

BIOLAB AND COLLABORATORS

UVOD V RUDARJENJE BES

BIOLAB

Copyright © 2021 Biolab and Collaborators

PUBLISHED BY BIOLAB

TUFTE-LATEX.GOOGLECODE.COM

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

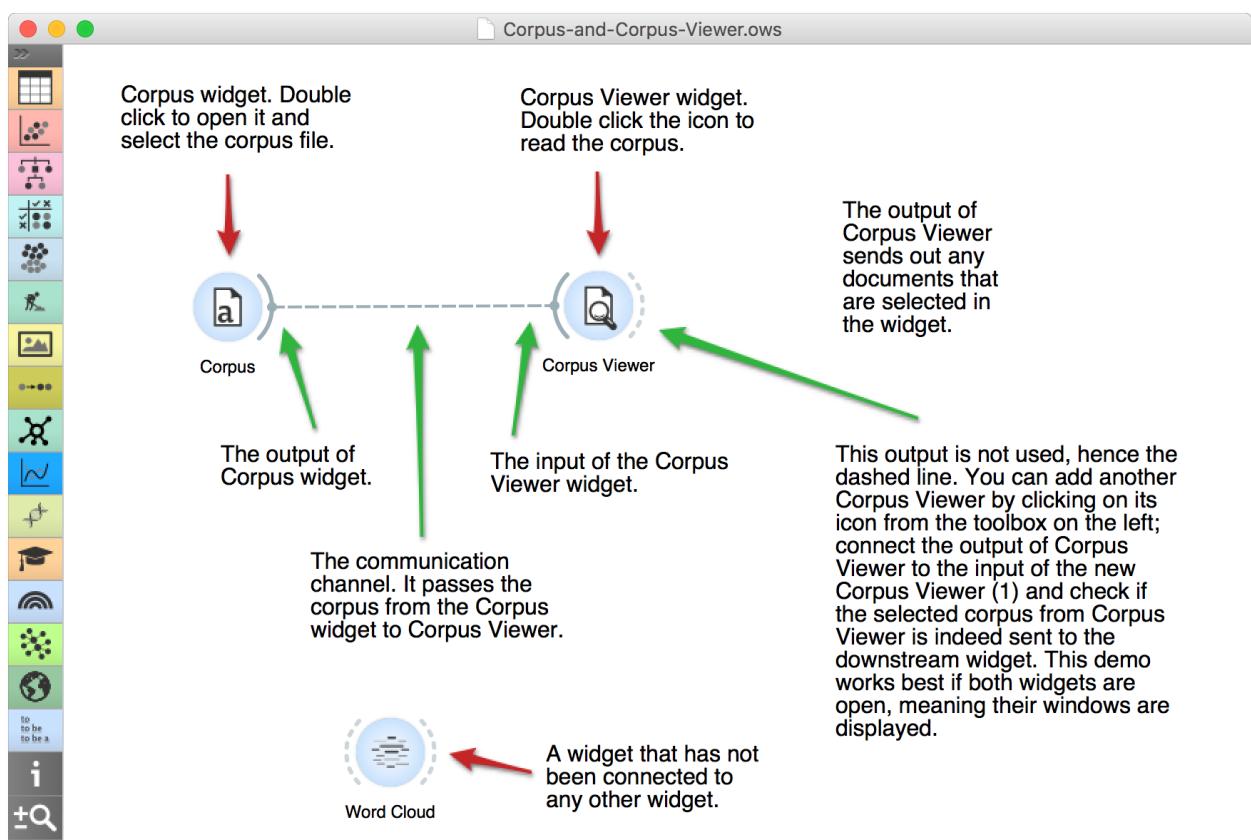
First printing, October 2021

Contents

<i>Delotoki v Orangeu</i>	5
<i>Priprava besedil</i>	8
<i>Kontekst</i>	11
<i>Vreča besed</i>	12
<i>Hierarhično razvrščanje v skupine</i>	13
<i>Hierarhično razvrščanje besedil</i>	15
<i>Klasifikacija</i>	18
<i>Logistična regresija</i>	19
<i>Ocenjevanje modelov</i>	20
<i>Obogatitev besed</i>	21
<i>Npovedovanje</i>	23
<i>Bibliography</i>	25
<i>Index</i>	26

Delotoki v Orangeu

DELOTOKI V ORANGEU so sestavljeni iz komponent, ki berejo, procesirajo in prikazujejo podatke. Te komponente imenujemo gradniki oz gradniki. Na desni je prazen prostor, t.i. platno. Nanj polagamo gradnike. Gradniki v Orangeu komunicirajo preko komunikacijskih kanalov. Izhod iz enega gradnika je uporabljen kot vhod za drug gradnik.

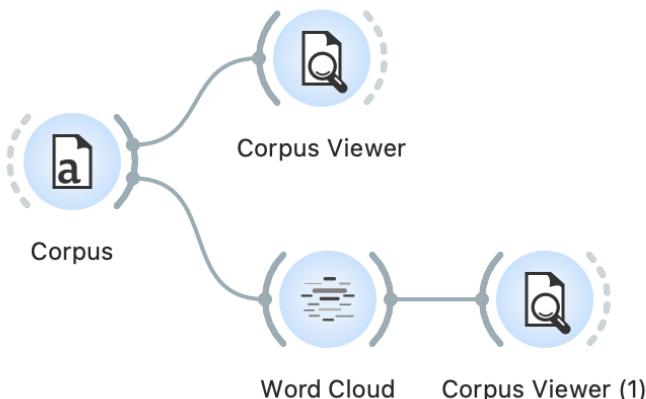


Delotoke sestavljamo tako, da polagamo gradnike na platno in jih povezujemo. Povezavo ustvarimo tako, da potegnemo črto od izhodnega v vhodni gradnik. Izhodi gradnika so na desni, vhodi pa na levi strani. V zgornjem delotoku gradnik *Corpus* pošilja podatke v gradnik *Corpus Viewer*.

Slika zgoraj kaže preprost delotok z dvema povezanimi gradnikoma in enim gradnikom brez povezav. Izhodi gradnika so na desni strani, vhodi pa na levi.

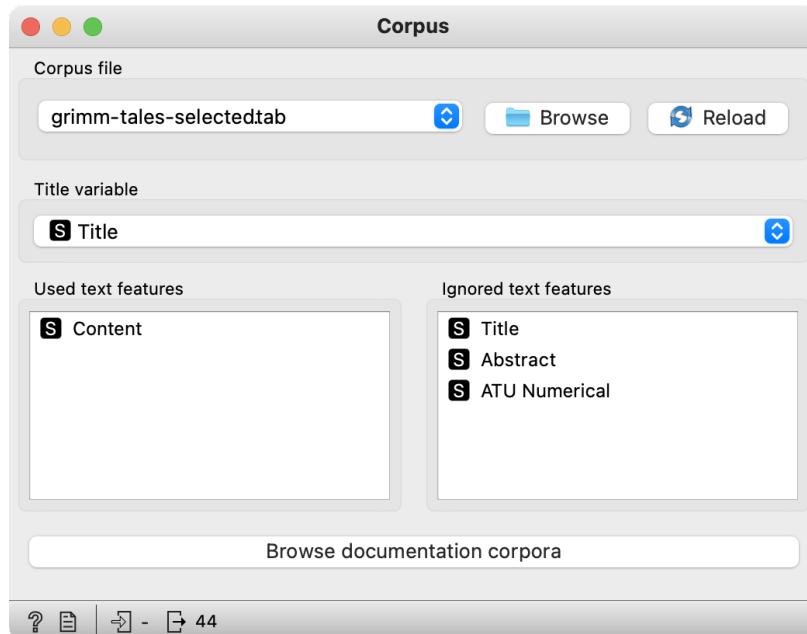
Pričnite z gradnjo delotoka, ki vsebuje gradnik *Corpus*, dva gradnika *Corpus Viewer* in gradnika *Word Cloud*:

Delotok z gradnikom *Corpus* bere podatke iz računalnika in jih pošlje v gradnika *Corpus Viewer* in *Word Cloud*. *Corpus Viewer* prikaže besedila v iskalniku, *Word Cloud* pa izriše najpogosteje besede. Dokumenti, ki vsebujejo izbrano besedo iz gradnika *Word Cloud*, so prikazani v gradniku *Corpus Viewer (1)*.



Gradnik *Corpus* bere podatke iz lokalnega diska. Odprite *Corpus* tako, da dvakrat kliknete na ikono. Dodatek Text že vsebuje nekaj prednaloženih korpusov. Iz teh ("Browse") izberite *Grimm-tales-selected.tab*, korpus z izbranimi Grimmovimi pravljicami.

Oranjevi delotoki se pogosto pričnejo z gradnikoma File ali *Corpus*. Korpus Grimmovih pravljic vsebuje 44 dokumentov. Polje "Used text features" na levi pove, katere stolpce bomo smatrali kot del besedila, medtem ko polje na desni vsebuje dodatne informacije (naslov, povzetek, itd.).



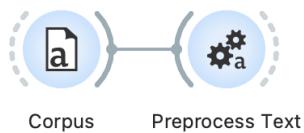
Odprite *Word Cloud*. *Word Cloud* (oblak besed) prikaže pogostost besed v dokumentih, kjer so pogostejše besede prikazane sorazmerno večje. Izberite besedo v oblaku in jo pošiljte v *Corpus Viewer (1)*. Sedaj lahko pregledate samo dokumente, ki vsebujejo izbrano besedo.



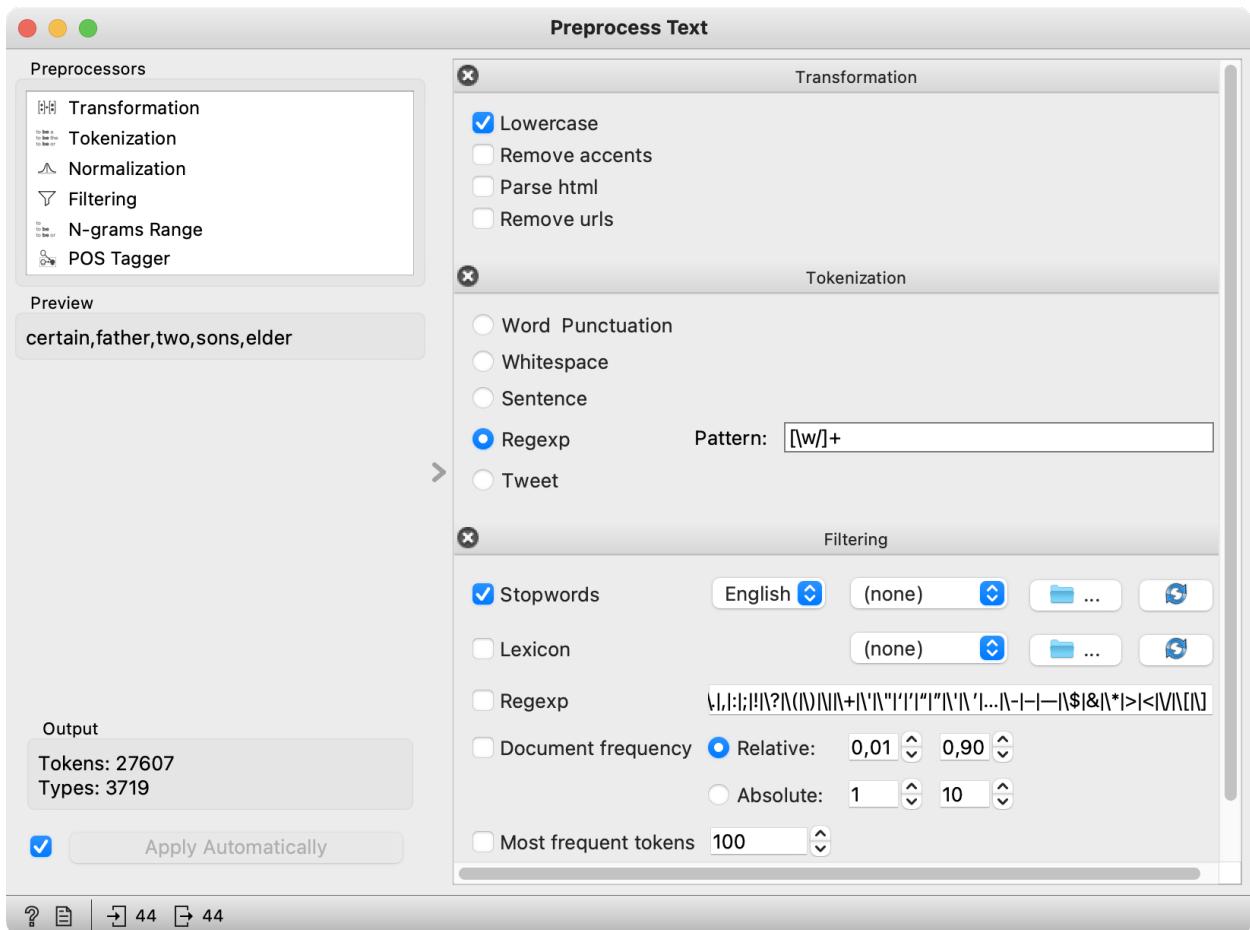
Počakajmo malo! Ta oblak besed je živa groza! Vidimo lahko celo kopico semantičnih smeti. Ali obstaja kakšen način, da to nekako uredimo?

Seveda! Odstraniti moramo vse delčke, ki ne vsebujejo nikakršne informacije, točneje ločila in odvečne besede (členke, pomožne glagole, veznike).

Príprava besedil



Word Cloud je preprosto prikazal vse besede in simbole, ki obstajajo v besedilu. Ampak običajno to ni to, kar hočemo. Ponavadi želimo prikazati zgolj pomenske enote, torej semantično bogate besede. Zato potrebujemo predprocesiranje.



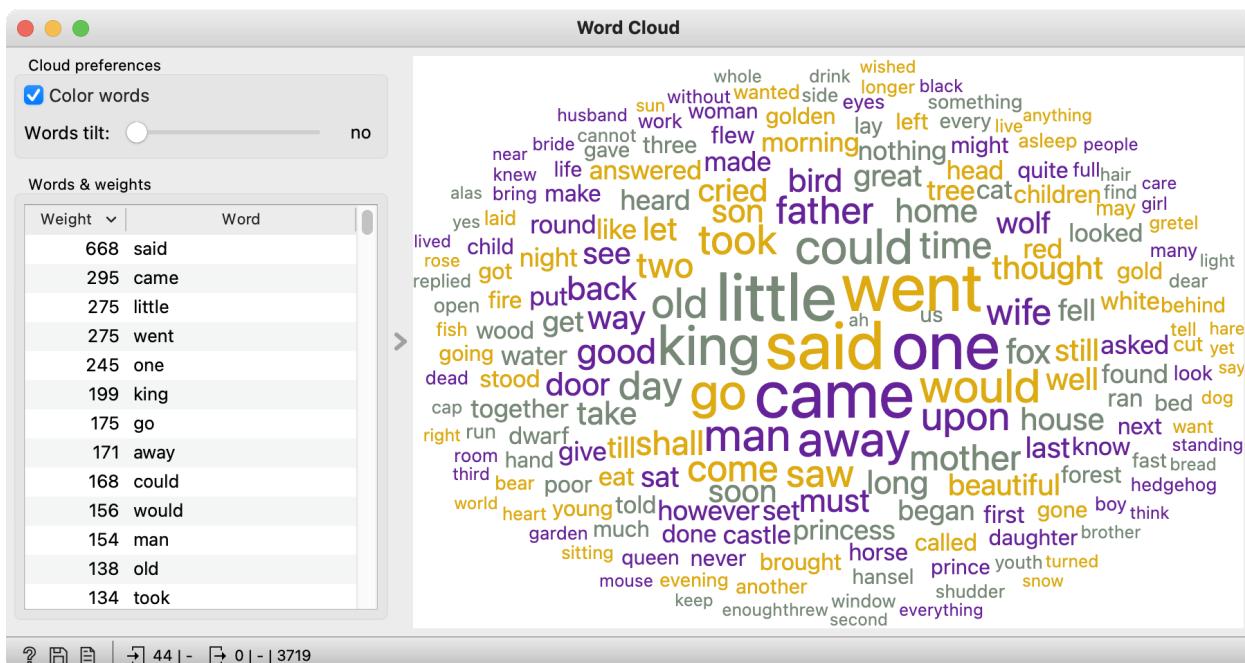
Preprocesiranje definira kaj je pomembno v podatkih. Je "Zdravnik" enako kot "zdravnik"? Naj upoštevamo besede kot so "in", "ali", "ko" ali naj jih izpustimo? Ali želimo upoštevati besedi "živel" in "živi" kot isto besedo? Preprocesiranje definira osnovne enote analize.

Token je osnovna enota naše analize. Lahko je beseda, besedna zveza, stavek... S predprocesiranjem definiramo osnovne enote za analizo.

V gradniku *Preprocess Text* smo vse besede pretvorili v male črke, vsako besedo smo obravnavali kot svojo *menoto (token)* in odstranili ločila, na koncu pa smo odstranili tudi nepomenske besede (npr. 'in', 'da', 'čeprav'). Takšno predprocesiranje ustvari sledeče enote:

"To je vzorčni stavek." → "vzorčni", "stavek"

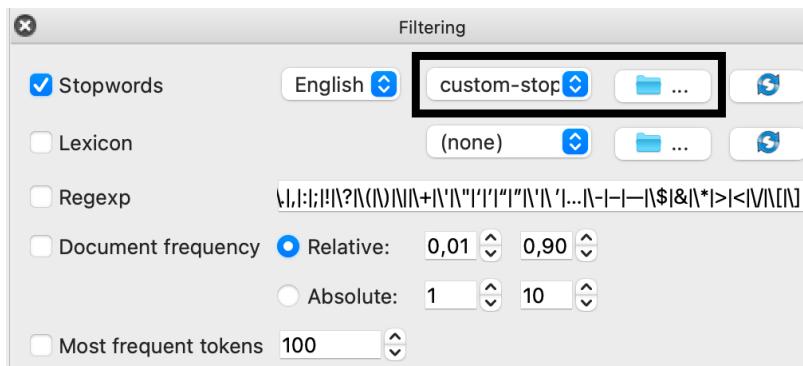
Rezultate predprocesiranja si lahko pogledamo v gradniku *Word Cloud*, kjer vidimo najpogosteje enote. S to vizualizacijo lahko identificiramo odvečne besede in nepravilnosti.



Ker ne želimo upoštevati besed brez pomena, smo že odstranili nekaj nepomenskih besed. Ampak morda generično filtriranje ni dovolj za našo analizo.

V tem primeru vedno lahko naložimo seznam besed po meri. Odprite urejevalnik besedil in ustvarite seznam nepomenskih besed oz. besed, ki jih želite odstraniti. Vsako besedo zapišite v svojo vrstico in shranite dokument v obliki *.txt*.

Rezultate predprocesiranja vidimo v gradniku Word Cloud. Dve najpogosteši besedi sta "would" in "could". Če se odločimo, da ti dve besedi nista primerni za našo analizo, ju moramo odstraniti. To lahko storimo s filtriranjem po meri.



Dober urejevalnik besedil je Sublime, lahko pa uporabite tudi WordPad ali Word.

Seznam besed naložite s klikom na ikono z mapo poleg opcije *Stopwords* v razdelku *Filtering*.

Filtriramo lahko tudi besede, ki so preredke ali prepogoste. Redke besede se pojavijo običajno le v nekaj dokumentih, medtem ko so prepogoste besede presplošne ali pa nimajo pomena (stopwords). Da bi ohranili le besede, ki zares predstavljajo naš korpus dokumentov, uporabimo filtriranje Document frequency (Pogostost v besedilu). Če nastavimo vrednosti na 0,1 and 0,9, bomo obdržali le tiste besede, ki se pojavijo v več kot 10 % in manj kot 90 % dokumentov.

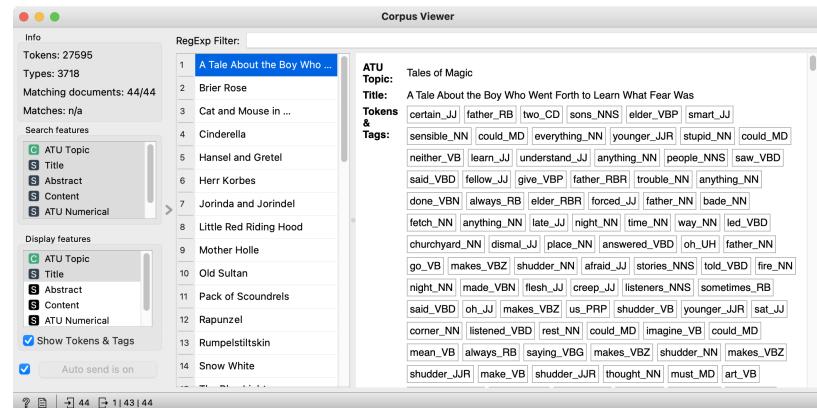
Predprocesiranje je ključ do uspešne analize besedil. Omenili smo le nekaj tehnik, sami pa lahko preizkusite še druge, na primer:

- *normalizacija (Normalization)* pretvori vse besede v korene oz. osnovne oblike (na primer sinovi v sin)
- *n-grami* so večje enote, na primer bigrami (par zaporednih besed) in trigrami (trojke besed)
- *oblikoskladenjsko označevanje (POS tagging)* označi vsako enoto s njenim oblikoskladenjskim vlogo (sinovi → samostalnik, množina, oznaka = NNS)

Pred kratkim smo za slovenščino dodali korenjenje z orodjem UDPipe

Za razlago POS oznak glejte:
<http://nl.ijs.si/imp/msd/html-sl/>

Na sliki vidite gradnik Corpus Viewer, s katerim si lahko pogledamo naslove, besedila dokuemntov in enote na katere je preprocesirane razbilo besedilo. V našem primeru imamo poleg enot prikazane tudi POS oznake.



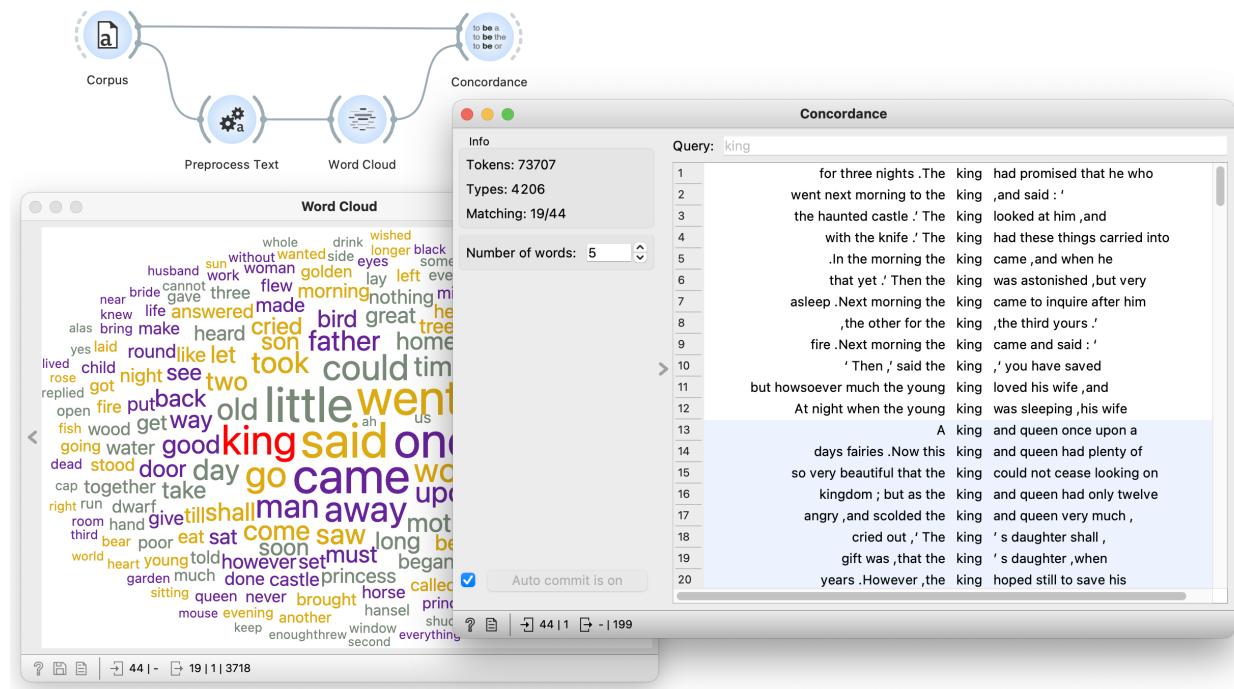
Kontekst

Sedaj smo pripravili korpus in čas je, da ga prikažemo. En način prikaza je oblak besed, ki ga že poznamo. Word Cloud nam prikaže pogostost besed. Pogostejsa kot je beseda, z večjimi črkami bo zapisana.

Še vedno pa ne vemo, kako se besede uporabljajo v besedilu. Na primer 'oh' je lahko maločrkovna verzija besede OH (kemijska spojina hidroksid), preprost vzklik 'Oh!' ali pa kratica za ameriško zvezno državo Ohio.

Da bi preverili kontekst posamezne besede, lahko uporabimo gradnik Concordance. Concordance na pokaže besedilo okrog izbrane besede. Pozvežite Concordance z gradnikom Corpus. Tako Concordance dobi vhodno besedilo. Besedo lahko poiščemo z iskalnikom na vrhu gradnika ali pa jo izberemo v gradniku Word Cloud.

Vizualizacije v Orangeu so narejene tako, da podpirajo izbor podmnožic. Odkrivanje zanimivih podmnožic in raziskovanje njihovih podobnosti je ključni del odkrivanja znanj iz podatkov.



V tem primeru smo izbrali besedo 'king' v oblaku besed in preverili njen kontekst v gradniku Concordance.

Dokumente, ki vsebujejo izbrano besedo, si pogledamo tako, da izberemo dokumente v gradniku Concordance in jih pošljemo v Corpus Viewer za podrobnejšo analizo.

Vreča besed

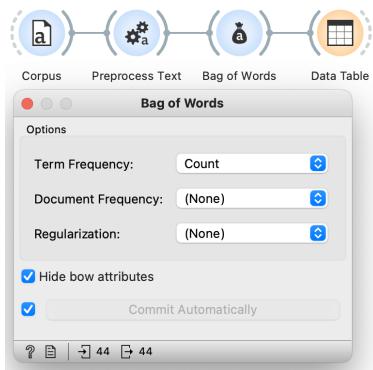
Sedaj imamo predprocesirano besedilo, s pravimi enotami, ampak še vedno pa ne moremo odkriti nikakršnih vzorcev v besedilu. Za to potrebujemo številke in preprost način, kako spremenimo besedila v številske vektorje je... da preštejemo besede v vsakem dokumentu!

	this	is	an	example	another	apple
"This is an example"	1	1	1	1	0	0
"Another example"	0	0	0	1	1	0
"This is another apple"	1	1	0	0	1	1

Gradnik Bag of Words ustvari tabelo z besedami v stolpcih in dokumenti v vrsticah. Vrednosti so pojavitve besed v vsakem dokumentu.

Besede lahko preprosto preštejemo (TF ali term frequency) ali pa besede utežimo glede na to, kako pogosto se pojavijo v dokumentih (IDF ali inverse document frequency). S TF-IDF bodo pogoste besede imele nizko vrednost, saj se pojavijo velikokrat pri velikem deležu dokumentov, medtem ko bodo imele pomembne besede visoko vrednost, saj se pojavijo pogosto v majhnem deležu dokumentov.

Podatke pošilje v gradnik Bag of Words in od tam naprej v Data Table. Vidimo nov stolpec, ki vsebuje pojavitve besed za vsak dokument. Sedaj imamo številke in končno lahko pričnemo z analizo!

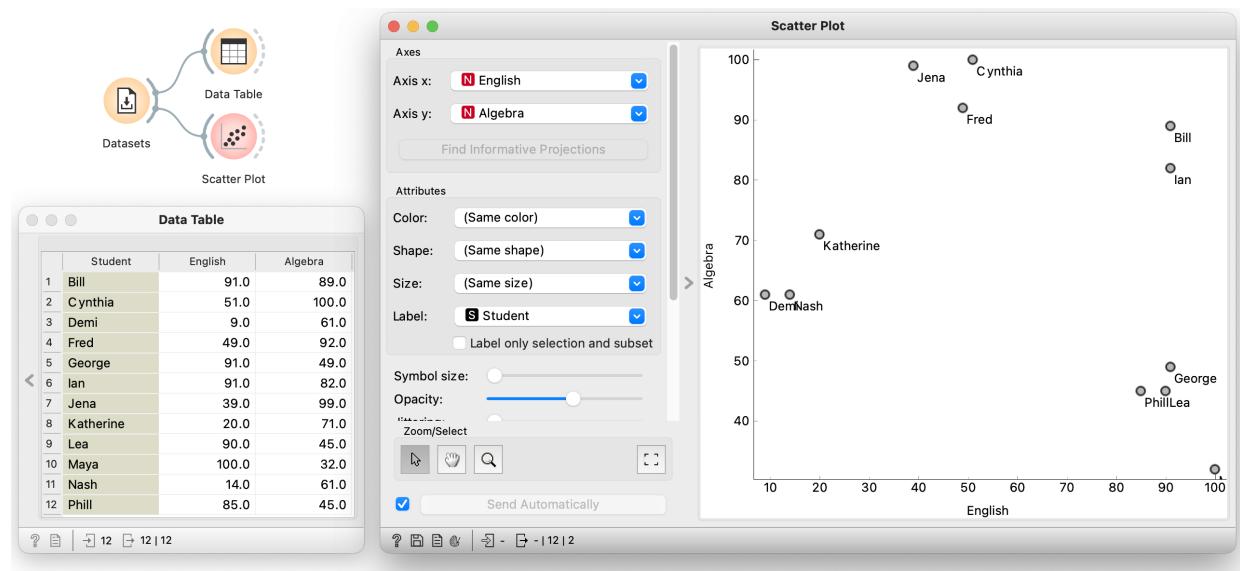


Data Table		
bow-feature hidden include skip-normalization title	ATU Topic	Title
1	Tales of Magic	A Tale About...
2	Tales of Magic	Brier Rose
3	Animal Tales	Cat and Mou...
4	Tales of Magic	Cinderella
5	Tales of Magic	Hansel and ...
6	Animal Tales	Herr Korbes
7	Tales of Magic	Jorinda and ...
8	Tales of Magic	Little Red Ri...
9	Tales of Magic	Mother Holle
10	Animal Tales	Old Sultan
11	Animal Tales	Pack of Sco...
12	Tales of Magic	Rapunzel
13	Tales of Magic	Rumpelstilts...
14	Tales of Magic	Snow White
15	Tales of Magic	The Blue Light
16	Animal Tales	The Bremen ...
17	Animal Tales	The Crumbs ...
18	Animal Tales	The Dog and...
19	Tales of Magic	The Elves an...
20	Tales of Magic	The Fisher...

Hierarhično razvrščanje v skupine

Ena od nalog rudarjenja besedil je iskanje zanimivih skupin dokumentov. Torej radi bi odkrili dokumente, ki so si podobni mes sabo.

Poglejmo si preproste podatke z dvema stolpcema (glejte opombo) in jih prikažimo v gradniku Scatter Plot. Koliko skupin imamo? Kaj predstavlja različne skupine? Kateri primeri sodijo v posamezno skupino?



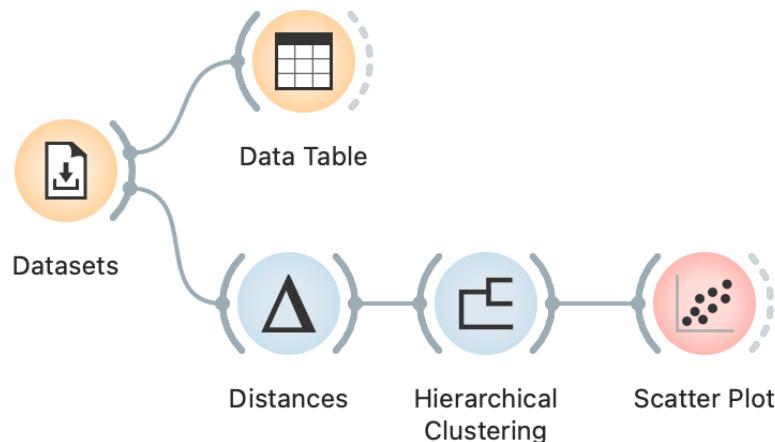
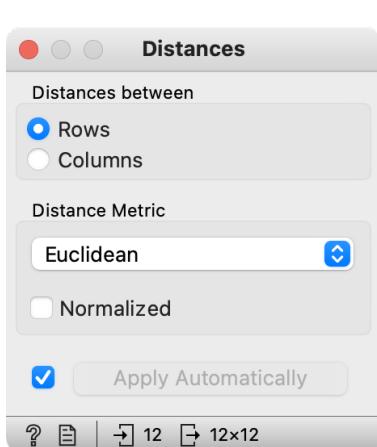
Kaj sploh pomeni "podobno"? Študenti so opisani s številskimi spremenljivkami, torej s ocenami pri predmetu. Ena od mer podobnosti je kosinusna razdalja. Pogostost besed iz vreče besed je predstavljena z vektorji, ki kažejo vsak v svojo smer glede na vsebino posameznega dokumenta. Kosinusna razdalja je kot med temi vektorji.

Sedaj definirajmo še postopek za razvrščanje v skupine. Recimo, da začnemo z vsakim dokumentom v svoji skupini, nato pa v vsakem koraku združimo skupini, ki sta si najbolj podobni. Razdaljo med skupinami izračunamo kot povprečje razdalj med posameznimi elementi skupine. Tak postopek imenujemo hierarhično razvrščanje v skupine.

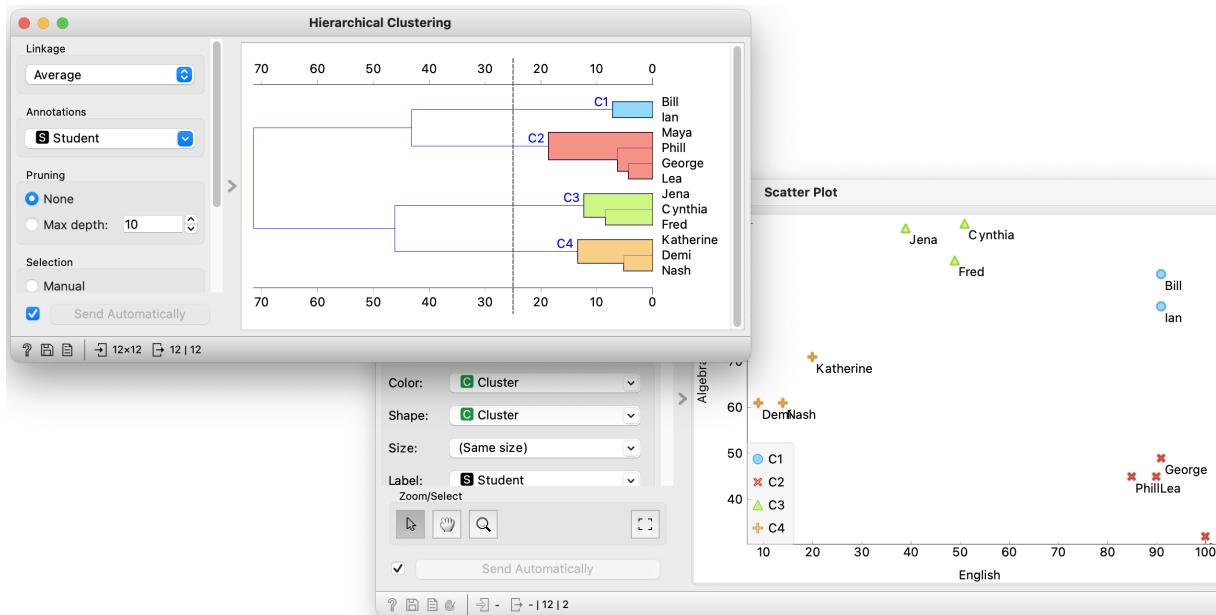
Razvrščanje v skupine bomo predstavili s preprostimi podatki o študentih in njihovih ocenah pri angleščini in matematiki. Podatki so dostopni v Datasets widgetu.

Načinov merjenja razdalj med skupinami je več. Način, ki smo ga opisali, se imenuje povprečna razdalja (average linkage). Lahko bi računali tudi razdaljo med najbližjima točkama v skupini (single linkage) ali pa med točkama, ki sta si najbolj oddaljeni (complete linkage).

Rezultate razvrščanja v skupine na primeru naših študentov si lahko pogledamo v sledečem delotoku:



Naložite podatke z gradnikom File, izračunajte razdalje z gradnikom Distances, uporabite Hierarchical Clustering in si poglejte rezultate v gradniku Scatter Plot. Gradnik Hierarchical Clustering omogoča, da hierarhijo skupin odrežemo pri določeni meri podobnosti in tako definiramo skupine.



Hierarhično razvrščanje besedil

Vrnimo se h Grimmovim pravljicam in ustvarimo naslednji delotok:

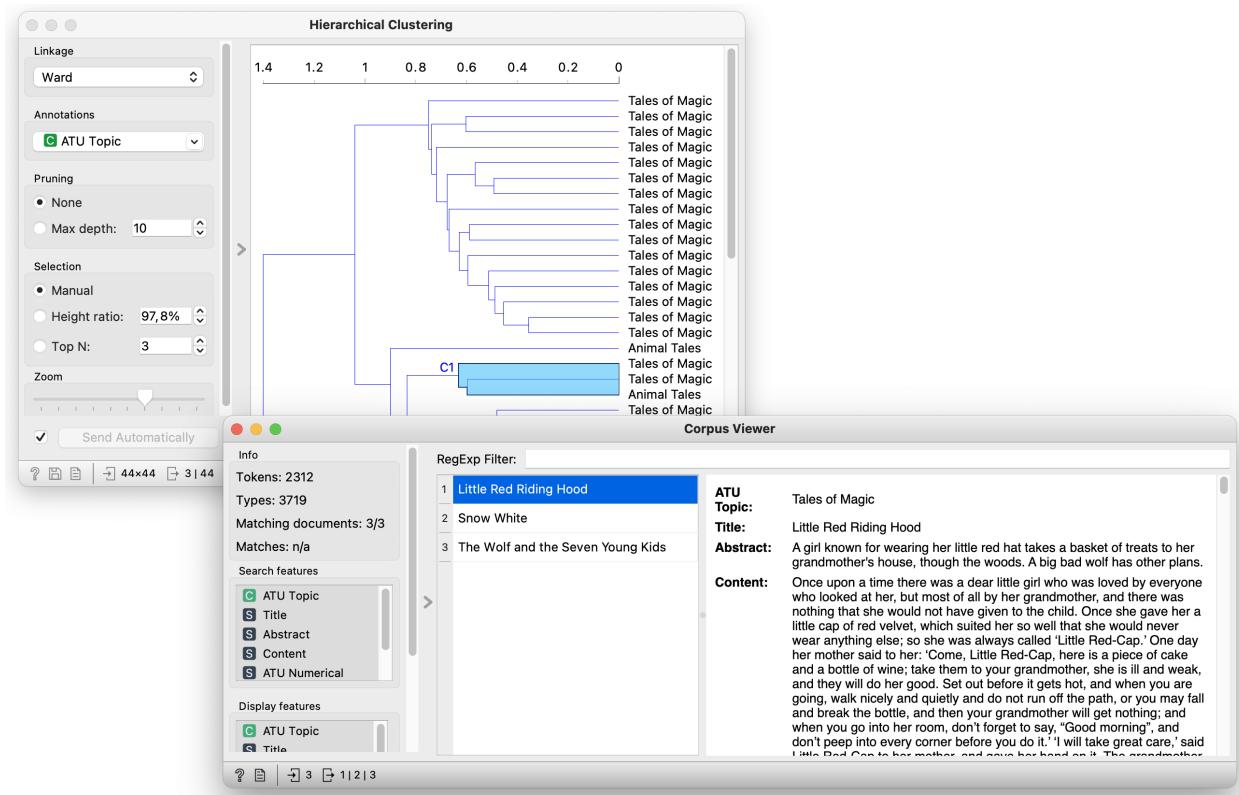


Gradnik Hierarchical Clustering prikaže gruče v obliki dendrograma. Hierarchical Clustering povežite z gradnikom Corpus Viewer in odprite oba gradnika. Izberite gručo v dendrogramu in v gradniku Corpus Viewer poglejte, kateri dokumenti pripadajo izbrani skupini.

Raziščite različne gruče. Zakaj so nekatere magične pravljice (Tales of Magic) pomešane z živalskimi pravljicami (Animal Tales)? Kaj imajo skupnega?

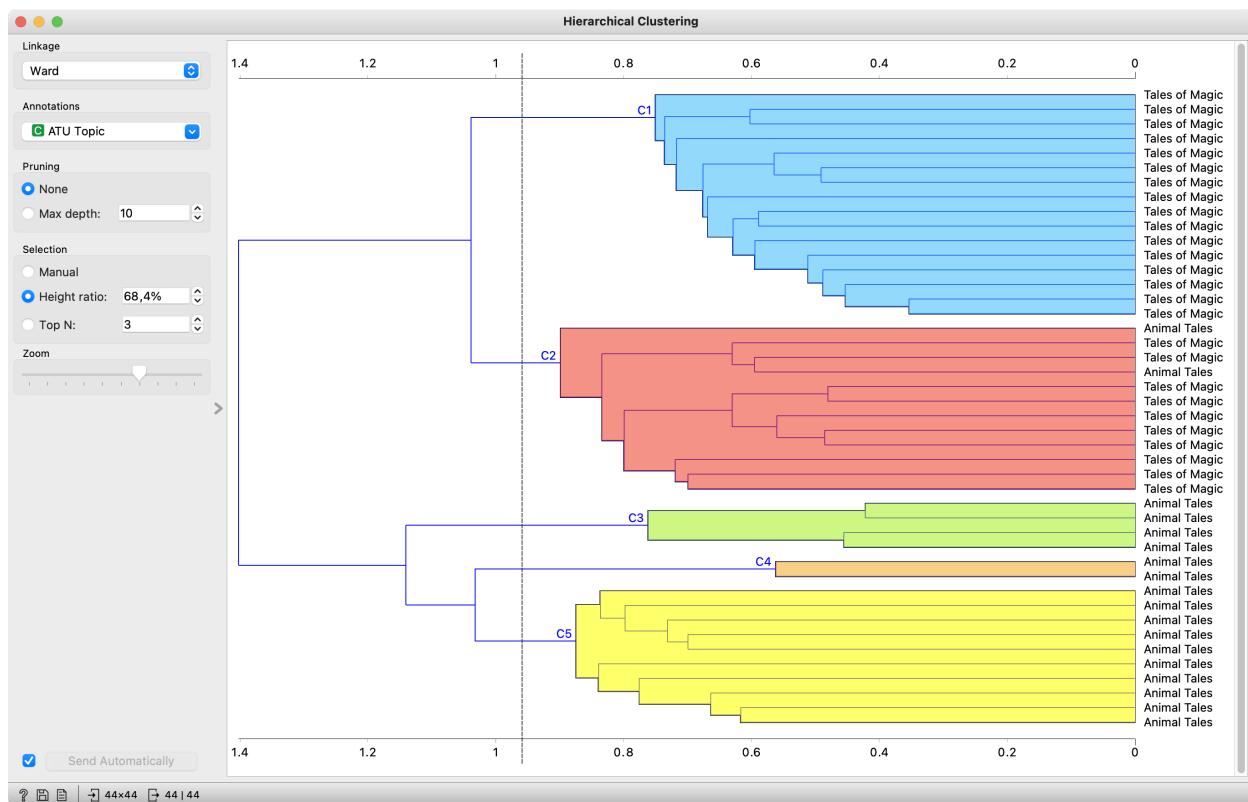
Enak delotok lahko preizkusite tudi na drugih podatkih, na primer na bookexcerpt.tab, ki vsebuje izvlečke knjig za odrasle in otroke.

Beseda dendrogram je sestavljena iz grških besede dendro "drevo" in gramma "risba" in pomeni hierarhično vizualizacijo v obliki drevesa.

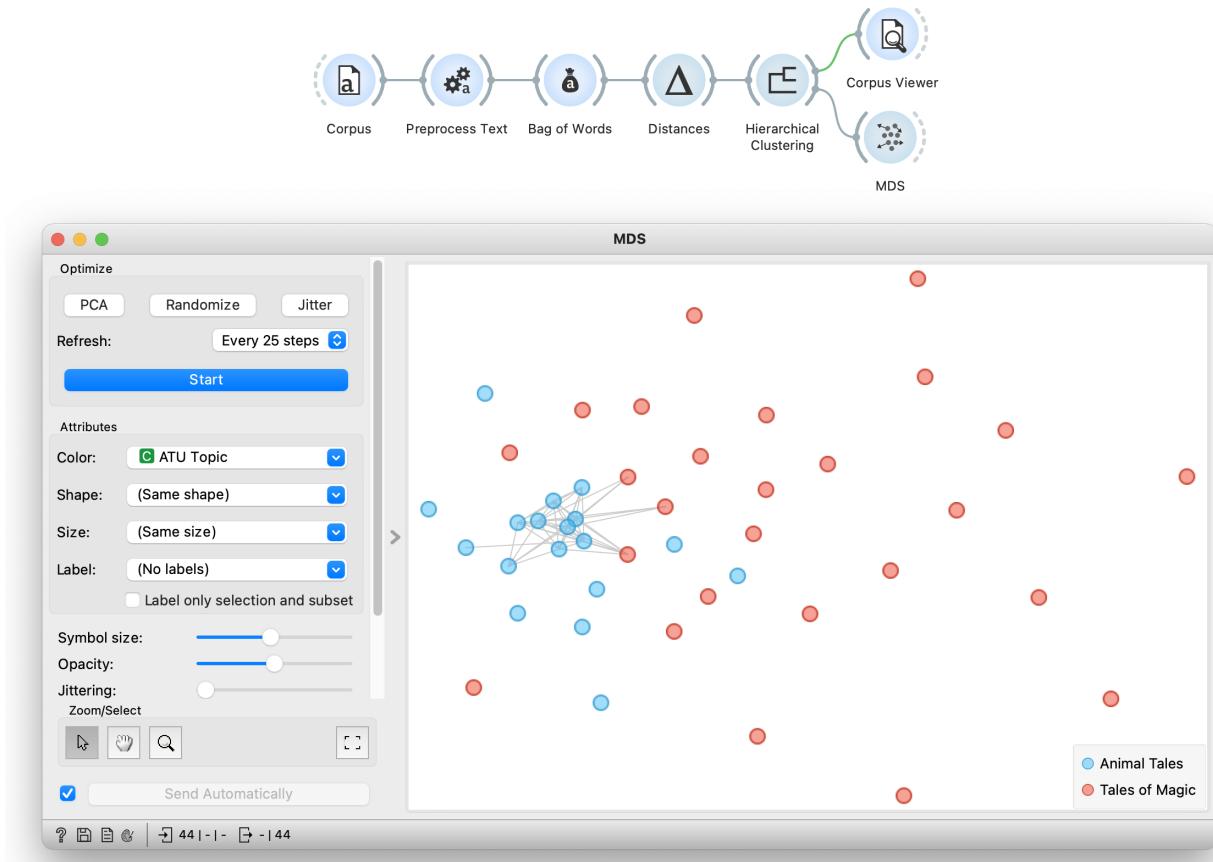


Hierarhično razvrščanje zgradi hierarhijo dokumentov, mi pa se moramo odločiti, kje zamejimo podobnost znotraj skupine. Mejo podobnosti nastavimo tako, da na ravnili zgoraj povlečemo črto desno ali levo in s tem zamejimo skupine.

Odločili smo se za pet skupin, saj po tem razdalja med skupinami kar precej naraste. Primerjajte pet skupin s štirimi, šestimi ali sedmimi. Gruče, ki jih odkrijemo, hkrati sovpadajo z oznako tipa Aarne-Thompson (ATU Topic).



Kako blizu pa so si v resnici živalske pravljice iz tretje in te iz četrte skupine? Ali ne bi bilo bolj zanimivo pogledati dokumente v ravnini, kjer bi se podobni dokumenti nahajali skupaj, različni pa narazen? Taka vizualizacija se imenuje večrazsežnostno lestvičenje oziroma Multidimensional Scaling (MDS).



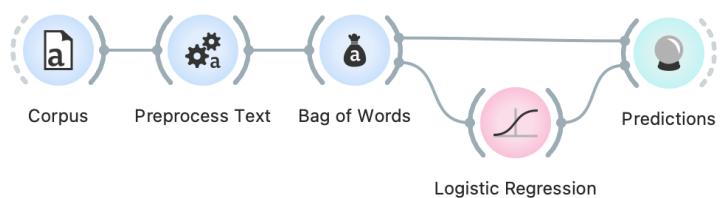
Magične pravljice tvorijo eno skupino, živalske pa drugo - tako kot smo pričakovali. Zanimivo, magične pravljice so si med sabo bolj podobne kot živalske (bolj so povezne). Raziščite podobne pravljice tako, da jih izberete v vizualizaciji in jih pogledate v gradniku Corpus Viewer.

Klasifikacija

Aarne and Thompson sta folklorista, ki uvedla sistem klasifikacije ljudskih pravljic glede na podlagi motiva. Sistem je v uporabi od leta 1810 je pogosto v uporabi v primerjalni folkloristiki. Končna U v ATU pomeni Uther, ki je leta 2004 zadnji posodobil indeks.

Omenili smo že tip Aarne-Thompson (ATU) To je indeks folklornih motivov in naše pravljice so že označene z visokonivojskim (žanr) in srednjenvojskim motivom (podžanr).

Ali bi morda lahko napovedali ATU tip na podlagi vsebine pravljice? Pa poglejmo.



Za začetek potrebujemo ciljno spremenljivko. To je spremenljivka, ki jo želimo napovedati, v našem primeru tip ATU. Potrebujemo tudi številsko reprezentacijo besedila - vrečo besed, ki nam jo je izračunal gradnik Bag of Word.

Sedaj bomo zgradili napovedni model. Model na podlagi enot (besed) napove ciljno vrednost (tip ATU). Vsak model potrebuje tudi postopek, ki definira, kako model upošteva besede. V našem primeru bo to logistična regresija.

V gradniku Predictions vidimo stolpec, kjer so napovedane vrednosti po postopku logistične regresije. Izgleda, da je naš model pravilno napovedal večino pravljic.

The screenshot shows a software window titled "Predictions". On the left, a sidebar says "Show probabilities for" with options "Animal Tales" and "Tales of Magic". Below this is a blue button "Restore Original Order". The main area has a table titled "Logistic Regression" with 9 rows. The columns are "ATU Topic", "Title", "Abstract", and several hidden columns. The rows show predictions like "0.00 : 1.00 → Tales of M..." for various tales. At the bottom, there's a summary table with columns "Model", "AUC", "CA", "F1", "Precision", "Recall", and a final column. The "Logistic Regres..." row has values 1.000 across all columns. The bottom status bar shows "44 | 44" and "44 | 1x44".

Logistic Regression	
1	0.00 : 1.00 → Tales of M...
2	0.00 : 1.00 → Tales of M...
3	1.00 : 0.00 → Animal Tales
4	0.00 : 1.00 → Tales of M...
5	0.00 : 1.00 → Tales of M...
6	1.00 : 0.00 → Animal Tales
7	0.00 : 1.00 → Tales of M...
8	0.00 : 1.00 → Tales of M...
9	0.00 : 1.00 → Tales of M...

Model	AUC	CA	F1	Precision	Recall
Logistic Regres...	1.000	1.000	1.000	1.000	1.000

Logistična regresija

V zgornjem delotoku smo uporabili logistično regresijo, priljubljeno metodo strojnega učenja. Pogosto se uporablja v rudarenju besedil zaradi hitrosti in napovedne točnosti. Kako pa deluje?

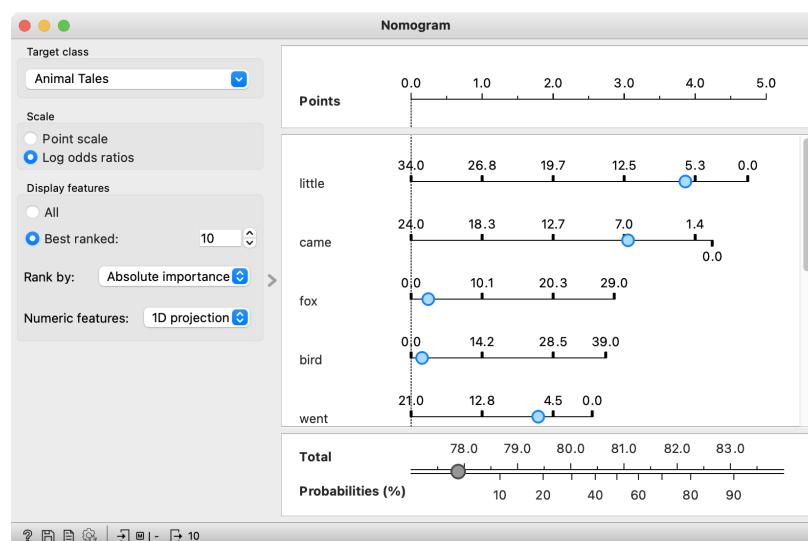


Pri postopku besede glasujejo. Na primer, beseda 'fox' v besedilu glasuje, da je pravljica živalskega tipa. Enako glasujejo mačke, ptiči in volkovi, ampak manj močno (črte v vizualizaciji so krajše). Lisica je očitno najboljši namig, da je pravljica živalska.

Beseda "little" glasuje obratno. Ravno tako beseda "came" (opazite, kako imajo te besede ničlo na desni strani črte?). Na drugi strani če se v pravljici pojavi več lisic, bolj verjetno bo pravljica živalska.

Vsaka beseda prispeva h končni oceni. Če v pravljici ni besede "majhen", potem živalskim dodeli 5 točk. In če je v besedilu 29 lisic, potem model dodeli 3 točk za živalske pravljice.

Jasno je dejanska metoda nekoliko bolj zapletena, saj poskuša najti pravilno ravnotežje med utežmi glasov in mejami odločitve. Ampak to vsebuje kot volk strašno linearno algebro, tako da se raje ne bomo podali na to pot.



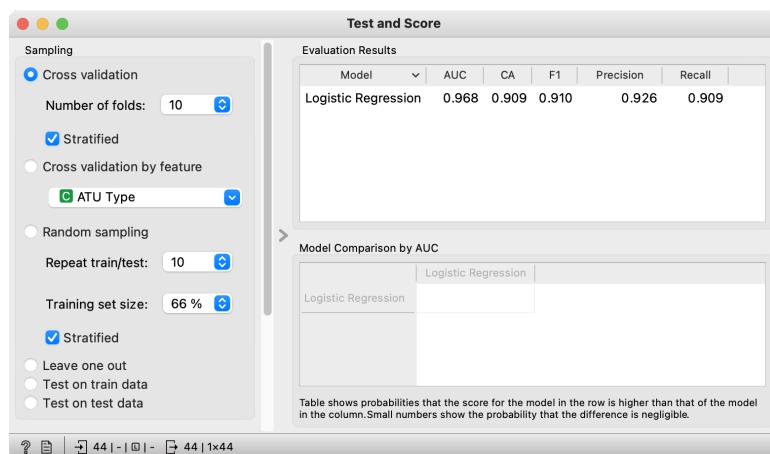
Vizualizacija se imenuje nomogram in prikaže točke (glasove) za najboljših 6 spremenljivk za ciljno vrednost, ki je izbrana levo zgoraj

V gradniku Nomogram lahko interaktivno opazujete odločitve modela. Povlecite modre točke levo ali desno tako, da bo končna vsota točk (Total) kar največja.

Ocenjevanje modelov

Pogledali smo si logistični model - lisice, ptice, kralje. Shema glasovanja je izgledala smiselna. Pred tem pa smo videli napovedi modela. Tudi te so bile točne. Kako pa bi lahko izračunali točnost modela?

Morda lahko izračunamo zgolj razmerje pravilno napovedanih pravljic? Taka mera se imenuje napovedna točnost (classification accuracy). Na primer, če smo pravilno napovedali 40 pravljic izmed 44, bo naša napovedna točnost $40/44$ oziroma 91%.

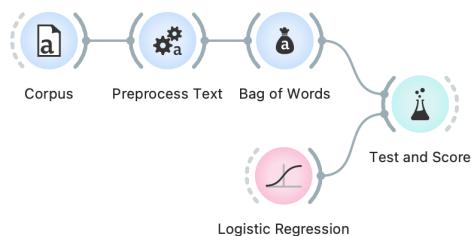


AUC je ena izmed boljših mer točnosti. Na kratko, bližje ko je mera vrednosti 1, boljša je točnost modela.

Gradnik za računanje točnosti modela se imenuje Test & Score. Potrebuje dva vhoda: podatke za testiranje modela in napovedni postopek.

"Ni ravno smiselno, da vprašamo model, ali je Zlatolaska živalska pravljica, če smo modelu že pred tem povedali, da je magična."

Kaj nismo tega storili zgoraj v gradniku Predictions? Prav zares in zato so bile napovedi tako dobre. Modelov se nikoli ne sme preverjati na podatkih, ki smo jih jim dali za učenje.



Tokrat logistična regresija ne potrebuje vhodnih podatkov. Namesto tega bo gradnik podal postopek za gradnjo modela. Nato bo Test & Score v več ponovitvah uporabil postopek na različnih podmnožicah podatkov. V vsaki ponovitvi bo naučil model na izbrani podmnožici in uporabil podatke izven množice za testiranje. Ni ravno smiselno, da vprašamo model, ali je Zlatolaska živalska pravljica, če smo modelu že pred tem povedali, da je magična.

Obogatitev besed

Sedaj o modelu vemo že veliko, ampak kaj ne bi bilo super pogledati, za katere pravljice se je model zmotil in za katere je imel prav? Gradnik Confusion Matrix omogoča prav to!

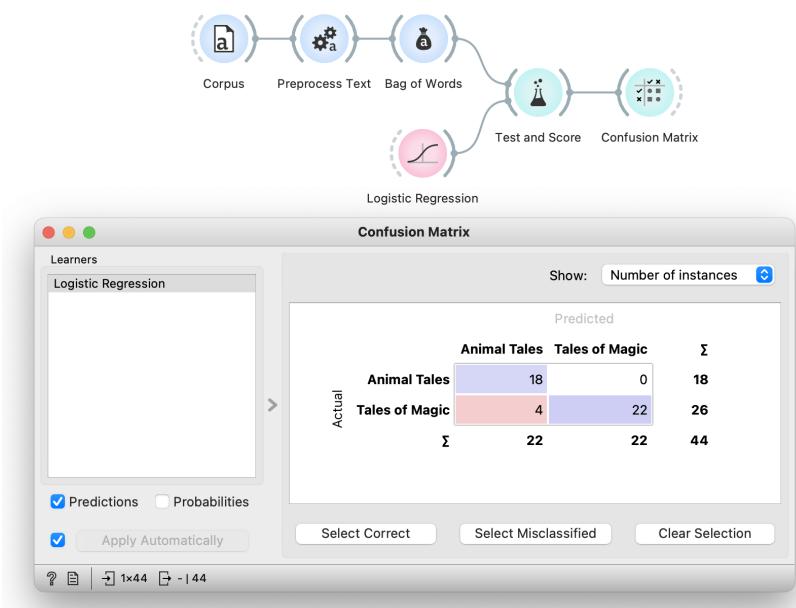


Tabela vsebuje pravilno napovedane dokumente v diagonali (modra) in nepravilno napovedane dokumente izven diagonale (rdeča). Vidimo, da so bile 4 živalske pravljice napovedane kot magične in 3 magične kot živalske.

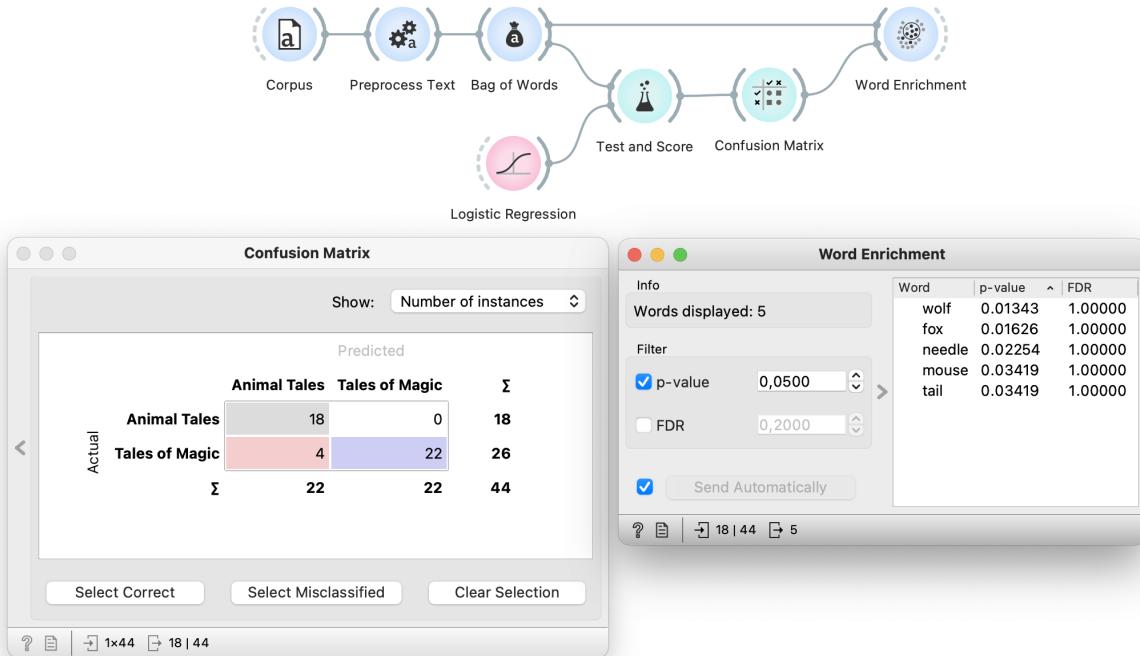
Ampak zakaj? Kaj je tako drugačnega na teh pravljicah, da jih model ni uspel razvrstiti v pravi razred?

Da preverimo napačno klasificirane dokumente, lahko uporabimo Corpus Viewer. Ali pa še bolje, pogledamo, katere besede so značilne za katero podmnožico!

Obogatitev besed primerja podmnožico dokumentov s celotnim korpusom in odkrije besede, ki statistično signifikantno zaznamujejo izbrano podmnožico.

Posamezne napačno klasificirane pravljice pogledamo s klikom na polje v tabeli.

Vse napačno klasificirane dokumente lahko izberemo z gumbom 'Select Misclassified'. To bo na izhod poslalo vse napačno klasificirane pravljice, ki jih nato pogledamo v gradniku Corpus Viewer.



Word Enrichment deluje na kakršni koli podmnožici. V gradniku Corpus Viewer poiščite vse dokumente, ki vsebujejo besedo queen. Sedaj pošljite izbor v Word Enrichment. Ne pozabite povezati celotnega korpusa iz gradnika Bag of Words - Word Enrichment potrebuje celoten korpus za primerjavo.

Besedi wolf in fox najbolj zaznamujta pravilno napovedane živalske pravljice. Seznam je nekoliko daljši za pravilno napovedane magične pravljice – king, beautiful, man, itd. Rezultati so zelo podobni tistim iz nomograma. Pravzaprav je to samo drugačen način, kako raziščemo model!

Torej ko boste naslednjič videli lisico v besedilu, ste lahko precej prepričani, da gre za živalsko pravljico! :)

Npovedovanje

todo: preveri kaj delas drugace?

Bibliography

Index

license, [2](#)