvit z določeno besedo. Podrobnosti orodja *TweetCaT* so opisane v Ljubešić et al. (2018).

  Ker orodje za svoje delovanje potrebuje vmesnik *Twitter API*, lahko uporabnik pri uporabi gradnika izbira med uporabo že privzetega vmesnika *ClowdFlows Twitter API*, kar stori z izborom parametra *Use ClowdFlows authentication,* ali pa ustvari lastni *Twitter API*. Zaradi množične uporabe privzetega vmesnika *ClowdFlows Twitter API* je njegova uporaba nezanesljiva in močno upočasni zajem tvitov, zato je priporočeno, da uporabnik ustvari lastni *Twitter API* po naslednjem postopku: uporabnik najprej ustvari račun na *Twitterju*, nato pa na spletni strani https://dev.twitter.com/ sledi navodilom za izdelavo API-ja.[6] Po končanem postopku izdelave API-ja uporabnik pridobi uporabniški ključ (angl. *consumer key*), uporabniško geslo (angl. *consumer secret*), žeton za dostop (angl. *access token*) in geslo za žeton za dostop (angl. *access token secret*), ki jih vpiše v besedilna polja kot parametre gradnika.

6 Podrobna navodila za uporabo vmesnika Twitter API so objavljena tudi na spletnem naslovu http://docs.inboundnow.com/guide/create-twitter-application/.

Gradnik *TweetCaT* sodi v posebno kategorijo gradnikov za pretočne podatke (angl. *streaming widgets*), za katere je značilen sprotni zajem in jih v platformi *ClowdFlow*s upravljamo s pomočjo posebej za te gradnike narejene nadzorne plošče. Vsak delotok, ki vsebuje gradnike za zajem, je predstavljen kot tok (angl. *stream*). Če v glavnem meniju *ClowdFlows* kliknemo na gumb *Your workflows,* pridemo do seznama delotokov. Vsak delotok, ki vsebuje gradnike za zajem podatkov, je v seznamu predstavljen z ikono, ki prikazuje trenutno stanje toka delotoka. Tok je lahko aktiviran, kar pomeni, da zajem podatkov trenutno poteka, ali pa deaktiviran, kar pomeni, da se novi podatki trenutno ne nabirajo. Aktivacijo, deaktivacijo in ponastavljanje (izbris do sedaj zbranih podatkov) toka je možno nadzirati s pomočjo nadzorne plošče za vsak posamezen tok (Slika 3), ki je dosegljiva s klikom na ikono za nastavitve poleg ikone za prikaz stanja toka v seznamu delotokov. Nadzorna plošča omogoča tudi prikaz do sedaj zbranih podatkov.



Slika 3: Nadzorna plošča za upravljanje s tokom (angl. stream).

Tipičen scenarij uporabe gradnika *TweetCaT* bi bil naslednji: uporabnik izbere gradnik, ki se prikaže na delovni površini, določi želeni način delovanja (npr. jezikovni način) ter vpiše potrebne vhodne parametre. Tok je potrebno najprej aktivirati po navedenem postopku, saj bi sicer ob kliku na gumb *Run* gradnik javil napako, ki uporabnika opozori, da ni bil zajet še noben tvit. Po aktivaciji toka bo v nekaj minutah nabranih nekaj tisoč tvitov, že en sam tvit pa omogoča zagon gradnika. Pri vsakem

zagonu gradnika se kot izhod vrnejo vsi trenutno nabrani tviti. Ko imamo zajetih dovolj tvitov, lahko tok deaktiviramo, kar pomeni, da zaključimo z zajemom tvitov in na ta način poskrbimo, da se izhod gradnika ne spreminja (dopolnjuje z novimi tviti) več z vsakim novim zagonom.

Naslednji scenarij uporabe gradnika *TweetCaT* bi bila uporaba že obstoječega delotoka, ki vsebuje gradnik in že zajete podatke. Kot je že bilo omenjeno, platforma *ClowdFlows* omogoča objavljanje zgrajenih delotokov, ki tako postanejo dostopni ostalim uporabnikom. Če uporabnik želi uporabiti objavljen delotok, se pravzaprav ustvari kopija tega delotoka z vsemi podatki, ki jo nato uporabnik lahko poljubno spreminja, ne da bi s tem vplival na originalni delotok.

3.2 Gradniki za obdelavo naravnega jezika

Pri implementaciji orodij za obdelavo naravnega jezika smo se osredotočili na orodja za obdelavo slovenščine, hrvaščine in srbščine, implementirali pa smo tudi nekaj jezikovno neodvisnih orodij:

- *Reldi Tokenizer* (Ljubešić in Erjavec 2016): Orodje za tokenizacijo (prepoznavanje pojavnic, tj. besed in ločil) slovenskih, hrvaških in srbskih korpusov. Želeni jezik uporabnik nastavlja kot parameter gradnika, privzeto je nastavljen slovenski jezik. Orodje ima dva načina delovanja, in sicer omogoča tokenizacijo standardnih besedil (npr. znanstveni članki, literarna dela) in nestandardnih besedil (npr. blogi, tviti in komentarji). Kot vhod orodje sprejme stolpec korpusa v tabelarični obliki in ponavadi je to stolpec z besedilom. Orodje privzeto vrne seznam dvojic pojavnic in njihovih pozicij v tekstu, s čimer je izhodni zapis kompatibilen z vhodi gradnikov *Reldi Tagger*, *Reldi Lemmatizer* in *Reldi Diacritic Restoration*. Če uporabnik izbere parameter za sploščen izhod (angl. *Flatten output*), gradnik vrne sploščeno (enodimenzionalno) zaporedje s presledkom združenih pojavnic za vsak tekst v korpusu, ki izhod naredi kompatibilen z nekaterimi gradniki za nadaljnjo obdelavo in izračun statistik, kot je na primer gradnik *Display Corpus Statistics*. Primer tokeniziranega stavka, zapisanega v sploščenem formatu, je »Ta suhi škafec pušča , ker nima dna .«.
- *Reldi Tagger* (Ljubešić in Erjavec 2016): Oblikoskladenjski označevalnik za slovenščino, hrvaščino in srbščino. Posamezen jezik se izbere kot parameter, privzeto pa je nastavljen slovenski jezik. Kot vhod gradnik dobi dvojice pojavnic in njihovih pozicij v tekstu (izhod, ki ga vrne zgoraj opisani gradnik *Reldi Tokenizer*) in za vsako besedilo vrne zaporedje s

presledkom združenih oblikoskladenjskih oznak (angl. *MSD* oz. *morphosyntactic descriptor*) posameznih pojavnic v besedilu.[7]

- *Reldi Lemmatizer* (Ljubešić in Erjavec 2016): Orodje, ki posamezni besedi pripiše njeno osnovno obliko oz. lemo. Lematizator *Reldi* podpira slovenščino, hrvaščino in srbščino, posamezen jezik pa tako kot pri prej opisanem gradniku uporabnik izbere kot parameter. Tudi pri tem gradniku je privzeto nastavljen slovenski jezik. Ta gradnik sprejme isti vhod kot zgoraj opisani *Reldi Tagger* in za vsak tekst vrne s presledkom združene leme besed. Prav tako velja omeniti, da sta gradnika *Reldi Lemmatizer* in *Reldi Tagger* dve povezani operaciji enega in istega orodja, ki sta bili umetno ločeni in implementirani kot dva ločena gradnika za potrebe *ClowdFlows* platforme.
- *Reldi Diacritic Restoration* (Ljubešić et al. 2016): Orodje za obnovo diakritičnih znakov na izpuščenih mestih v nestandardnih zapisih (npr. sola > šola, pac > pač). Podprti jeziki so nestandardna slovenščina, hrvaščina in srbščina (privzeti jezik je slovenščina). Kot vhod ta gradnik dobi dvojice pojavnic in njihovih pozicij (izhod, ki ga vrne gradnik *Reldi Tokenizer*) in privzeto vrne seznam dvojic pojavnic z obnovljenimi diakritiki in njihovih pozicij za vsak vhodni tekst. Tako kot *Reldi Tokenizer* tudi ta gradnik omogoča sploščeni izhod s presledkom združenih pojavnic z obnovljenimi diakritičnimi znaki, ki izhod naredi kompatibilen z nekaterimi gradniki za nadaljnjo jezikovno obdelavo in izračun statistik.
- *Emoji Sentiment* (Novak et al. 2015): Gradnik za izračun sentimenta tvita s pomočjo emojijev kot vhod dobi stolpec korpusa v tabelarični obliki in za vsak dokument vrne sentiment (pozitiven, negativen ali nevtralen). Sentiment dokumenta je izračunan kot vsota sentimentov posameznih emojijev, izračun sentimenta posameznega emojija pa je odvisen od njegove pojavitve v pozitivnih, nevtralnih in negativnih dokumentih učnega korpusa.

Poleg dodajanja že obstoječih orodij za izdelavo korpusnih anotacij smo sami implementirali še naslednje gradnike:

- *Remove Punctuation*: Ta gradnik kot vhod dobi stolpec korpusa v tabelarični obliki (ponavadi stolpec s samim besedilom) in iz njega odstrani najpogostejša ločila:
  - #@!«$%&()*+,-./:;<=>?[\]^_`{|}~'.

  Kot izhod za vsako besedilo v korpusu vrne besedilo brez ločil. Odstranitev ločil se pogosto uporablja pri klasifikacijskih nalogah na nivoju teksta, kot

7 Množica oblikoskladenjskih oznak je bila definirana v okviru projekta Jezikoslovno označevanje slovenščine (http://nl.ijs.si/jos/msd/html-en/index.html).

je npr. klasifikacija spola avtorja teksta, saj ločila predvidoma ne vplivajo na točnost klasifikacijskega modela, hkrati pa njihova odstranitev zmanjša število značilk in s tem kompleksnost klasifikacijskega modela.

- *Remove Stopwords*: Gradnik kot vhod dobi stolpec korpusa v tabelarični obliki, ki vsebuje besedila in iz njih odstrani pomensko prazne besede (angl. *stopwords*). Za vsako besedilo korpusa gradnik vrne besedilo brez pomensko praznih besed. Trenutno gradnik vsebuje sezname slovenskih, angleških, španskih in portugalskih pomensko praznih besed,[8] kot privzeti jezik je nastavljena slovenščina.
- *Affix Extractor*: Gradnik kot vhod dobi stolpec korpusa v tabelarični obliki in privzeto vrne s presledkom ločene predpone besed dolžine tri (prve tri črke vsake besede). Uporabnik lahko kot parameter nastavlja dolžino vrnjenih zaporedij črk, hkrati pa lahko izbira med predponami, končnicami ter tako imenovanimi ločilniškimi končnicami (angl. *beg-puncts*) (Sapkota et al. 2015), kjer gre za zaporedja, ki se pričnejo z ločilom, ostali znaki v zaporedju pa so črke, številke ali presledki (npr., ločilniški končnici dolžine 3 vhodnega zaporedja »Pazi, drek! Ups« bi bili », d« in »! U«). Ločilniške končnice se uporabljajo kot značilke v klasifikacijskih modelih za profiliranje avtorjev teksta (npr. v modelih za napovedovanje spola, starosti, dialekta ...). V primeru ločilniških končnic, ki lahko vsebujejo tudi presledke, ne dobimo s presledkom ločenih predpon, temveč zaporedja, ločena z nizom ### (ta niz je bil izbran zaradi nizke verjetnosti pojavitve v originalnem tekstu). Posebnost gradnika *Affix Extractor* je, da ga lahko smiselno uporabimo tudi na stolpcih korpusa v tabelarični obliki, ki vsebujejo korpusne anotacije. Tako bi lahko na primer gradniku kot vhod dali s presledkom združene oblikoskladenjske oznake, ki jih vrne gradnik *Reldi Tagger,* in iz njih izluščili le predpono dolžine ena. Na ta način bi dobili le besedno vrsto pojavnice namesto celotne oznake, saj prva črka oznake določa besedno vrsto.
- *Count Patterns*: Gradnik za štetje vzorcev v korpusu kot vhod dobi stolpec korpusa v tabelarični obliki in za vsak dokument v korpusu vrne število vzorcev. Uporabnik lahko sam definira besede ali besedne zveze, ki naj jih gradnik prešteje (v obliki seznama z vejicami ločenih besed ali besednih zvez), lahko pa izbere, da naj gradnik prešteje vnaprej definirane vzorce, kot so ponavljajoči se znaki (angl. *character flood*), kot je na primer »jaaaa«, ali emojiji. Privzeta nastavitev je, da kot izhod gradnika dobimo seznam frekvenc vzorca za vsak posamezen dokument, če pa obkljukamo parameter *Count for entire corpus*, dobimo frekvenco pojavitev za celoten korpus. Gradnik privzeto vrne relativno frekvenco

8 Seznami praznih besed za angleščino, španščino in portugalščino so iz knjižnice NLTK, za slovenščino pa je seznam praznih besed izdelala Iza Škrjanec na podlagi korpusa Kres in zajema predloge, veznike, členke in zaimke.

pojavitev (število pojavitev vzorca se normalizira s številom vseh znakov v dokumentu), če pa obkljukamo parameter *Raw frequency*, gradnik vrne absolutno frekvenco oziroma število pojavitev vzorca.

- *Tweet Cleaner*: Ta gradnik kot vhod dobi stolpec korpusa v tabelarični obliki in za vsak dokument vrne dokument brez povezav (URL-jev), omemb (angl. *mentions*) in oznak (angl. *hashtags*). Uporabnik lahko izbira med odstranitvijo teh entitet in njihovo nadomestitvijo z žetoni *HTTPURL*, *TWEETMENTION* in *HASHTAG*. Gradnik je uporaben predvsem v primeru, če želi uporabnik procesirati in izluščiti informacije iz korpusa tvitov, saj so omembe in oznake zelo specifične za to vrsto besedil. Gradnik se lahko uporabi tudi pri obdelavi drugih besedil v vseh jezikih, če bi želeli na primer odstraniti povezave do spletnih strani iz poljubnega besedila.
- *Gender Classifier* (Martinc et al. 2017): Gradnik kot vhod dobi korpus v tabelarični obliki in vrne korpus v tabelarični obliki z dodatnim stolpcem, ki vsebuje avtomatsko pripisano oznako za spol avtorja posameznega teksta v korpusu. Uporabnik mora kot vhodni parameter podati ime stolpca v korpusu, ki vsebuje tekste. Klasifikator je bil naučen na tvitih in omogoča klasifikacijo slovenskih, angleških, španskih, portugalskih in arabskih tvitov (privzet jezik je slovenski). Za točno klasifikacijo potrebujemo besedilo, ki je daljše od enega tvita, idealno je to dokument, sestavljen iz dvesto ali več tvitov posameznega avtorja.

3.3 Gradniki za upravljanje s korpusom

Korpus besedil z vsemi pripadajočimi anotacijami je v platformi *ClowdFlows* zapisan v tabelarični obliki. Ta oblika zapisa omogoča različne vrste pretvorb, izborov in filtriranja korpusa. Za te vrste operacij so bili implementirani naslednji gradniki:

- *Select Corpus Attribute*: Gradnik kot vhod dobi korpus v tabelarični obliki in vrne stolpec korpusa z imenom, ki ga definira uporabnik. Stolpec je vrnjen v obliki pythonskega seznama, kjer posamezen element v seznamu predstavlja eno vrstico stolpca (tekst ali anotacije posameznega dokumenta v korpusu). Na ta način je mogoče iz korpusa v tabelarični obliki izločiti le informacije, ki jih potrebujemo za nadaljnjo obdelavo, kar zmanjša časovno in spominsko kompleksnost nadaljnjega procesiranja.
- *Concatenate Corpora*: Gradnik kot vhod dobi množico korpusov v tabelarični obliki in vrne vse vhodne korpuse, združene v en sam korpus v tabelarični obliki. Omejitev gradnika je, da morajo imeti vsi vhodni korpusi enako obliko (enako število in imena stolpcev).

- *Group by Column*: Gradnik kot vhod dobi korpus v tabelarični obliki in vrne korpus, ki ima vrednosti vseh atributov združene glede na vrednost atributa, ki ga definira uporabnik. Za primer lahko vzamemo korpus 30 tisoč tvitov, napisanih s strani 100 različnih avtorjev. Uporabnik določi stolpec z imeni avtorjev kot stolpec, ki naj vodi združevanje korpusa, gradnik pa vrne 100 dokumentov, pri čimer vsak dokument vsebuje s presledkom združene tvite posameznega avtorja.
- *Build Dataframe From Corpus*: Gradnik kot vhod dobi posamezne stolpce korpusa enake dolžine in vrne korpus v tabelarični obliki, sestavljen iz omenjenih stolpcev. Uporabnik lahko definira imena stolpcev izhodnega korpusa (v obliki seznama z vejico ločenih imen stolpcev v istem vrstnem redu, kot je vrstni red vhodnih stolpcev), če tega ne stori, so stolpci privzeto poimenovani kot *column_1*, *column_2*, *column_3* ...
- *Corpus to CSV*: Gradnik kot vhod dobi korpus v tabelarični obliki in kot izhod vrne datoteko CSV, v kateri je zapisan korpus v formatu CSV.
- *Filter Corpus*: Gradnik, ki omogoča filtriranje dokumentov v korpusu s pomočjo različnih poizvedb. Gradnik kot vhod dobi korpus v tabelarični obliki in vrne filtriran korpus v tabelarični obliki. Uporabnik lahko korpus filtrira s pomočjo petih različnih fraz za filtriranje:
  - *Je enako* – fraza ima obliko *ime_stolpca == poizvedba*, vrne pa se korpus v tabelarični obliki, ki vsebuje le dokumente, pri katerih vrednost posameznega stolpca ustreza iskani poizvedbi.
  - *Ni enako* – fraza ima obliko *ime_stolpca != poizvedba* in vrne le dokumente, pri katerih vrednost v stolpcu ni enaka iskani poizvedbi.
  - *Je večje* – fraza ima obliko *ime_stolpca > število* in vrne le dokumente, pri katerih je vrednost v stolpcu večja od števila. Ta poizvedba deluje le pri stolpcih z numeričnimi vrednostmi.
  - *Je manjše* – fraza ima obliko *ime_stolpca < število* in deluje na enak način kot *Je večje*, le da vrne obraten rezultat.
  - *Je v* – fraza ima obliko *poizvedba in ime_stolpca* in vrne le dokumente, ki vsebujejo iskano poizvedbo.

3.4 Gradniki za izračun in prikaz statistik

V tej kategoriji je bil implementiran le gradnik *Display Corpus Statistics*, ki kot vhod dobi stolpec korpusa. Ta gradnik omogoča izračun več različnih statistik:

- frekvenco posameznih pojavnic v korpusu (absolutne in relativne frekvence),

- izpis *hapax legomena*, ki so besede, ki se v korpusu pojavijo le enkrat, oz. *dis legomena*, ki se pojavijo točno dvakrat,
- izpis bigramskih in trigramskih kolokacij, ki temeljijo na meri povezanosti besed PMI (*pointwise mutual information*) (Church in Hanks 1990).

Pri izračunu statistik lahko uporabnik definira, za kakšne pojavnice želi izračun želene statistike. Izbira lahko med besednimi unigrami, bigrami, trigrami, tetragrami, pentagrami ali sekstagrami. Gradnik pri vsakem zagonu vrne tudi izračunano raznolikost besedišča (angl. type-token ratio), število dokumentov, število pojavnic in povprečno število pojavnic na dokument. Glede na vhodni gradnik lahko frekvence izpišemo tako za leme kot npr. za pojavnice ali oblikoskladenjske oznake.

4 PRIMERI DELOTOKOV

V tem razdelku bomo predstavili dva primera uporabe delotokov za obdelavo naravnega jezika, ki jih je mogoče implementirati v platformi *ClowdFlows* s pomočjo novih gradnikov. Oba primera sta javno objavljena in prikazujeta primer uporabe novih gradnikov na dveh različnih nalogah, s katerima se jezikoslovci pogosto srečujejo - izgradnja novega korpusa in analiza obstoječega korpusa. V obeh primerih gre za procesiranje računalniško posredovane komunikacije.

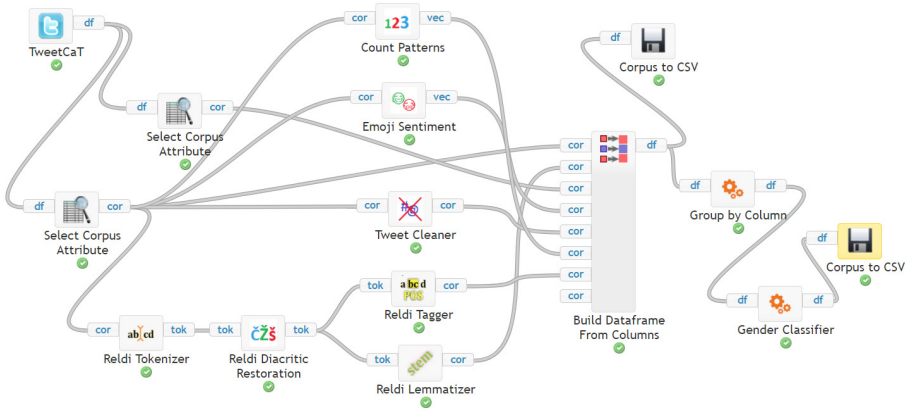
4.1 Nabor korpusa z orodjem TweetCaT

V tem razdelku je prikazano, kako lahko s pomočjo ClowdFlows platforme na preprost način zgradimo nov korpus slovenskih tvitov. Slika 4 prikazuje delotok, prav tako je delotok javno dostopen na spletu.[9] S pomočjo gradnika *TweetCaT* smo najprej nabrali okoli 23.000 slovenskih tvitov. Vsi parametri gradnika so bili privzeti, dodani so bili le podatki, potrebni za uporabo lastnega Twitter API-ja. Z gradnikom *TweetCaT* v naslednji fazi povežemo dva gradnika *Select Corpus Attribute*. Prvi iz korpusa izloči stolpec z imeni avtorjev tvitov, ki jih želimo imeti v izhodnem korpusu, drugi pa izloči stolpec z besedili. Besedila v nadaljevanju obdelamo na več načinov. Najprej s pomočjo gradnika *Count Patterns* izračunamo frekvenco emojijev za vsak posamezen tvit. S pomočjo gradnika *Emoji Sentiment* dobimo sentiment posameznega tvita, ki je izračunan na podlagi emojijev v tvitu. V izhodnem korpusu želimo imeti tudi prečiščene tvite brez oznak, omemb in oznak URL, zato uporabimo gradnik *Tweet Cleaner*.

9 http://clowdflows.org/workflow/10619/

Da bi dobili leme in oblikoskladenjske oznake za vse tvite, izpeljemo naslednje zaporedje operacij. Besedila najprej tokeniziramo s pomočjo gradnika *Reldi Tokenizer*, pri čemer pazimo, da izberemo parameter *Non-standard text*. Izhod gradnika nato povežemo z gradnikom *Reldi Diacritic Restoration*, ki normalizira besedila z obnovo manjkajočih diakritičnih oznak.

Ta gradnik nato povežemo z gradnikoma *Reldi lemmatizer* in *Reldi Tagger*, ki določita leme in oblikoskladenjske oznake. V končni fazi vse stolpce, ki jih želimo imeti v izhodnem korpusu (imena avtorjev, neprečiščena besedila, prečiščena besedila, frekvenco emojijev, sentiment na podlagi emojijev, leme in oblikoskladenjske oznake) združimo v nov korpus s pomočjo gradnika *Build Dataframe From Columns.* Korpus nato zapišemo v datoteko CSV s pomočjo gradnika *Corpus to CSV*.



Slika 4: Delotok TweetCaT.

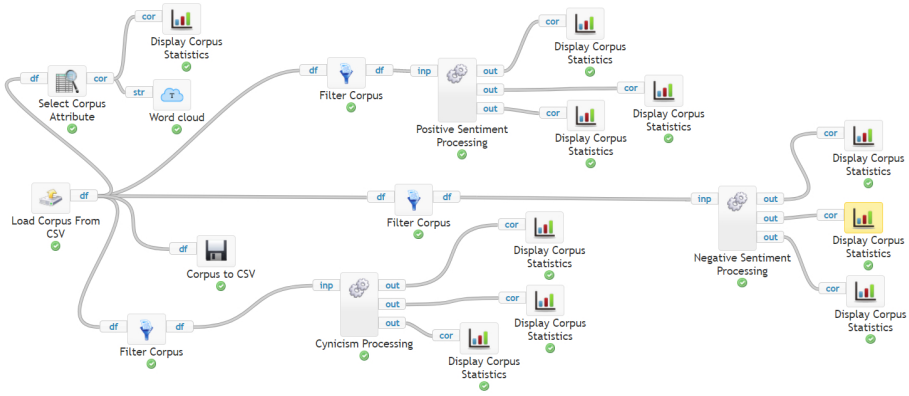
Ker želimo pridobiti tudi informacije o spolu avtorjev tvitov, dodamo še tri dodatne korake obdelave. Za začetek združimo vsa besedila posameznega avtorja v en dokument s pomočjo gradnika *Group by Column*, ki mu kot parameter damo ime stolpca, ki vsebuje imena avtorjev. Ta gradnik nato povežemo z gradnikom *Gender Classifier*, ki korpusu doda stolpec z oznakami spola. Na koncu tudi ta korpus zapišemo v datoteko s pomočjo gradnika *Corpus to CSV*.

Končni rezultat delotoka sta dva na novo izdelana korpusa v obliki CSV. Prvi vsebuje okoli 23.000 dokumentov in 7 stolpcev z različnimi informacijami o tvitih. Drugi vsebuje 10 dokumentov, en dokument na avtorja, sestavljen iz vseh dokumentov posameznega avtorja, in 8 stolpcev, saj je dodan stolpec z oznakami spola avtorjev.

Obstoječi delotok je mogoče uporabiti na dva načina. Če želi uporabnik delotok uporabiti na novih podatkih, bo moral najprej aktivirati tok po postopku, opisanem pri gradniku *TweetCaT*. To bo sprožilo nov zajem tvitov, uporabnik pa bo na ta način pridobil nov korpus tvitov z želenimi parametri, na katerih bodo preostali gradniki v delotoku izračunali enake statistike, kot so bile izračunane na originalnih podatkih. Če pa želi uporabnik preveriti dobljene rezultate, lahko ponovno zažene vse gradnike objavljenega delotoka. Seveda pa se lahko v obeh primerih delotok tudi poljubno prilagaja in spreminja glede na lastne potrebe.

4.2 Analiza ročno izbranega in označenega korpusa: primer Evrovizije

V tem razdelku prikažemo, kako lahko v platformi ClowdFlows obdelujemo obstoječe korpuse z različnimi anotacijami. Za primer vzamemo analizo korpusa Eurosong, ki je bila objavljena v prispevku Zwitter Vitez in Fišer (2016), tokrat pa ji dodajamo možnost avtomatskega označevanja, lematizacije in izračuna osnovnih statistik. Korpus Eurosong zajema komentarje na portalu rtvslo.si, ki so jih uporabniki objavili kot odziv na novico, da se je slovenska predstavnica na tekmovanju za Pesem Evrovizije uvrstila v finale. Komentarji izražajo ali odrekajo podporo slovenski predstavnici in omenjenemu tekmovanju nasploh, zato je bila naravnanost komentarjev najprej ročno označena s kategorijama pozitivnega in negativnega sentimenta, nato pa smo ročno dodali tudi oznake o podsentimentu (cinizem, kritika, optimizem, pesimizem). Cilj analize je identificirati jezikovne razlike med različno naravnanimi komentarji. Slika 5 prikazuje delotok, ki je dostopen na naslednji povezavi: http://clowdflows.org/workflow/10980/.

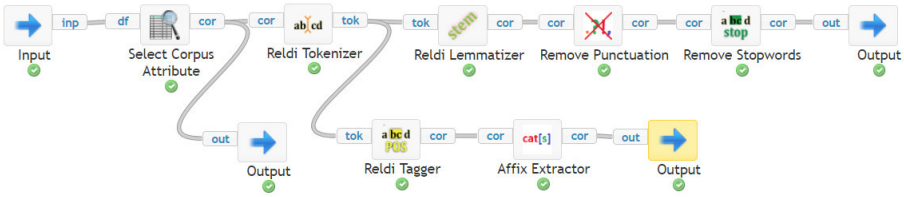


Slika 5: Delotok Eurosong.

Za izhodišče vzamemo korpus komentarjev Evrovizije v obliki CSV in ga naložimo s pomočjo gradnika *Load Corpus From CSV*. Korpus vsebuje ročne anotacije sentimenta in podsentimenta, ni pa še avtomatsko označen in lematiziran. Za začetek si želimo ogledati splošne statistike o besedilih v korpusu, zato s pomočjo gradnika *Select Corpus Attribute* izberemo le stolpec z besedili in ga povežemo z gradnikoma *Display Corpus Statistics* in *Word cloud,* saj nas zanima nekaj splošnih statistik o korpusu (število dokumentov, število pojavnic, povprečno število pojavnic na dokument), ki so prikazane v Tabeli 1, in pa najpogostejše polnopomenske pojavnice v korpusu (Slika 6). Vidimo, da se med zanimivimi polnopomenskimi besedami pojavljajo pojavnice *finale*, *Tinkara*, *uvrstitev, upam* ipd.

V naslednjem koraku želimo ugotoviti, kakšne so jezikovne razlike med pozitivnimi in negativnimi komentarji. Zato korpus razdelimo na dva podkorpusa glede na sentiment komentarjev. To storimo s pomočjo dveh gradnikov *Filter Corpus*, ki ju povežemo z gradnikom *Load Corpus From CSV*. S prvim gradnikom *Filter Corpus* želimo dobiti le podkorpus dokumentov s pozitivnim sentimentom, zato nastavimo parameter *query* na *Sentiment == positive*. Z drugim gradnikom *Filter Corpus* pa želimo dobiti podkorpus dokumentov z negativnim sentimentom, zato nastavimo parameter *query* na *Sentiment == negative*. Posebej nas zanimajo tudi cinični komentarji, zato tudi te izločimo iz korpusa s pomočjo gradnika *Filter Corpus,* ki ima parameter *query* nastavljen na *Subsentiment == cynicism.* Vse tri dobljene podkorpuse v nadaljevanju obdelujemo ločeno v treh skoraj identičnih podprocesih (*Positive Sentiment Processing, Negative Sentiment Processing, Cynicism Processing*), ki jih prikazuje Slika 7. Edina razlika med tremi procesi je, da podprocesa *Positive Sentiment Processing* in *Negative Sentiment Processing* ne vsebujeta gradnika *Remove Punctuation* (kar bo razloženo v nadaljevanju).

Slika 6: Najpogostejše polnopomenske pojavnice v korpusu Eurosong.



Slika 7: Podproces za obdelavo podkorpusov korpusa komentarjev Evrovizije.

V podprocesih iz korpusa izločimo le besedila s pomočjo gradnika *Select Corpus Attribute* ter ta gradnik povežemo z izhodom podprocesa (*Output*). Na ta izhod nato pri vseh podprocesih navežemo gradnik *Display Corpus Statistics*, da dobimo splošne statistike o vseh podkorpusih, ki jih prikazuje Tabela 1. Iz tabele je razvidno, da so dokumenti s pozitivnim sentimentom v povprečju krajši kot tisti z negativnim sentimentom, vendar hkrati opozarjamo pred posplošitvijo teh rezultatov, saj korpus zaradi svoje majhnosti statistično gledano ne dovoljuje zanesljivih zaključkov in je v tej raziskavi uporabljen predvsem za ponazoritev možne uporabe.

Tabela 1: Splošne statistike o vseh, pozitivnih, negativnih in ciničnih komentarjih v korpusu Eurosong.

Korpus	Število dokumentov	Število pojavnic	Povp. število pojavnic na dokument
Vsi dokumenti	70	894	12,77
Dokumenti s pozitivnim sentimentom	33	341	10,33
Dokumenti z negativnim sentimentom	37	553	14,95
Cinični dokumenti	20	284	14,20

Podobne ugotovitve so podane tudi v prispevku Zwitter Vitez in Fišer (2016), kjer je bilo z ročno analizo skladenjske strukture pozitivnih in negativnih komentarjev ugotovljeno, da imajo pozitivni komentarji pogosteje zgradbo enostavčne povedi (*Imam dober občutek*), medtem ko so negativni komentarji pogosteje formulirani v obliki večstavčne povedi (*Če zaupaš našim medijem, so še skoraj vsako leto bile kritike glede naših pesmi pozitivne, ampak rezultata pa nobenega in isto bo letos)*. Možno razlago za zaznano razliko na ravni dolžine in skladnje komentarjev vidimo v dejstvu, da pri izražanju negativnega mnenja uporabnik čuti dolžnost, da svojo kritiko utemelji, pri izražanju podpore pa se s tem ne ukvarja.

Nato se v podprocesih lotimo leksikalne analize besedil, ki jih tokeniziramo s pomočjo gradnika *Reldi Tokenizer* in lematiziramo z uporabo gradnika *Reldi Lemmatizer*, obenem pa odstranimo funkcijske besede z gradnikom *Remove Stopwords*. Seznam lem povežemo z izhodom (*Output*) podprocesa, ki ga v glavnem procesu povežemo z gradnikom *Display Corpus Statistics* (Slika 5).

N-gram	Raw frequency	Frequency	N-gram	Raw frequency	Frequency
,	43	0.1150	,	29	0.1189
.	26	0.0695	.	15	0.0615
...	9	0.0241	!	12	0.0492
finale	7	0.0187	tinkara	9	0.0369
iti	6	0.0160	finale	7	0.0287
spet	5	0.0134	dober	5	0.0205
res	5	0.0134	sreča	4	0.0164
pesem	4	0.0107	upati	4	0.0164
naprej	4	0.0107	:)	3	0.0123
nastop	4	0.0107	pesem	3	0.0123
zadnji	4	0.0107	pripravljen	3	0.0123
leto	3	0.0080	verjeti	3	0.0123
en	3	0.0080	srečno	3	0.0123
izpasti	3	0.0080	zaželeti	3	0.0123
upati	3	0.0080	iti	2	0.0082
....	3	0.0080	pozitiven	2	0.0082
danes	3	0.0080			

Slika 8: Najpogostejše leme v negativnih (levo) in pozitivnih (desno) komentarjih korpusa Eurosong.

Seznam najpogostejših lem v pozitivnih in negativnih komentarjih (Slika 8) pokaže zanimive razlike med obema podkorpusoma. Čeprav se v obeh podkorpusih pojavlja nekaj enakih lem, npr. *iti* in *finale*, opazimo zanimive razlike: v pozitivnih komentarjih se pojavi ime izvajalke *Tinkara*, v negativnih pa ne. V pozitivnih komentarjih prevladujejo leme, kot so *sreča, dober, verjeti, držati pest*, ki bi se lahko pojavili tudi pri izražanju podpore na kakšnem drugem področju (npr. pri športu). Med negativnimi komentarji pa so najpogostejše leme, bolj tipično vezane na tekmovanje za Pesem Evrovizije (*pesem, nastop, izpasti, kuhna*).

Pri analizi pozitivnih in negativnih komentarjev namenoma nismo uporabili gradnika *Remove punctuation*, ker smo želeli preveriti rabo ločil v pozitivnih in negativnih komentarjih. Pike in vejice so v obeh podkorpusih zelo pogoste, takoj za njima pa se v podkorpusu pozitivnih komentarjev pojavita klicaj in dvopičje z oklepajem v funkciji emotikona (*Dajmo klobasica!!!:*), medtem ko je pri negativnih precej bolj prisotno ločilo tri pike (*tokrat bomo res nesrečno izpadli...66.667% imamo možnosti da gremo naprej...hočem rečt, če danes ne zguramo naprej, pol nam res ni več pomoči*). Na podlagi zaznane razlike v rabi ločil

sklepamo, da v pozitivnih komentarjih uporabniki izražajo več čustev kot v negativnih, ko poskušajo svoje mnenje karseda racionalno utemeljiti.

Na koncu izvedemo v vseh podprocesih še analizo na besednovrstni ravni (Slika 9), ki v raziskavi Zwitter Vitez in Fišer (2016) ni bila mogoča. To storimo s pomočjo gradnika *Reldi Tagger,* ki vrne oznake MSD. Zanima nas le prvi znak v oznaki MSD, ki označuje besedno vrsto, zato ta gradnik povežemo z gradnikom *Affix Extractor* s parametroma *Affix type* in *Affix length,* nastavljenima na *prefix* in 1. Ta gradnik nato navežemo na izhod podprocesa, na katerega nato v glavnem procesu (Slika 5) navežemo gradnik *Display Corpus Statistics.*

N-gram	Raw frequency	Frequency	N-gram	Raw frequency	Frequency
N	125	0.1900	N	80	0.1975
V	122	0.1854	V	74	0.1827
R	78	0.1185	Z	60	0.1481
Z	77	0.1170	C	48	0.1185
C	64	0.0973	P	42	0.1037
P	51	0.0775	R	31	0.0765
Q	46	0.0699	A	23	0.0568
S	44	0.0669	S	22	0.0543
A	29	0.0441	Q	15	0.0370
Y	14	0.0213	M	8	0.0198
M	7	0.0106	Y	1	0.0025
X	1	0.0015	X	1	0.0025

Slika 9: Zastopanost besednih vrst (negativni komentarji levo, pozitivni desno).

Poleg samostalnikov in glagolov, ki so izrazito pogosti v obeh podkorpusih, pri pozitivnih komentarjih izstopajo ločila, kar smo pripisali večji čustveni nabitosti sporočil (*Tinkara je res profi in vsaka ji čast !!!*), pri negativnih komentarjih pa izstopajo prislovi (*važno, odvisno, nesrečno*), ki poudarijo naravnanost avtorja besedila do dogajanja v sporočilu.

Poleg osnovne naravnanosti avtorjev smo v korpusu ročno označili tudi podsentiment in posebej analizirali ciničnе pripombe. Te pripombe obdelujemo na enak način kot pozitivne in negativne komentarje, z izjemo že prej omenjenega dodatnega koraka izločitve ločil s pomočjo gradnika *Remove Punctuation*. Seznam najpogostejših lem (Slika 10) v ciničnih komentarjih omogoča nadaljnjo analizo.

Za razliko od specifik negativnih komentarjev, v katerih izstopata npr. lemi *izpasti* in *zadnji*, je v ciničnih komentarjih opaziti nekoliko drugačno leksikalno zastopanost. Na vrhu seznama se znajdejo leme, ki na prvi pogled delujejo nevtralno ali celo pozitivno, kvalitativna analiza pa pokaže, da imajo ciničen ali ironičen prizvok. Tako vlogo imajo na primer leme *finale* (*Finale ja... V soboto med publiko :)*), *dopust* (*važn da smo mišo peljali na dopust*), in *molkov*, ki je lema, pripisana pojavnici Molkova (*Sorry admin me bo brisal a resnica zgleda boli ali je to direktiva RTV-ja, da ne bi kdo*

rekel kaj čez Molkovo?). Ostale primerjave med ciničnimi in neciničnimi komentarji (npr. dolžina komentarjev, oznake MSD) bi bile smiselne pri večjem korpusu.

N-gram	Raw frequency	Frequency
finale	5	0.0360
pesem	3	0.0216
leto	3	0.0216
en	3	0.0216
dati	3	0.0216
dopust	2	0.0144
res	2	0.0144
publika	2	0.0144
kea	2	0.0144
zadnji	2	0.0144
molkov	2	0.0144
potem	2	0.0144
it	2	0.0144
lahko	2	0.0144
skuhati	2	0.0144
dobro	2	0.0144
kuhinja	2	0.0144
mrbit	2	0.0144

Slika 10: Najpogostejše leme v ciničnih komentarjih korpusa Eurosong.

Uporaba delotoka na korpusu Eurosong je pokazala, kako je mogoče preveriti hipoteze, ki so nastale s kvalitativnimi raziskavami (npr. glede leksikalnih značilnosti besedil), in analizirati besedila na novih ravneh (npr. na ravni oznak MSD), ki pred vključitvijo v orodja niso bile mogoče. Delotok prikazuje, kako lahko na hiter in enostaven način izvedemo osnovno statistično in primerjalno analizo besedil na leksikalni, besednovrstni in znakovni ravni tudi na poljubnih novih označenih korpusih.

5 SKLEP

V poglavju smo predstavili implementacijo orodij za obdelavo naravnega jezika v obstoječo platformo za vizualno programiranje *ClowdFlows*, ki omogoča lažjo in hitrejšo analizo besedil. Na kratko smo opisali platformo *ClowdFlows* in njen uporabniški vmesnik za vizualno programiranje ter podrobneje opisali na novo implementirana orodja, ki so bila v okviru projekta JANES razvita za gradnjo korpusa tvitov in obdelavo nestandardne slovenščine, pa tudi vrsto orodij za obdelavo kakršnih koli besedilnih korpusov. Kot primer uporabe orodij smo predstavili delotok za izdelavo in označevanje novega korpusa tvitov ter delotok za analizo komentarjev Evrovizije.

Nova orodja širijo nabor možnosti kvantitativnih analiz, ki jih je mogoče izvajati na obširnejših korpusih besedilih brez znanja programiranja in brez zapletenih pretvorb v formate, primerne za nadaljnje procesiranje v obstoječih orodjih za obdelavo besedil. Pri implementaciji orodij smo poseben poudarek namenili obdelavi slovenskih besedil, saj se ravno pri jezikih z manj govorci najbolj jasno kaže pomanjkanje orodij za analizo.

Primeri delotokov kažejo, da platforma *ClowdFlows* omogoča veliko svobode pri korpusnih analizah in dokaj enostavno implementacijo zapletenih znanstvenih postopkov. Kolaborativna narava in odprtokodnost platforme pa omogočata, da tudi drugi razvijalci dodajo nove gradnike in tako omogočijo nove možnosti analize.

Oba primera delotoka ponujata rešitev, kako omogočiti jezikoslovno analizo besedil, ki nastajajo kot odziv na aktualna družbena dogajanja. Obstoječi zaključeni korpusi, vključno s korpusom Janes, so namreč zelo aktualni z vidika preučevanja jezika računalniško posredovane komunikacije, vendar z vidika družbenih tematik, ki jih pokrivajo, hitro zastarajo. Korpus lahko pripravimo ročno in ga vnesemo, kot smo prikazali na primeru korpusa *Eurosong* v razdelku 4.2, ali pa korpus s poljubnega družbeno aktualnega področja sestavimo z orodjem *TweetCaT*.

Načrt nadaljnjega dela je večplasten. V prvi fazi želimo razširiti nabor orodij za obdelavo naravnega jezika v smislu funkcionalnosti in podpore drugih jezikov. Trenutno platforma *ClowdFlows* nima orodij za lematizacijo in oblikoskladenjsko označevanje angleščine, ki pa so podprta v veji *TextFlows*, kar nameravamo v prihodnje združiti v okviru platforme *ClowdFlows 3.0*. Manjkajo tudi orodja za izdelavo odvisnostne drevesnice (angl. *dependency trees*) in orodja za prepoznavanje imenskih entitet (angl. *named entity recognition*). Druga faza, ki je dolgoročnejša in že poteka, je optimizacija same platforme v smislu hitrejšega procesiranja in manjše porabe pomnilnika. Prav tako se bo izboljšala modularnost strukture platforme, kar bo omogočilo večjo prilagodljivost in lažjo implementacijo novih gradnikov.

Zahvala

Raziskava, opisana v prispevku, je bila delno opravljena v okviru projekta »ClowdFlows spletno tržišče za podatkovno in tekstovno analitiko« (RIA CF-Web, 2017–2018, H2020).

Literatura

Anthony, Laurence, 2013: A critical look at software tools in corpus linguistics. *Linguistic Research* 30/2.141–161.

Anthony, Laurence, 2014: AntConc (različica 3.4.3). Tokio: Waseda University.

Berthold, Michael R., Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel in Bernd Wiswedel, 2009: KNIME-the Konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter* 11/1. 26–31.

Church, Kenneth Ward in Patrick Hanks, 1990: Word association norms, mutual information, and lexicography. *Computational linguistics* 16/1. 22–29.

Demšar, Janez, Blaž Zupan, Gregor Leban in Tomaž Curk, 2004: Orange: From experimental machine learning to interactive data mining. *Knowledge discovery in databases: PKDD 2004*. 537–539.

Erjavec, Tomaž, Nikola Ljubešić in Darja Fišer, 2018: Korpus slovenskih spletnih uporabniških vsebin Janes. Fišer, Darja (ur.): *Viri, orodja in metode za analizo spletne slovenščine*. Ljubljana: Znanstvena založba Filozofske fakultete v Ljubljani. 16–43.

Kilgarriff, Adam, Pavel Rychly, Pavel Smrz in David Tugwell, 2004: Itri-04-08 the sketch engine. *Information Technology* 105. 116.

Kranjc, Janez, Vid Podpečan in Nada Lavrač, 2012: ClowdFlows: A Cloud Based Scientific Workflow Platform. *Proceedings of ECML/PKDD (2)*. 816–819.

Ljubešić, Nikola, Darja Fišer in Tomaž Erjavec, 2014: TweetCaT: a tool for building Twitter corpora of smaller languages. *Proceedings of LREC*. 2279–2283.

Ljubešić, Nikola, Tomaž Erjavec in Darja Fišer, 2016: Corpus-based diacritic restoration for south slavic languages. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016*). European Language Resources Association (ELRA). 3612–3616.

Ljubešić, Nikola in Tomaž Erjavec, 2016: Corpus vs. lexicon supervision in morphosyntactic tagging: the case of Slovene. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).* European Language Resources Association (ELRA). 1527–1531.

Ljubešić, Nikola, Tomaž Erjavec in Darja Fišer, 2018: Orodja za procesiranje nestandardne slovenščine. Fišer, Darja (ur.): *Viri, orodja in metode za analizo spletne slovenščine*. Ljubljana: Znanstvena založba Filozofske fakultete v Ljubljani. 74–99.

Martinc, Matej, Iza Škrjanec, Katja Zupan in Senja Pollak, 2017: PAN 2017: Author Profiling-Gender and Language Variety Prediction. *Working Notes Papers of the CLEF 2017 Evaluation Labs. CEUR Workshop Proceedings*.

McKinney, Wes, 2011: Pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*. 1–9.

Mierswa, Ingo, Michael Wurst, Ralf Klinkenberg, Martin Scholz in Timm Euler, 2006: Yale: Rapid prototyping for complex data mining tasks. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 935–940.

Kralj Novak, Petra, Jasmina Smailović, Borut Sluban in Igor Mozetič, 2015: Sentiment of emojis. *PloS one* 10/12.

Perovšek, Matic, Janez Kranjc, Tomaž Erjavec, Bojan Cestnik in Nada Lavrač, 2016: TextFlows: A visual programming platform for text mining and natural language processing. *Science of Computer Programming* 121. 128–152.

Pollak, Senja, Anže Vavpetič, Janez Kranjc, Nada Lavrač in Špela Vintar, 2012a: NLP workflow for online definition extraction from English and Slovene text corpora. *KONVENS*. 53–60.

Pollak, Senja, Nejc Trdin, Anže Vavpetič in Tomaž Erjavec, 2012b: NLP web services for Slovene and English: morphosyntactic tagging, lemmatisation and definition extraction. *Informatica* 36/4. 441–449.

Sapkota, Upendra, Steven Bethard, Manuel Montes-y-Gómez in Thamar Solorio, 2015: Not All Character N-grams Are Created Equal: A Study in Authorship Attribution. *HLT-NAACL*. 93–102.

Scott, Mike, 1998: WordSmith Tools Version 3. Oxford: Oxford University Press.

Zwitter Vitez, Ana in Darja Fišer, 2016: Linguistic Analysis of Emotions in Online News Comments-an Example of the Eurovision Song Contest. *Proceedings of the 4th Conference on CMC and Social Media Corpora for the Humanities*. Ljubljana, Slovenia, 27–28 September 2016. 74–76.