Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tomislav Slijepčević

Klasifikacija biomedicinskih člankov z globokimi modeli

magistrsko delo

magistrski program druge stopnje
računalništvo in informatika

Mentor: prof. dr. Blaž Zupan

Ljubljana, 2018

Zahvala

Zahvaljujem se mentorju prof. dr. Blaž Zupan za pomoč in strokovno vodenje. Zahvala gre tudi osebju laboratorija za bioinformatiko, ki mi je ponudilo strojno opremo in mi pomagalo pri razvoju komponente za programsko okolje Orange.

Še posebej bi se zahvalil družini, ki mi je nudila moralno podporo tekom celotnega študija.

Tomislav Slijepčević, 2018

Družini.

Kazalo

Povzetek

Abstract

1 Uvod **1**

2 Podatki **3**

2.1 Vir podatkov 3

2.2 Predpriprava podatkov 5

2.3 Razdelitev podatkov na podmnožice 6

3 Metode **7**

3.1 Model s porazdeljenim spominom 7

3.2 Model s porazdeljeno vrečo besed 10

3.3 Model konvolucijske nevronske mreže 11

3.4 Mera točnosti modela 17

3.5 Ocenjevanje kvalitete vektorskih predstavitev 18

4 Rezultati in razprava **19**

4.1 Izbor arhitekture konvolucijske mreže 19

4.2 Primerjava uspešnosti napovedovanja pripisov MeSH 22

4.3 Analiza kvalitete vektorskih predstavitev 25

5 Sklepne ugotovitve **31**

A Podrobni rezultati napovedovanja pripisov MeSH **37**

B Dodatne projekcije t-SNE **45**

C Implementacije modelov **49**

C.1 Večrazredna logistična regresija 49

C.2 Konvolucijska nevronska mreža 50

D Uporaba modela v programskem okolju Orange3 **51**

D.1 Gručenje vektorjev besedil 51

D.2 Vizualizacija vektorjev besedil 54

Povzetek

Naslov: Klasifikacija biomedicinskih člankov z globokimi modeli

V magistrskem delu smo razvili model, ki lahko besedila s področja znanosti v življenju predstavi v vektorski obliki, ki je primerna za uporabo v strojnem učenju. Naša ciljna skupina besedil so bili povzetki člankov iz zbirke MEDLINE, kjer so povzetki člankov označeni s pripisi iz ontologije MeSH. Razviti model uporablja globoko nevronsko mrežo za napovedovanje pripisov iz besedil. Za vektorsko predstavitev besedil smo uporabili predzadnji nivo mreže s 1000 nevroni. Model smo primerjali z večrazredno logistično regresijo, ki pripise MeSH napove iz vektorskih predstavitev besedil od modelov doc2vec. V poskusih napovedovanja pripisov MeSH na testni množici je točnost našega modela boljša. Prav tako so vektorske predstavitve besedil od našega modelom v primerjavi z vektorskimi predstavitvami besedil od modelov doc2vec boljše v točkovnih vizualizacijah z metodo t-SNE.

Ključne besede

biomedicinska literatura, vektorska predstavitev besedil, globoko učenje, napovedovanje pripisov MeSH

Abstract

Title: Deep Models for Classification of Biomedical Documents

In this master thesis, we developed a model that can present texts from life sciences in the vector form that is suitable for machine learning. Our corpus were abstracts from the MEDLINE collection, where abstracts are labeled with annotations from the MeSH ontology. The developed model uses a deep neural network for predicting MeSH annotations from a text. For the vector representation of a text, we used penultimate layer of a network that has 1000 neurons. The model was compared to the multinomial logistic regression, which predicts MeSH annotations from vector representations of texts that are obtained with doc2vec. In the task of predicting MeSH annotations on the test dataset, our model achieved higher accuracy. Also, vector representations of texts obtained with our model were in comparison with vector representations of texts obtained with doc2vec, better in point-based visualizations using the t-SNE method.

Keywords

biomedical literature, vector representation of text, deep learning, prediction of MeSH terms

Poglavje 1

Uvod

Uspeh tekstovnega rudarjenja je zelo odvisen od vektorske predstavitve besedil, zato je običajno veliko truda vloženo v izluščanje informativnih značilk [3]. Tekstovni podatki so vektorsko najpogosteje predstavljeni z vrečo besed [10]. Ta besedilo predstavi z značilkami, ki označujejo pogostost besed v besedilu. Predstavitev je preprosta, intuitivna in učinkovita, vendar ima zelo veliko značilk, saj ima vsaka beseda svojo značilko. V nasprotju porazdeljena predstavitev besedil [7] besedilo predstavi z manj značilkami, ki predstavljajo prikrite in izluščene vzorce iz besedila. Predstavitev se pridobi z globokimi modeli strojnega učenja, ki uporabljajo večnivojsko nevronsko arhitekturo. Takšna arhitektura modelu omogoča oblikovanje predstavitve, ki je lahko koristna v nadaljni analizi podatkov. Oblikovano predstavitev je težje tolmačiti od vreče besed, vendar v praksi kot kaže deluje zelo dobro [19, 14].

Globoki modeli so lahko nadzorovani ali nenadzorovani. Nenadzorovani se učijo podatke predstavitvi na podlagi označenih podatkov, nenadzorovani pa na podlagi neoznačenih podatkov. Za porazdeljeno predstavitev besedil sta bila v delu Le in Mikolov [19] predlagana dva zelo učinkovita nenadzorovana modela, ki se imenujeta "model s porazdeljenim spominom" (angl. Distributed Memory Model) in "model s porazdeljeno vrečo besed" (angl. Distributed Bag of Words Model). Ideja obeh modelov je oblikovati predstavitev besedil, ki je koristna za napovedovanje prisotnosti besed v besedilu. Zaradi nenad-

zorovanega učenja sta predstavitvi zelo splošni in sta posledično uporabni za različne naloge, kot so naloge napovedovanja, iskanja, ali razvrščanja v skupine. Predstavitvi sta se v primerjavi z vrečo besed in drugimi tradicionalnimi predstavitvami besedil izkazali za učinkovitejši [18, 16, 7].

Na področju biomedicine so mnoga znanstvena in strokovna besedila indeksirana s pripisi MeSH (angl. Medical Subject Headings) [27], zato je besedila smiselno predstaviti v predstavitvi, ki je koristna za napovedovanje teh pripisov. V takem primeru je bolje uporabiti nadzorovane modele, ki za razliko od nenadzorovanih modelov predstavitev oblikujejo glede na oznake primerov [26]. Zato smo v nalogi razvili nadzorovani model za vektorsko predstavitev biomedicinskih besedil, ki predstavitev oblikuje glede na pripise MeSH. Naloga je sorazmerno težka, saj je pripisov več kot 28.000. Za razliko od modelov doc2vec, ki uporablja navadne polno povezane nivoje, smo za naš model uporabili konvolucijske nivoje, ki trenutne dosegajo najboljše rezultate v raznih nalogah z različnimi tipi podatkov, med drugim v nalogah s slikami [17], zvokom [11], senzorskimi meritvami [1], in besedili [14, 13, 30, 28]. V nalogi smo primerjali kvaliteto vektorske predstavitev našega modela in modelov doc2vec v napovedovanju pripisov MeSH in ločevanju povzetkov glede na različne pripise MeSH. Za učenje predstavitev smo uporabili povzetke indeksiranih znanstvenih in strokovnih člankov iz zbirke MEDLINE, ki vsebuje več kot 14,5 milijonov povzetkov. Za enostavno uporabo našega modela smo model implementirali kot komponento v programskem okolju Orange [8].

Magistrsko delo je sestavljeno iz petih poglavij in prilog. V poglavju 2 predstavimo uporabljene podatke in predpripravo podatkov. Nato v poglavju 3 predstavimo uporabljene modele in vrednotenje njihove uspešnosti. V poglavju 4 podamo rezultate in razpravljamo o ugotovitvah. Delo zaključimo s poglavjem 5. V prilogah predstavimo uporabo modela v programskem okolju Orange.

Poglavje 2

Podatki

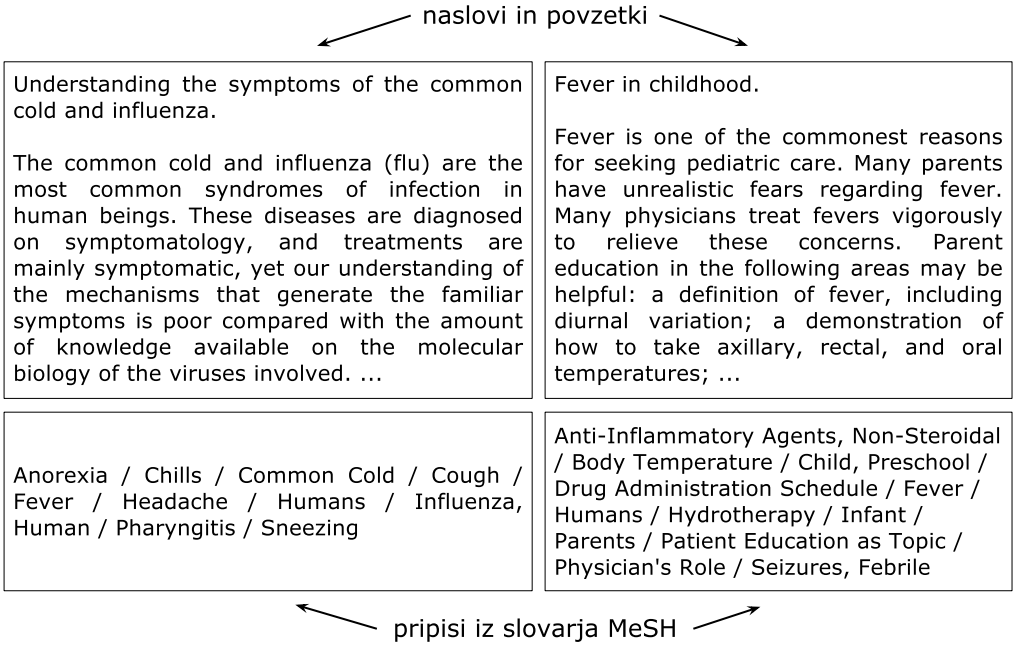
V poglavju predstavimo uporabljene podatke, predpripravo podatkov in razdelitev podatkov na učno, testno ter validacijsko množico.

2.1 Vir podatkov

Podatke smo pridobili iz obsežne podatkovne zbirke MEDLINE[1], ki indeksira znanstvene in strokovne članke s področja ved o življenju. Zbirka primarno hrani bibliografske informacije, vendar ima tudi mnogo povzetkov teh člankov. Za lažje poizvedovanje po zbirki so članki indeksirani s slovarjem pripisov MeSH, ki vsebuje več kot 28.000 pripisov. Za indeksiranje je zadolžena skupina strokovnjakov, ki to počne deloma ročno, deloma samodejno z uporabo računalniškega programa Medical Text Indexer[2]. Primer povzetkov in pripisov je prikazan na sliki 2.1.

Od podatkov v zbirki smo uporabili naslove, povzetke in pripise. V nalogi smo se ukvarjali z angleškimi besedili, zato smo iz zbirke uporabili samo podatke člankov, ki so napisani v angleščini. Teh je na dan 30. 6. 2017 bilo 14.513.202. Za vsak pripis smo želeli imeti vsaj deset tisoč primerov, zato smo ustrezno odstranili manjkrat uporabljene pripise. Po odstranjevanju nam je

[1] `https://www.nlm.nih.gov/pubs/factsheets/medline.html`
[2] `https://ii.nlm.nih.gov/MTI/`



Slika 2.1: Primera indeksiranih člankov iz zbirke MEDLINE. Na sliki sta prikazana naslova, povzetka in pripisi člankov. Indeksi so ločeni s poševnico.

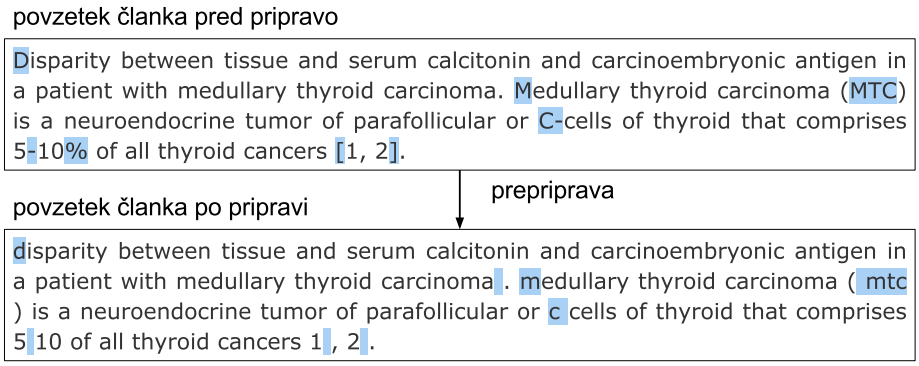
ostalo 2.890 pripisov. V slovarju pripisov so pripisi razvrščeni v skupine, ki so prav tako pripisi. Skupine pripisov in število uporabljenih pripisov znotraj posamezne skupine je prikazano v tabeli 2.1. Članki so v povprečju indeksirani z 9,5 uporabljenih pripisov.

Tabela 2.1: Skupine pripisov in število uporabljenih pripisov znotraj posamezne skupine.

Skupine pripisov	Število pripisov
Chemicals and Drugs	763
Analytical), Diagnostic and Therapeutic Techniques and Equipment	566
Biological Sciences	503
Diseases	356
Anatomy	308
Health Care	270
Psychiatry and Psychology	162
Organisms	112
Information Science	73
Anthropology), Education), Sociology and Social Phenomena	58
Technology and Food and Beverages	54
Physical Sciences	47
Geographic Locations	45
Persons	35
Humanities	5

2.2 Predpriprava podatkov

Besedila smo predpravili na način, priporočen za učenje na angleških besedilih [14], ki: (1) odstrani znake, ki niso črke, števila ali ločila, (2) loči besede od ločil, (3) loči združene besed kot na primer “they’re”, (4) odstrani podvojene presledke in odvečne presledke na koncema besedila, (5) ter na koncu pretvori velike črke v male. Primer predpriprave povzetka članka je prikazan na sliki 2.2. Predpriprava besedil običajno zahteva še odstranitev manj pogostih besed, saj so sicer modeli preveliki za učenje. V našem primeru smo morali odstraniti besede, ki so se pojavile manj kot 14-krat, da smo lahko



Slika 2.2: Primer predpriprave povzetka članka, kjer so spremembe obarvane z modro.

modele učili z našimi računskimi viri. Od 2.703.192 različnih besed nam je ostalo 415.253 (15,36%) besed, od skupaj 2.165.549.630 besed uporabljenih v besedilih pa nam je ostalo 2.159.268.084 (99,7%) besed. Po odstranjevanju besed je mediana števila besed v besedilih znašala 229 besed, 99-ti centil pa 498 besed.

2.3 Razdelitev podatkov na podmnožice

Naš korpus 14.513.202 člankov oziroma njihovih povzetkov smo naključno razdelili na učno, validacijsko in testno množico. Najprej smo primere razdelili v učno množico z 80% primerov in testno množico z 20% primerov. Validacijsko množico smo ustvarili iz 20% primerov učne množice, kar je 16% vseh primerov, pri čemer je učni množici ostalo 64% primerov. V učni množici je tako 9.288.448 primerov, v testni množici 2.902.641 primerov in v validacijski množici 2.322.113 primerov. Učno množico smo uporabili za učenje modelov. Validacijsko množico smo uporabili za ovrednotenje uspešnosti modelov med učenjem in za zgodnjo prekinitev učenja, če se uspešnost modela na validacijski množici ni izboljševala. Testno množico smo uporabili za ovrednotenje uspešnosti naučenih modelov.

Poglavje 3

Metode

Za porazdeljeno vektorsko predstavitev biomedicinskih besedil smo v delu razvili globok model, ki predstavitev oblikuje na podlagi besedil označenih s pripisi MeSH. Predstavitev novega modela smo primerjali s predstavitvama modelov doc2vec [19], ki predstavitev oblikujeta na podlagi neoznačenih besedil. Modela doc2vec se imenujeta "model s porazdeljenim spominom" (angl. Distributed Memory Model) in "model s porazdeljeno vrečo besed" (angl. Distributed Bag of Words Model).

3.1 Model s porazdeljenim spominom

Model s porazdeljenim spominom temelji na modelih za učenje porazdeljene vektorske predstavitve besed. Primer arhitekture teh modelov je prikazan na sliki 3.1. Njihova naloga je napovedati besedo glede na sobesedilo. V takem modelu je beseda w_i preslikana v vektor, ki je predstavljen kot i-ti stolpec v matriki $W \in \mathbb{R}^{d \times |V|}$, kjer je i indeks besede v slovarju V, d je dimenzija vektorjev in $|V|$ je število vseh besed v slovarju. Recimo, da model sprejme besedilo kot zaporedje besed $w_1, w_2, \ldots, w_T$, kjer je T število besed v besedilu. Cilj modela je maksimirati povprečno logaritmično verjetnost:

$$\frac{1}{T} \sum_{t=k}^{T-k} \log p(w_t | w_{t-k}, \ldots, w_{t+k}),$$

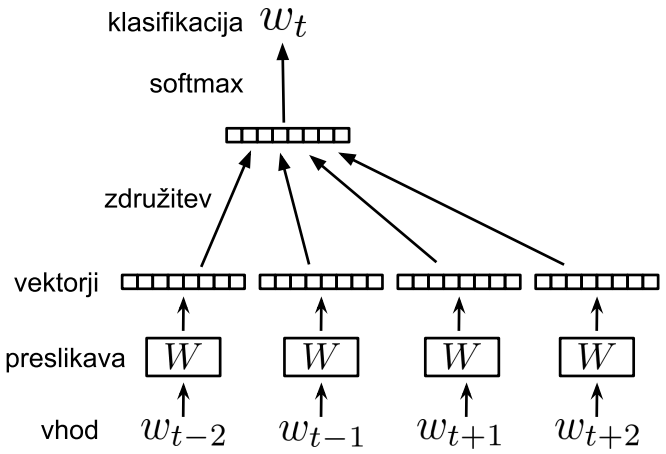
kjer je k velikost okna besed in je p verjetnost napovedi modela, da se beseda w_t pojavi ob prejšnjih k besedah in naslednjih k besedah. Verjetnost p je izračunana z večrazrednim klasifikatorjem, kot je na primer softmax:

$$p(w_t|w_{t-k}, \ldots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i^T e^{y_i}},$$

kjer je y_i nenormalizirana verjetnost napovedi modela za besedo z indeksom i. Verjetnost y_i se izračuna kot:

$$y_i = b + Uh(w_{t-k}, \ldots, w_{t+k}; W),$$

kjer sta U in b parametra klasifikatorja softmax in je h stik, povprečje ali vsota vektorjev besed iz W. Sobesedila so fiksne dolžine in so vzorčena z oknom, ki se pomika po besedilu. Vektorji besed so skupnim vsem besedilom. Parametri modela W, U in b so pridobljeni s stohastičnim gradientnim spustom, pri čemer je gradient izračunan z vzvratnim razširjanjem napake [24]. Model v vsaki iteraciji stohastičnega gradientnega spusta vzorči sobesedilo iz naključnega besedila, izračuna gradient napake in ga uporabi za posodobitev parametrov modela.

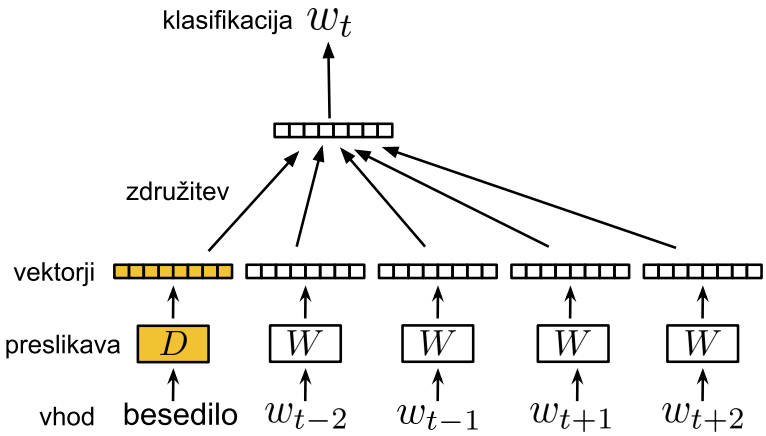


Slika 3.1: Arhitektura modela za učenje vektorjev besed. Za napovedovanje besede iz besedila uporabi vektorje sosednjih besed, ki so predstavljeni kot stolpci v matriki W.

Model s porazdeljenim spominom razširja to idejo tako, da za napoved besede uporabi tudi vektor besedila. V tem modelu je besedilo d_j preslikano v vektor, ki je predstavljen kot j-ti stolpec v matriki $D \in \mathbb{R}^{d \times |C|}$, kjer je j indeks besedila v korpusu besedil C, d je dimenzija vektorjev in $|C|$ je število vseh besedil v korpusu. Model za napoved besede uporabi vektor, ki je povprečje, vsota ali stik vektorjev besed in vektorja besedila. Od modelov za učenje vektorjev besed se razlikuje le v napovedovanju verjetnosti y_i, da se beseda w_i pojavi ob besedah $w_{t-k}, \ldots, w_{t+k}$, kjer upošteva tudi vektor besedila d_j:

$$y_i = b + Uh(w_{t-k}, \ldots, w_{t+k}, d_j; W)$$

Vektorji besed so skupnim vsem besedilom tako kot v prejšnjem modelu, medtem ko je posamezen vektor besedila skupen vsem sobesedilom iz besedila. Parametri modela W, D, U in b so pridobljeni s stohastičnim gradientnim spustom in vzvratnim širjenjem napake. Po zaključenem učenju, ko model prejme novo besedilo, zanj doda stolpec v matriko D in ga nato spreminja z vzorčenjem sobesedil in gradientnim spustom, pri čemer ne spreminja vektorjev besed in uteži klasifikatorja softmax.

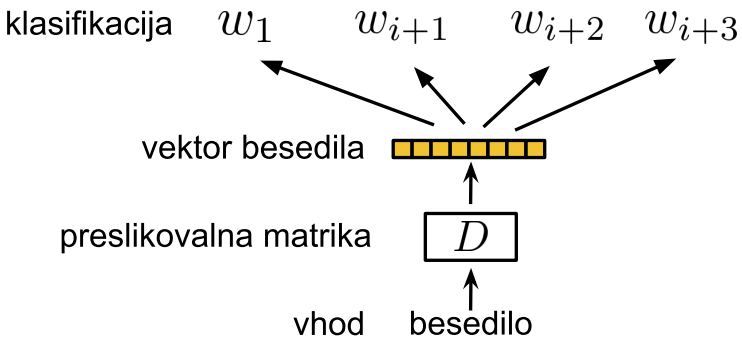


Slika 3.2: Arhitektura modela s porazdeljenim spominom. Za napovedovanje besede iz besedila uporabi vektorje sosednjih besed in vektor besedila, ki je predstavljen kot stolpec v matriki D.

V delu smo uporabili implementacijo modela iz programske knjižnice gensim [23]. Preizkusili smo vse tri načine združevanj vektorjev: združevanje s stikom, povprečjem in vsoto. Za napovedovanje besede smo uporabili 5 predhodnih in 5 naslednjih besed tako, kot je privzeto nastavljeno v knjižnici. Za vektorje besed in besedil smo izbrali iste dimenzije kot v modelu konvolucijske nevronske mreže.

3.2 Model s porazdeljeno vrečo besed

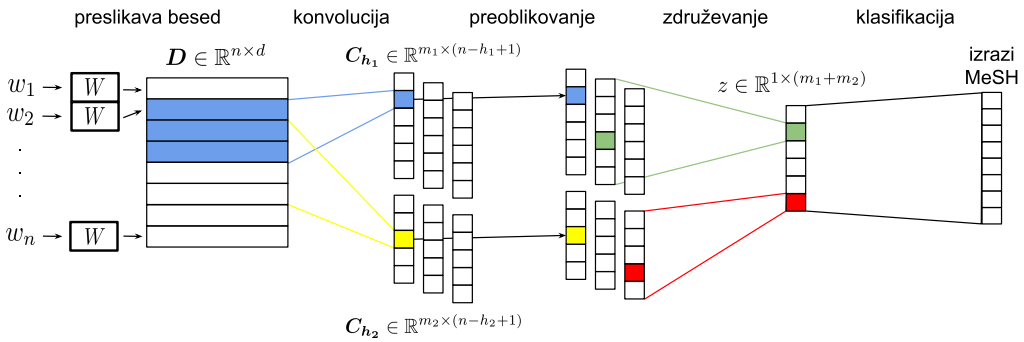
Model s porazdeljeno vrečo je poenostavitev prejšnjega modela, saj pri napovedovanju besede iz besedila ne uporabi sobesedilo, temveč besedo napove neposredno iz vektorja besedila. Arhitektura modela je prikazana na sliki 3.3. Model v vsaki iteraciji stohastičnega gradientnega spusta vzorči besede iz besedila in jih nato poskuša napovedati. Preslikava besedil v vektorje je enaka kot pri prejšnjemu modelu. Parametri so pridobljeni s stohastičnim gradientnim spustom in vzvratnim razširjanjem napake. V delu smo uporabili implementacijo modela s porazdeljeno vreča besed iz programske knjižnice gensim [23]. Za vektorje besedil smo izbrali isto dimenzijo kot v modelu konvolucijske nevronske mreže.



Slika 3.3: Arhitektura modela s porazdeljeno vrečo besed. Model se uči napovedati besede iz besedila na podlagi vektorja besedila.

3.3 Model konvolucijske nevronske mreže

Sestavni del našega modela so konvolucijski nivoji [20]. Ti nivoji se na podlagi primerov učijo izluščiti atribute, ki so koristni za izbrano klasifikacijsko nalogo. Sprva so se nivoji uporabljali v računalniškem vidu za odkrivanje atributov slik, kot so na primer krivulje ali obrazi, kasneje pa so se izkazale tudi na področju obdelave naravnega jezika v mnogih nalogah [5]. Za izhodiščno arhitekturo smo izbrali arhitekturo iz dela Yoon Kim [14], prikazano na sliki 3.4, ki se je izkazala v mnogih klasifikacijskih nalogah [14, 29]. V primerjavi z drugimi arhitekturami [6, 29, 13, 28] je ta arhitektura preprostejša, zato jo je lažje optimizirati za določeno nalogo. V tej arhitekturi model sprejme besedilo kot seznam besed dolžine n, kjer je n parameter modela. Če ima besedilo manj kot n besed, se seznamu doda prazno indikatorsko besedo tolikokrat, da bo dolžine n. V našem primeru smo n nastavili na 500, ker je 99-ti centil števila besed v besedilih iz učne množice znašal 498 besed. Učni proces modela je sestavljen iz preslikave besed v vektorje, konvolucije, preoblikovanja, združevanja, klasifikacije, regularizacije in učenja.



Slika 3.4: Arhitektura uporabljene konvolucijske nevronske mreže. Na vhodu so besede preslikane v vektorje besed, ki so predstavljeni kot stolpci v matriki W. Konvolucijski del mreže je sestavljen iz enega nivoja, ki uporablja različno velike konvolucijske filtre.

3.3.1 Preslikava besed

Na vhodu modela je beseda w_i preslikana v vektor, ki je predstavljen kot i-ta vrstica v matriki $W \in \mathbb{R}^{|V| \times d}$, kjer je i indeks besede v slovarju V, d je dimenzija vektorjev besed in $|V|$ je število besed v slovarju. Po preslikavi besed iz posameznega besedila so vektorji teh besed združeni v matriko $D \in \mathbb{R}^{n \times d}$ tako, da je v njeni j-ti vrstici vektor za j-to besedo iz besedila.

3.3.2 Konvolucija

Konvolucijski nivo iz vsakega besedila izlušči vzorce, ki so pogosti v učnih primerih. V našem primeru izlušči pogosta zaporedja besed. Za ta namen uporablja konvolucijske filtre, ki se tekom učenja samodejno naučijo, kateri vzorci so pogosti. Konvolucijski filtri so uporabljeni na vseh možnih zaporedjih besed določene dolžine. Zaporedja so pridobljena s pomičnim oknom, ki se pomika vzdolž vrstic matrike besedila D. Okno definiramo kot $D_{i:i+h-1} \in \mathbb{R}^{h \times d}$:

$$D_{i:i+h-1} = \begin{bmatrix} - & D_i & - \\ - & D_{i+1} & - \\ & \vdots & \\ - & D_{i+h-1} & - \end{bmatrix},$$

kjer je h parameter modela in označuje število besed v oknu. Konvolucijski filter je definiran kot matrika $F \in \mathbb{R}^{h \times d}$, ki je istih dimenzij kot okno. Prisotnost vzorca v oknu se izračuna s konvolucijo okna in konvolucijskega filtra, kjer je konvolucija z operatorjem $*$ definirana kot:

$$F * D_{i:i+h-1} = \sum_{j=0}^{i+h-1} \sum_{k=0}^{d-1} F_{j,k} D_{i+j,k}.$$

Rezultat konvolucije filtra na vseh zaporedij $\{D_{1:h}, D_{2:h}, \ldots, D_{n-h+1:n}\}$ je vektor $c = [c_1, c_2, \ldots, c_{n-h+1}]$ v $\mathbb{R}^{n-h+1}$, kjer je c_i rezultat konvolucije filtra in zaporedja $D_{i:i+h-1}$. Rezultatom pravimo atributne preslikave. Če želimo odkriti več vzorcev, moramo uporabiti več filtrov, ki so lahko različnih velikosti. Filtri iste velikosti se predstavi z matriko $F \in \mathbb{R}^{m \times h \times d}$, kjer je m

število filtrov in je h velikost filtrov. Rezultat konvolucije s tako matriko je matrika atributnih preslikav $C \in \mathbb{R}^{m \times (n-h+1)}$. Na sliki 3.4 sta prikazani dve skupini filtrov velikosti h_1 in h_2 ter pripadajoči matriki atributnih preslikav C_{h_1} in C_{h_2}. V eksperimentih smo preizkusili različne velikosti filtrov in različno število filtrov.

3.3.3 Preoblikovanje

V primeru ko je klasifikacijski problem nelinearen, moramo atribute preoblikovati z nelinearno funkcijo. V našem primeru smo z nelinearnimi funkcijami preoblikovali atributne preslikave. Za preoblikovanje smo preizkusili najpogostejše nelinearne funkcije: ReLU, hiperbolični tangens (tanh), sigmoidno funkcijo [21] in softplus. Poleg tega smo preizkusili, kako se obnese model brez uporabe nelinearnih funkcij. Po preoblikovanju so atributne preslikave podane združevalnemu nivoju.

3.3.4 Združevanje

Združevalni nivo se uporablja za združevanje atributnih preslikavah in za zmanjšanje njihovih dimenzionalnosti. Najpogostejša načina združevanja sta maksimalno in povprečno združevanje. Maksimalno združevanje izbere največjo vrednost, medtem ko povprečno združevanje izračuna povprečje vrednosti. Združuje se lahko tudi s k-maksimalnim združevanjem [13], ki je posplošitev maksimalnega združevanja in namesto največje vrednosti izbere k največjih vrednosti. Združevanje na matriki atributnih preslikav $C \in \mathbb{R}^{m \times (n-h+1)}$ poteka po vrsticah, kjer se nahajajo preslikave posameznih filtrov. Naš model združuje z maksimalnim združevanje, zato združevanje na matriki C pridela vektor $z = [\max(c_1), \max(c_2), \ldots, \max(c_m)]^T$, kjer je $\max(c_i)$ največja vrednost v atributni preslikavi iz vrstice i. Združevalni nivo je zadnji nivo pred klasifikacijo, zato vektor z predstavlja končno predstavitev besedila. V primeru da imamo več matrik atributnih preslikav $\{C_1 \in \mathbb{R}^{m_1 \times (n-h_1+1)}, C_2 \in \mathbb{R}^{m_2 \times (n-h_2+1)}\}$, je besedilo predstavljeno kot vektor

$z = z_1 \otimes z_2$ v $\mathbb{R}^{m_1+m_2}$, kjer je vektor z_i pridelan iz matrike C_i in je $\otimes$ operator za stolpično združevanje vektorjev.

3.3.5 Klasifikacija

Cilj učenja je iskanje parametrov modela, ki na učnih primerih minimizirajo logaritmično izgubo:

$$logloss = -\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[z_j^{(i)} \log y_j^{(i)} + (1 - z_j^{(i)}) \log(1 - y_j^{(i)}) \right],$$

kjer je n število učnih primerih, m je število razredov, $y_j^{(i)}$ je verjetnost napovedi j-tega nevrona, da je i-ti primer označen z j-tim pripisom MeSH, in $z_j^{(i)}$ je 1, če je i-ti primer označen z j-tim izrazom MeSH, sicer je 0.

3.3.6 Regularizacija

Za regularizacijo smo uporabili regularizacijo tipa L2 in zelo učinkovito tehniko izpuščanja nevronov (angl. dropout) [25]. Tehnika izpuščanja nevronov nevrone med učenjem izpušča z verjetnostjo p, kjer je p parameter modela. Po zaključenem učenju so uteži w med nivojem, ki uporablja to tehniko, in naslednjim nivojem zmanjšane za p: $\hat{w} = wp$. V našem primeru smo tehniko uporabili na vektorski predstavitvi besedila z v združevalnem nivoju.

3.3.7 Učenje

Učenje predstavljenega modela vključuje učenje vektorjev besed v preslikovalni matriki W, učenje konvolucijskih filtrov F in učenje uteži med združevalnim in izhodnim nivojem. Model smo učili s stohastičnim gradientnim spustom in vzvratnim širjenjem napake. Za gradientni spust smo uporabil optimizacijski algoritem Adam [15].

3.3.8 Implementacija

V preizkusih smo arhitekturo optimizirali za klasifikacijo pripisov MeSH, pri čemer smo izhajali iz arhitekture iz dela [14], ki uporablja filtre velikost 3, 4 in 5, za vsako velikost filtrov uporablja 100 filtrov in za preoblikovanje atributnih preslikav uporablja funkcijo ReLU. Pri optimiziranju smo se zgledovali po delu Zhang in sodelavci [29], kjer so raziskovalci prav tako izhajali iz iste arhitekture. V poskusih smo izbrali velikosti konvolucijskih filtrov [1], število konvolucijskih filtrov [2] in nelinearno funkcijo [3]. Poleg tega smo preverili učinkovitost tehnike izpuščanja nevronov [4]. Mrežo smo implementirali v programskem jeziku Python in s pomočjo programske knjižnice Keras [4]. Implementacija mreže v izhodiščni arhitekturi je prikazana v izseku kode 3.1, kjer so nivoji in koraki mreže definirani v naslednjih vrsticah:

- vhodni nivo je v 11. vrstici,
- preslikovalni nivo v 12. vrstici,
- konvolucijski nivoji s filtri velikost 3, 4 in 5 so v vrsticah od 14. do 17.,
- združevalni nivoji so v vrsticah od 19. do 22.,
- oblikovanje vektorja besedila iz izhodov združevalnih nivojev je v 24. vrstici,
- regularizacija s tehniko izpuščanja nevronov je v 25. vrstici,
- regularizacija tipa L2 je v 26. vrstici,
- izhodni nivo je v 26. vrstici,
- in učenje z optimizacijski algoritem Adam glede na logaritmično izgubo je v 30. vrstici.

[1] `https://github.com/tomislijepcevic/medline_embedding#region-sizes`
[2] `https://github.com/tomislijepcevic/medline_embedding#number-of-filters`
[3] `https://github.com/tomislijepcevic/medline_embedding#activation`
[4] `https://github.com/tomislijepcevic/medline_embedding#dropout`

```
from keras.models import Model
from keras.layers import Input, Dense, Dropout
from keras.layers.embeddings import Embedding
from keras.layers.convolutional import Conv1D
from keras.layers.pooling import GlobalMaxPooling1D
from keras.layers.merge import concatenate
from keras.constraints import max_norm
from keras.optimizers import Adam

input_layer = Input(shape=(500,))
embed_layer = Embedding(415253, 300)(input_layer)

conv_layers = []
for conv_size in [3, 4, 5]:
    conv_layer = Conv1D(100, conv_size, activation='relu')(embed_layer)
    conv_layers.append(conv_layer)

pool_layers = []
for conv_layer in conv_layers:
    pool_layer = GlobalMaxPooling1D()(conv_layer)
    pool_layers.append(pool_layer)

text_vector = concatenate(pool_layers)
text_vector = Dropout(0.5)(text_vector)
output_layer = Dense(2890, activation='sigmoid',
                     kernel_constraint=max_norm(3))(text_vector)

model = Model(inputs=input_layer, outputs=output_layer)
model.compile(loss='binary_crossentropy', optimizer=Adam())
```

Koda 3.1: Implementacija konvolucijske nevronske mreže z izhodiščno arhitekturo.

3.4 Mera točnosti modela

Uporabljene modele smo ovrednotili v napovedovanju pripisov MeSH [5]. Za modela doc2vec, ki ne napovedujeta pripisov, smo uporabili večrazredno logistično regresijo, ki pripise napove na podlagi njunih vektorskih predstavitev besedil. Večrazredno logistično regresijo smo implementirali z navadno nevronsko mrežo, ki ima polno povezan vhodni in izhodni nivo. Arhitekura mreže je strukturno enaka kot zadnji del konvolucijske nevronske mreže, ki pripise napove na podlagi vektorske predstavitve besedila iz predzadnjega nivoja. Večrazredno logistično regresijo smo učili z minimiziranjem logaritmične izgube [2] in stohastičnim gradientnim spustom tako kot model konvolucijske nevronske mreže.

Uspešnost napovedovanja za posamezni pripis MeSH smo ovrednotili s povprečno točnostjo [31]. Uvedimo množico A_k, v kateri je k primerov, za katere model napove največje verjetnosti, da so označeni s pripisom. Naj bo P_k točnost, ki je izražena kot delež primerov v A_k, ki so pozitivni:

$$P_k = \frac{|A_k \cap \text{pozitivni}|}{|A_k|}$$

Uvedimo še priklic R_k, ki je izražen kot delež pozitivnih v A_k:

$$R_k = \frac{|A_k \cap \text{pozitivni}|}{|\text{pozitivni}|}$$

Povprečna točnost AP je definirana kot:

$$AP = \sum_{k=1}^{n} P_k(R_k - R_{k-1}),$$

kjer je n število testnih primerov. Za vrednotenje skupne uspešnosti za vse pripise smo uporabili povprečje povprečnih točnosti.

[5] https://github.com/tomislijepcevic/medline_embedding/blob/master/CLASSIFICATION.md

3.5 Ocenjevanje kvalitete vektorskih predstavitev

Kvaliteto vektorskih predstavitev smo ovrednotili s silhuetno mero [6], ki ovrednoti, kako dobro so ločeni primeri iz različnih razredov. V našem primeru smo ovrednotili ločevanje besedil z različnimi pripisi MeSH. Za primerjavo smo izbrali pripise, ki so sorodni, kot so na primer različne bolezni pljuč in možganov, za vsak pripis pa smo izbrali tisoč povzetkov iz testne množice. Vektorske predstavitve smo primerjali v dveh dimenzijah, da bi lahko ločevanje besedil prikazali z razsevnimi diagrami. Za zmanjšanje dimenzionalnosti predstavitev smo uporabili zelo uspešno metodo t-Distributed Stochastic Neighbour (t-SNE) [22], ki pri zmanjšanju ohrani najbližje sosede iz izvornega vektorskega prostora. Silhuetna mera za vsak primer izračuna, kako blizu je primerom iz istega razreda v primerjavi s primeri iz drugih razredov. Naj bo a_i povprečna razdalja med primerom i in primeri iz istega razreda. Uvedimo še b_i, ki je povprečna razdalja med primerom i in primeri iz najbližjega razreda B, v katerem ni i. Silhueta s_i za primer i je definirana kot:

$$s_i = \frac{b_i - a_i}{max\{a_i, b_i\}},$$

in zavzame vrednosti od -1 do 1. Pozitivne vrednosti pomenijo, da je primer i bližje primerom iz istega razreda kot primerom iz razreda B, sicer velja obratno. Vrednost 0 pomeni, da je primer enako oddaljen od primerov iz istega razreda in primerom iz razreda B. Kot končno mero smo uporabili povprečje silhuet s_i od vseh primerov.

[6] `https://github.com/tomislijepcevic/medline_embedding/blob/master/VISUALIZATION.md`

Poglavje 4

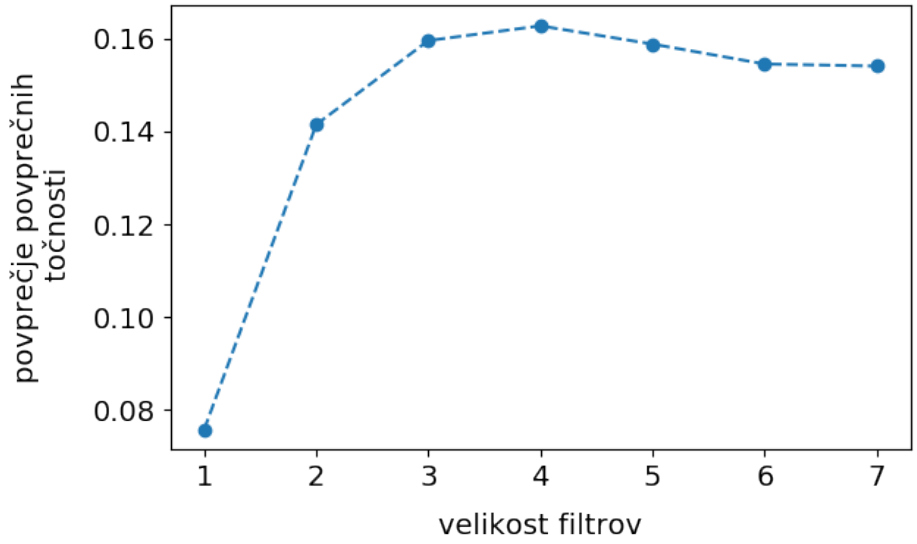
Rezultati in razprava

V poglavju najprej poročamo o rezultatih poskusov, s katerimi smo izbrali optimalno arhitekturo konvolucijske nevronske mreže za napovedovanje pripisov MeSH. Nato poročamo o uspešnosti modela konvolucijske nevronske mreže in modelov doc2vec v napovedovanja pripisov MeSH. Poglavje zaključimo z vizualizacijami, ki prikazujejo, kako dobro vektorske predstavitve modelov ločijo besedila z različnimi pripisi.

4.1 Izbor arhitekture konvolucijske mreže

V konvolucijskem nivoju mreže smo preizkusili različne velikosti in konfiguracije konvolucijskih filtrov. Najprej smo preizkusili velikosti od ena do sedem, pri čemer smo za vsako uporabili 100 konvolucijskih filtrov. Vpliv velikosti na uspešnost modela je prikazan na sliki 4.1, kjer vidimo, da se je za optimalno velikost izkazala velikost 4. Poleg tega smo preizkusili uporabo različnih konfiguracij konvolucijskih filtrov. V tabeli 4.1 vidimo, da je izmed konfiguracij (2, 3, 4), (3, 4, 5) in (4, 5, 6) najbolj učinkovita konfiguracija (3, 4, 5), ki ima velikosti najbližje optimalni velikosti. Še boljši rezultat smo dobili, če smo uporabili samo optimalno velikost z istim številom konvolucijskih filtrov (300), zato smo v nadaljevanju uporabili samo to velikost. Za nelinearno preoblikovanje atributnih preslikav smo preizkusili funkcijo ReLU,

hiperbolični tangens (tanh), sigmoidno funkcijo [21] in softplus [9]. Poleg tega smo preizkusili, kako se obnese model brez preoblikovanja. Za konvolucijo smo uporabili 100 filtrov velikosti 4, rezultati pa so prikazani v tabeli 4.2. Najbolje se je izkazal model s hiperboličnim tangensom, skoraj enako dobro pa se je izkazal tudi model brez preoblikovanja, kar pomeni, da modelu zadoščajo linearne transformacije za zajem soodvisnosti med vektorji besed in pripisi MeSH. Tehnika izpuščanja nevronov [25] na predzadnjem nivoju ni pomagala, zato je v končni arhitekturi nismo uporabili. Preverili smo tudi, kako se uspešnost modela spreminja glede na število konvolucijskih filtrov. Preizkusili smo od 200 do 1000 filtrov velikosti 4, rezultati pa so prikazani na sliki 4.2. Uspešnost se z večanjem filtrov izboljšuje, vendar ne toliko. Za končno arhitekturo konvolucijske mreže smo izbrali arhitekturo, ki za konvolucijo uporablja 1000 konvolucijskih filtrov velikosti 4 in za preoblikovanje atributnih preslikav uporablja hiperbolični tangens.



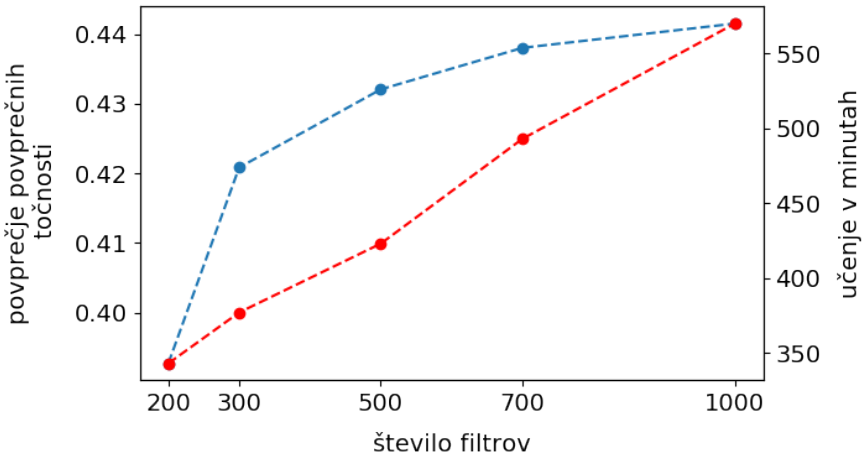
Slika 4.1: Vpliv velikosti konvolucijskih filtrov na uspešnost modela.

Tabela 4.1: Vpliv konfiguracij z različnimi velikostmi na uspešnost modela.

Velikost filtrov	Povprečje povprečnih točnosti
2, 3, 4	0.2639
3, 4, 5	0.2661
4, 5, 6	0.2591
4	**0.2681**

Tabela 4.2: Vpliv nelinearne funkcije na uspešnost modela.

Nelinearna funkcija	Povprečje povprečnih točnosti
ReLU	0.1627
sigmoid	0.1582
softplus	0.1496
tanh	**0.2159**
brez	0.2154



Slika 4.2: Vpliv števila konvolucijskih filtrov na uspešnost modela.

4.2 Primerjava uspešnosti napovedovanja pripisov MeSH

Rezultati povprečne uspešnosti modelov v napovedovanju pripisov MeSH so prikazani v tabeli 4.3. Napovedovanje z modelom konvolucijske nevronske mreže se je izkazala za uspešnejše od napovedovanja z večrazredno logistično regresijo, ki napoveduje na osnovi vektorskih predstavitev besedil od modelov doc2vec. Izmed različic modela s porazdeljenim spominom se je najbolje izkazala različica, ki uporablja povprečje kontekstnih vektorjev, in je po uspešnosti približno enaka modelu s porazdeljeno vrečo besed. Rezultati po skupinah pripisov MeSH so prikazani v tabeli 4.4, kjer vidimo, da je model konvolucijske nevronske mreže v vseh skupinah boljši od modelov doc2vec. V tabeli 4.5 so prikazani rezultati za posebne pripise [1], ki se člankom rutinsko dodajajo in zajemajo živalske vrste, spol, starostne skupine, zgodovinska obdobja ter nosečnost. Rezultati modelov doc2vec za te pripise so boljši od prejšnjih, verjetno zaradi večjega števila učnih primerov, vendar so vseeno slabši od rezultatov modela konvolucijske nevronske mreže. V prilogah so priloženi še rezultati napovedovanja za najpogostejše pripise iz posamezne skupine pripisov.

Tabela 4.3: Povprečna uspešnost modelov za vse uporabljene pripise MeSH.

Model	Povprečje povprečnih točnosti
model s porazdeljeno vrečo besed	0.1391
model s porazdeljenim spominom	
- s povprečjem kontekstnih vektorjev	0.1415
- s stikom kontekstnih vektorjev	0.0556
- z vsoto kontekstnih vektorjev	0.0933
model konvolucijska nevronska mreža	**0.4415**

[1] `https://www.nlm.nih.gov/bsd/indexing/training/CHK_010.html`

.

Tabela 4.4: Povprečna uspešnost modelov za pripise MeSH iz posameznih skupin pripisov.

Skupina pripisov MeSH	DBOW	DMm	DMc	DMs	CNN
Anatomy	0.140	0.152	0.052	0.095	**0.464**
Organisms	0.174	0.171	0.068	0.110	**0.571**
Diseases	0.158	0.170	0.066	0.117	**0.540**
Chemicals and Drugs	0.133	0.148	0.049	0.092	**0.488**
Analytical), Diagnostic and Therapeutic Techniques and Equipment	0.137	0.120	0.055	0.085	**0.370**
Psychiatry and Psychology	0.131	0.132	0.060	0.091	**0.397**
Biological Sciences	0.148	0.142	0.061	0.096	**0.385**
Physical Sciences	0.098	0.089	0.039	0.061	**0.280**
Anthropology), Education), Sociology and Social Phenomena	0.109	0.120	0.050	0.078	**0.358**
Technology and Food and Beverages	0.148	0.142	0.049	0.092	**0.431**
Humanities	0.189	0.156	0.065	0.118	**0.400**
Information Science	0.137	0.122	0.062	0.083	**0.331**
Persons	0.182	0.197	0.104	0.124	**0.477**
Health Care	0.104	0.102	0.050	0.067	**0.299**
Publication Characteristics	0.084	0.082	0.029	0.036	**0.542**

DBOW model s porazdeljeno vrečo besed

DMm model s porazdeljenim spomin in povprečjem kontekstnih vektorjev

DMc model s porazdeljenim spomin in stikom kontekstnih vektorjev

DMs model s porazdeljenim spomin in vsoto kontekstnih vektorjev

CNN model konvolucijske nevronske mreže

Tabela 4.5: Uspešnost modelov za posebne pripise MeSH.

Pripis MeSH	DBOW	DMm	DMc	DMs	CNN
Adolescent	0.316	0.326	0.210	0.215	**0.611**
Adult	0.513	0.504	0.376	0.375	**0.784**
Aged	0.467	0.466	0.319	0.318	**0.738**
Aged, 80 And Over	0.223	0.235	0.144	0.146	**0.472**
Animals	0.765	0.758	0.527	0.560	**0.966**
Cats	0.260	0.309	0.206	0.178	**0.864**
Cattle	0.297	0.304	0.101	0.200	**0.773**
Chick Embryo	0.185	0.159	0.065	0.106	**0.728**
Child	0.350	0.390	0.262	0.265	**0.730**
Child, Preschool	0.268	0.299	0.197	0.195	**0.650**
Dogs	0.347	0.414	0.300	0.252	**0.877**
Female	0.722	0.702	0.552	0.566	**0.886**
Guinea Pigs	0.294	0.167	0.087	0.085	**0.816**
History, 19Th Century	0.248	0.166	0.039	0.127	**0.457**
History, 20Th Century	0.263	0.179	0.056	0.142	**0.437**
History, 21St Century	0.041	0.035	0.013	0.026	**0.089**
Humans	0.934	0.917	0.799	0.817	**0.989**
Infant	0.246	0.277	0.171	0.185	**0.608**
Infant, Newborn	0.370	0.381	0.240	0.291	**0.665**
Male	0.697	0.679	0.556	0.562	**0.880**
Mice	0.483	0.458	0.302	0.300	**0.887**
Middle Aged	0.559	0.544	0.398	0.393	**0.809**
Pregnancy	0.580	0.612	0.369	0.523	**0.846**
Rabbits	0.253	0.216	0.217	0.114	**0.786**
Rats	0.511	0.457	0.282	0.298	**0.929**
Young Adult	0.142	0.156	0.086	0.095	**0.327**

[DBOW] model s porazdeljeno vrečo besed

[DMm] model s porazdeljenim spomin in povprečjem kontekstnih vektorjev

[DMc] model s porazdeljenim spomin in stikom kontekstnih vektorjev

[DMs] model s porazdeljenim spomin in vsoto kontekstnih vektorjev

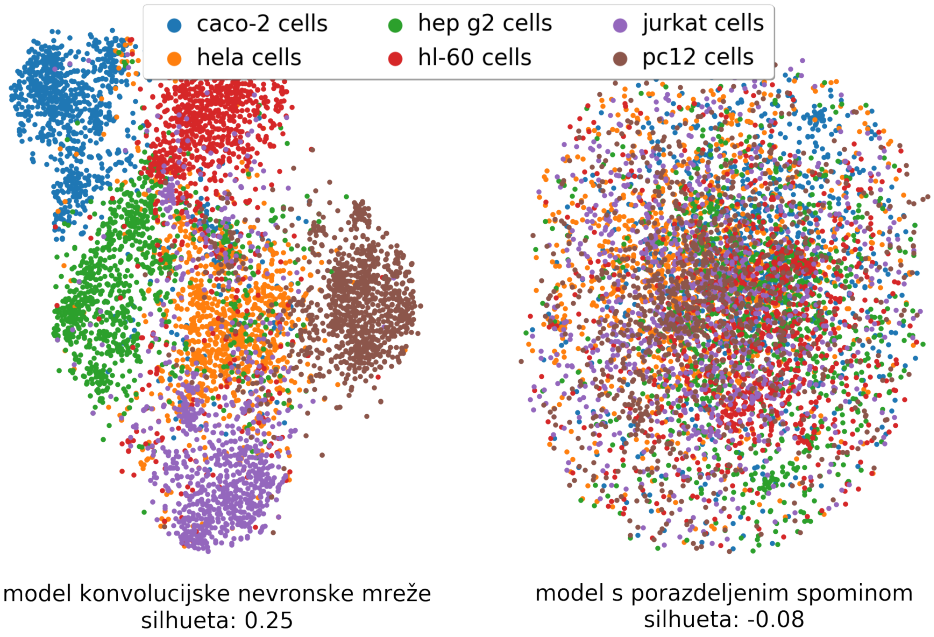
[CNN] model konvolucijske nevronske mreže

4.3 Analiza kvalitete vektorskih predstavitev

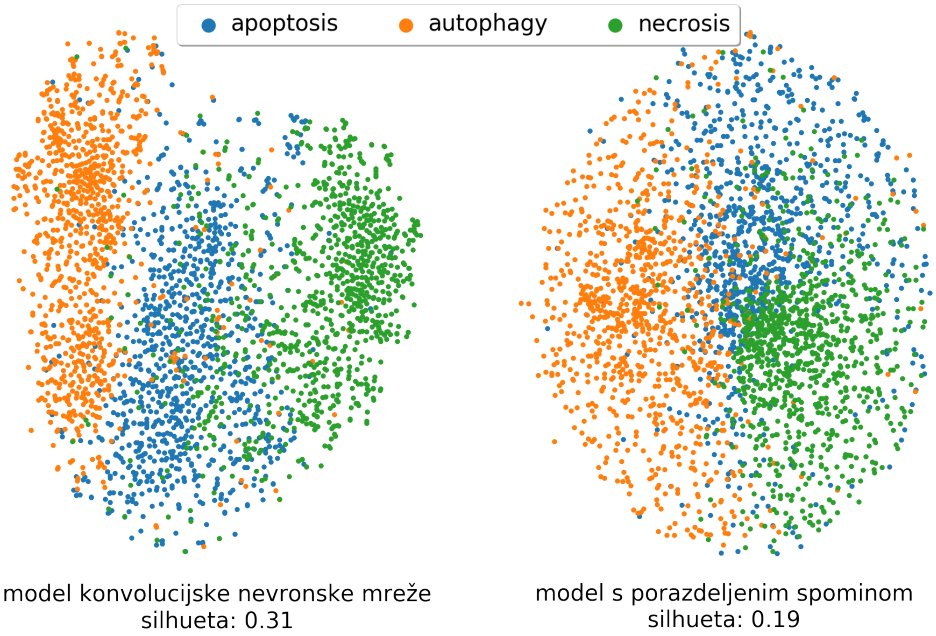
V tem razdelku prikažemo, kako dobro vektorske predstavitve modela konvolucijske nevronske mreže in modelov doc2vec ločijo povzetke člankov z različnimi pripisi MeSH. Ločevanje prikažemo v dvodimenzionalnih projekcijah t-SNE [22], ki ohranijo najbližje sosede iz izvornega vektorskega prostora. V primeru modelov doc2vec prikažemo samo projekcije, ki so najbolje ločile povzetke člankov. V vseh primerjavah so bile to projekcije, ki so bile ustvarjene na osnovi vektorskih predstavitev modela s porazdeljenim spominom, ki uporablja povprečje kontekstnih vektorjev. Ločevanje smo primerjali na povzetkih člankov, zadevajo:

- različne celične linije (slika 4.3),
- različne celične smrti (slika 4.4),
- različne antigene (slika 4.5),
- različne encimske zaviralce (slika 4.6),
- različne bolezni možganov (slika 4.7),
- različne bolezni pljuč (slika 4.8),
- različne hormone (slika 4.9),
- različne duševne motnje (slika 4.10),
- različne ribonukleinske kisline (slika 4.11),
- in različne rane ter poškodbe (slika 4.12).

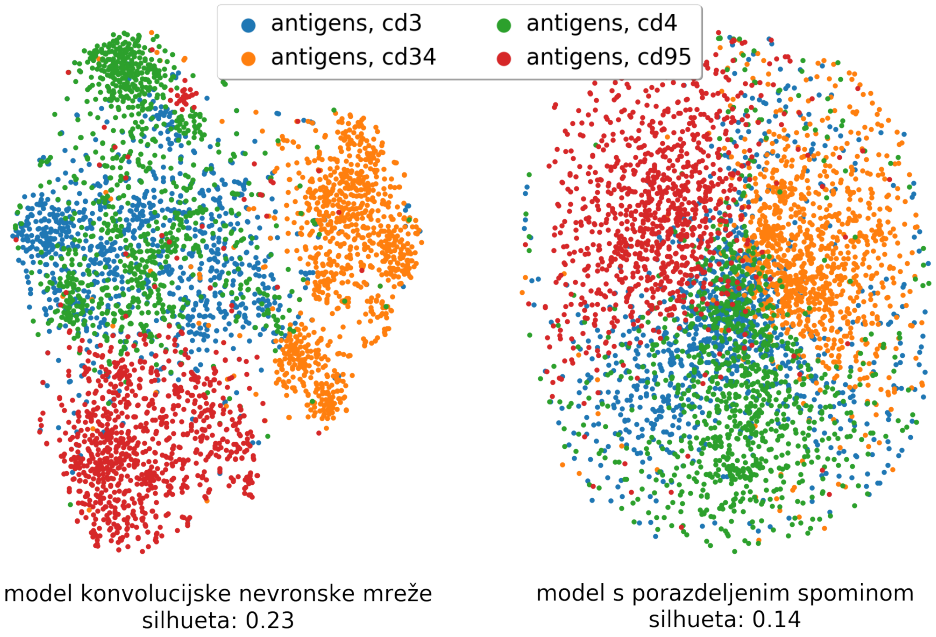
Pod vsako vizualizacijo je navedena silhueta, ki ovrednoti kvaliteto ločevanja. V vseh primerjavah je ločevanje z vektorsko predstavitvijo modela konvolucijske nevronske mreže boljše, kar potrjujejo tudi silhuete, ki so v vseh primerjavah večje.



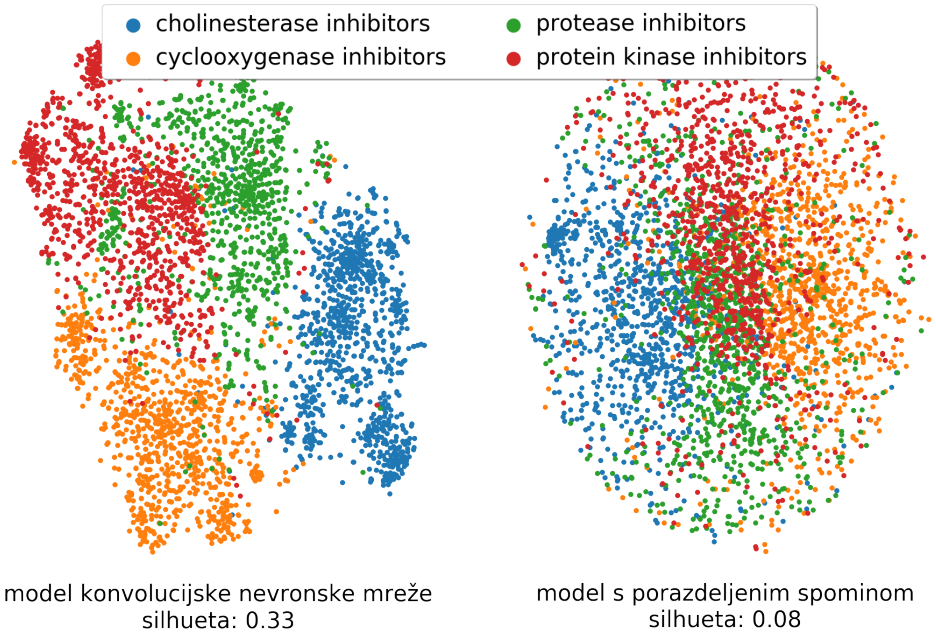
Slika 4.3: Projekcija t-SNE povzetkov člankov o različnih celičnih linijah na osnovi dveh različnih vektorskih predstavitev.



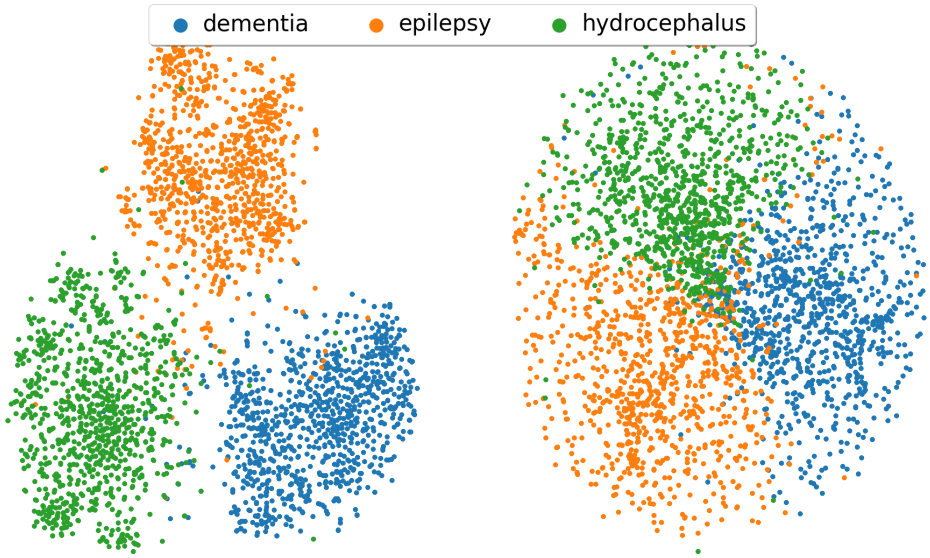
Slika 4.4: Projekcije t-SNE povzetkov člankov o različnih celičnih smrti na osnovi dveh različnih vektorskih predstavitev.



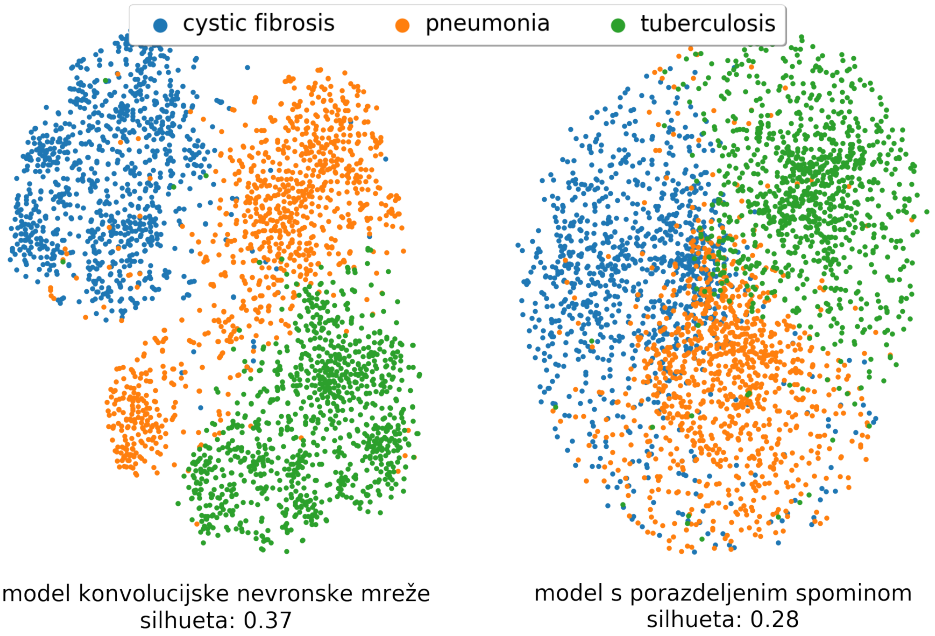
Slika 4.5: Projekcije t-SNE povzetkov člankov o različnih antigenih na osnovi dveh različnih vektorskih predstavitev.



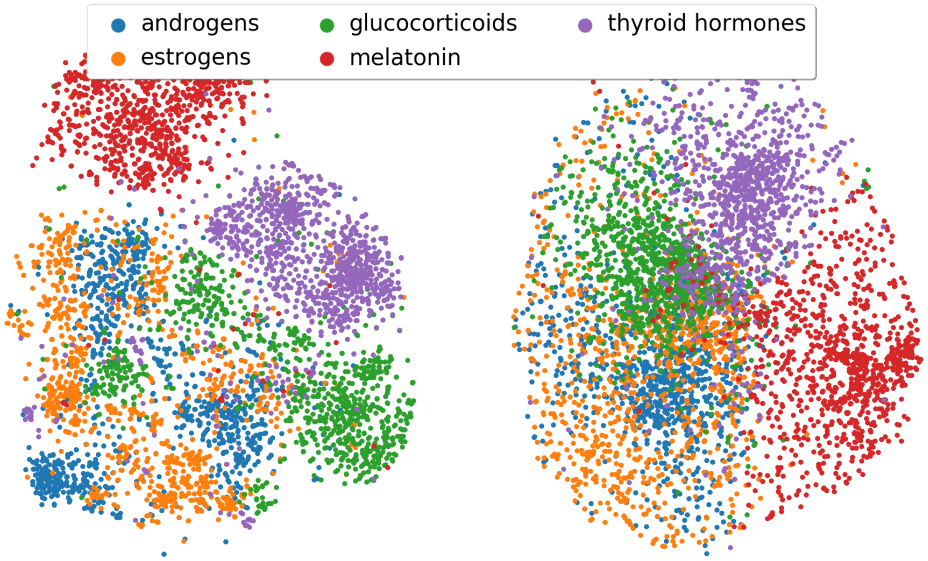
Slika 4.6: Projekcije t-SNE povzetkov člankov o različnih encimskih zaviralcih na osnovi dveh različnih vektorskih predstavitev.



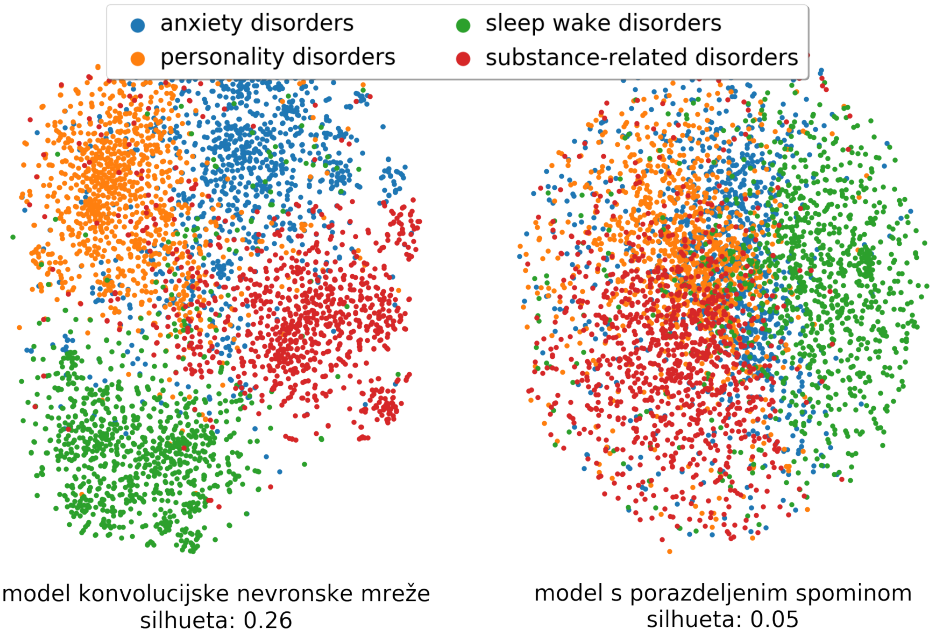
Slika 4.7: Projekcija t-SNE povzetkov člankov o različnih boleznih možganov na osnovi dveh različnih vektorskih predstavitev.



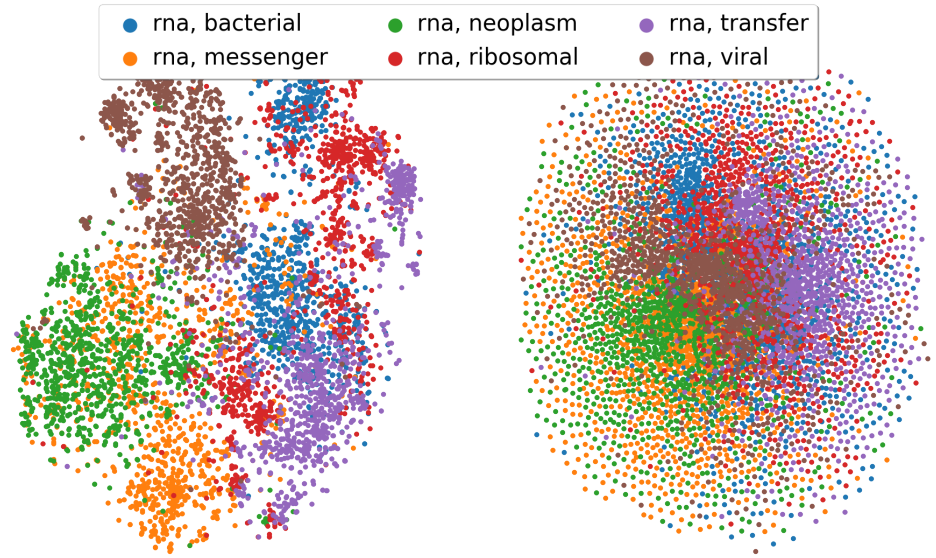
Slika 4.8: Projekcija t-SNE povzetkov člankov o različnih boleznih pljuč na osnovi dveh različnih vektorskih predstavitev.



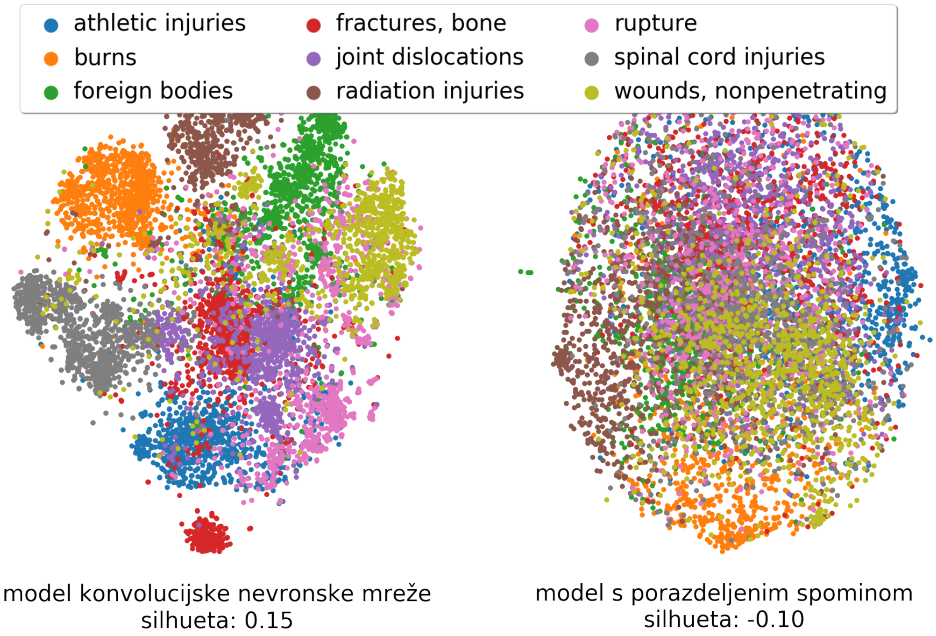
Slika 4.9: Projekcije t-SNE povzetkov člankov o različnih hormonih na osnovi dveh različnih vektorskih predstavitev.



Slika 4.10: Projekcija t-SNE povzetkov člankov o različnih duševnih motnjah na osnovi dveh različnih vektorskih predstavitev.



Slika 4.11: Projekcije t-SNE povzetkov člankov o različnih ribonukleinskih kislinah (RNA) na osnovi dveh različnih vektorskih predstavitev.



Slika 4.12: Projekcija t-SNE povzetkov člankov o različnih ranah in poškodbah na osnovi dveh različnih vektorskih predstavitev.

Poglavje 5

Sklepne ugotovitve

V magistrskem delu smo razvili model konvolucijske nevronske mreže za vektorsko predstavitev besedil s področja ved o življenju in za napovedovanje pripisov MeSH. Rezultati poskusov kažejo, da je vektorska predstavitev modela konvolucijske nevronske mreže v primerjavi z vektorskimi predstavitvami modelov doc2vec uspešnejša v napovedovanju pripisov MeSH in v ločevanju besedil z različnimi pripisi.

Za vektorsko predstavitev besedil bi lahko uporabili tudi modele konvolucijske nevronske mreže, ki se namesto besed učijo iz znakov [29, 6]. Ti modeli dosegajo boljše rezultate, vendar jih je težje optimizirati zaradi kompleksnejših arhitekturih. Lahko bi uporabili tudi globok model s klasično nevronsko mrežo [12], ki je podobna modelu s porazdeljenim spominom, vendar na izhodu namesto besed napove razrede besedil. Model se v primerjavi z modeli konvolucijskih nevronskih mrež hitreje uči in je v nekaterih primerih tudi učinkovitejši.

Literatura

[1] Claire Adam-Bourdarios, Glen Cowan, Cécile Germain, Isabelle Guyon, Balázs Kégl, and David Rousseau. The higgs boson machine learning challenge. In *Proceedings of the NIPS 2014 Workshop on High-energy Physics and Machine Learning*, volume 42, pages 19–55, 2015.

[2] Andrew R. Barron. Statistical properties of artificial neural networks. In *Proceedings of the 28th IEEE Conference on Decision and Control*, pages 280–285, 1989.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[4] François Chollet et al. Keras. `https://github.com/keras-team/keras`, 2015.

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[6] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

[7] Andrew M. Dai, Christopher Olah, and Quoc V. Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[8] Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, Miha Štajdohar, Lan Umek, Lan Žagar, Jure Žbontar, Marinka Žitnik, and Blaž Zupan. Orange: Data mining toolbox in Python. *Journal of Machine Learning Research*, 14:2349–2353, 2013.

[9] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *Advances in Neural Information Processing Systems 13*, pages 472–478, 2001.

[10] Zellig S. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[11] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[12] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[13] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[14] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vec-

tors. In *Advances in Neural Information Processing Systems 28*, pages 3294–3302, 2015.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[18] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

[19] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, volume 32, pages 1188–1196, 2014.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. 2013.

[22] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[23] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

[24] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back propagating errors. 323:533–536, 10 1986.

[25] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[26] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174, 2015.

[27] Antonio Jose Jimeno Yepes, Laura Plaza, Jorge Carrillo-de Albornoz, James G. Mork, and Alan R. Aronson. Feature engineering for medline citation categorization with mesh. *BMC Bioinformatics*, 16(1):113, 2015.

[28] Wenpeng Yin and Hinrich Schütze. Multichannel variable-size convolution for sentence classification. *arXiv preprint arXiv:1603.04513*, 2016.

[29] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, pages 649–657, 2015.

[30] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

[31] Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2, 2004.

Dodatek A

Podrobni rezultati napovedovanja pripisov MeSH

Za vsako skupino pripisov MeSH iz tabele 2.1 prilagamo povprečne točnosti modelov za najpogostejše pripise iz posamezne skupine. Od različic modela s porazdeljenim spominom prilagamo samo rezultate najboljše različice, ki uporablja povprečje kontekstnih vektorjev. V rezultatih smo model s porazdeljenim spomin označili kot DMm, model s porazdeljeno vreča besed kot DBOW in model konvolucijske nevronske mreže kot CNN.

Tabela A.1: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Humanities.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Quality Of Life	24060	0.312	0.354	**0.695**
History, 20Th Century	10767	0.263	0.179	**0.437**
History, 19Th Century	5465	0.248	0.166	**0.457**
History, 21St Century	3072	0.041	0.035	**0.089**
Ethics, Medical	2628	0.083	0.048	**0.322**

Tabela A.2: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Anatomy.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Cells, Cultured	87497	0.243	0.251	**0.493**
Cell Line	67073	0.175	0.168	**0.392**
Brain	62719	0.209	0.232	**0.500**
Liver	52759	0.294	0.333	**0.653**
Cell Line, Tumor	41202	0.353	0.304	**0.573**
Neurons	39709	0.279	0.294	**0.564**
Kidney	30924	0.217	0.247	**0.487**
Tumor Cells, Cultured	30714	0.160	0.154	**0.393**
Lung	26243	0.208	0.271	**0.520**
Cell Membrane	25618	0.134	0.156	**0.337**

Tabela A.3: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Organisms.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Humans	1911332	0.934	0.917	**0.989**
Animals	885550	0.765	0.758	**0.966**
Rats	229068	0.511	0.457	**0.929**
Mice	220522	0.483	0.458	**0.887**
Rats, Sprague-Dawley	53179	0.219	0.189	**0.585**
Cattle	45322	0.297	0.304	**0.773**
Mice, Inbred C57Bl	43241	0.228	0.216	**0.543**
Rats, Wistar	42227	0.164	0.152	**0.521**
Rabbits	40485	0.253	0.216	**0.786**
Escherichia Coli	38382	0.295	0.278	**0.707**

Tabela A.4: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Diseases.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Disease Models, Animal	46820	0.150	0.187	**0.383**
Postoperative Complications	42432	0.192	0.178	**0.348**
Neoplasms	36961	0.099	0.132	**0.601**
Breast Neoplasms	35296	0.382	0.439	**0.898**
Chronic Disease	30157	0.076	0.087	**0.399**
Hypertension	26305	0.349	0.405	**0.765**
Body Weight	25779	0.154	0.153	**0.316**
Lung Neoplasms	25670	0.252	0.285	**0.814**
Acute Disease	24715	0.084	0.089	**0.380**
Recurrence	23890	0.101	0.121	**0.346**

Tabela A.5: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Chemicals and Drugs.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Rna, Messenger	64603	0.269	0.285	**0.599**
Dna	41800	0.157	0.165	**0.470**
Calcium	39009	0.347	0.386	**0.703**
Recombinant Proteins	37157	0.129	0.130	**0.414**
Anti-Bacterial Agents	35614	0.318	0.306	**0.563**
Antineoplastic Agents	34832	0.224	0.211	**0.437**
Biomarkers	34800	0.166	0.200	**0.370**
Antibodies, Monoclonal	30265	0.256	0.240	**0.632**
Bacterial Proteins	28067	0.255	0.214	**0.500**
Transcription Factors	27325	0.210	0.174	**0.463**

Tabela A.6: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Analytical), Diagnostic and Therapeutic Techniques and Equipment.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Treatment Outcome	131585	0.288	0.273	**0.422**
Retrospective Studies	112052	0.325	0.288	**0.703**
Risk Factors	109942	0.277	0.281	**0.501**
Follow-Up Studies	92767	0.200	0.195	**0.352**
Prospective Studies	80495	0.197	0.166	**0.642**
Surveys And Questionnaires	65081	0.293	0.278	**0.514**
Prognosis	62727	0.270	0.258	**0.474**
Reproducibility Of Results	59860	0.215	0.191	**0.349**
Sensitivity And Specificity	56584	0.205	0.201	**0.354**
Magnetic Resonance Imaging	54435	0.392	0.365	**0.737**

Tabela A.7: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Psychiatry and Psychology.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Smoking	19445	0.323	0.428	**0.667**
Pain	15979	0.152	0.204	**0.425**
Reaction Time	15884	0.199	0.194	**0.404**
Motor Activity	15381	0.149	0.137	**0.387**
Stress, Psychological	14782	0.145	0.221	**0.490**
Behavior, Animal	14777	0.166	0.183	**0.320**
Neuropsychological Tests	14664	0.287	0.239	**0.542**
Mental Disorders	14607	0.158	0.145	**0.484**
Health Knowledge, Attitudes, Practice	14368	0.179	0.180	**0.415**
Depression	13335	0.158	0.191	**0.458**

Tabela A.8: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Biological Sciences.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Time Factors	166082	0.121	0.143	**0.170**
Pregnancy	89480	0.580	0.612	**0.846**
Amino Acid Sequence	82792	0.414	0.338	**0.613**
Base Sequence	78955	0.374	0.295	**0.537**
Kinetics	74987	0.259	0.232	**0.411**
Dose-Response Relationship, Drug	65358	0.142	0.141	**0.243**
Mutation	61849	0.259	0.265	**0.486**
Signal Transduction	60022	0.313	0.266	**0.526**
Sensitivity And Specificity	56584	0.205	0.201	**0.354**
Protein Binding	41226	0.187	0.179	**0.316**

Tabela A.9: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Physical Sciences.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Immunohistochemistry	50751	0.249	0.224	**0.410**
Electrophysiology	11282	0.105	0.098	**0.159**
Research Design	10758	0.076	0.073	**0.184**
Computational Biology	9631	0.138	0.118	**0.247**
Outcome Assessment (Health Care)	9271	0.044	0.049	**0.117**
Mathematics	8948	0.045	0.044	**0.140**
Statistics As Topic	8893	0.013	0.014	**0.032**
Drug Design	7544	0.112	0.082	**0.289**
Evidence-Based Medicine	7523	0.058	0.057	**0.256**
Histocytochemistry	7492	0.076	0.071	**0.225**

Tabela A.10: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Anthropology), Education), Sociology and Social Phenomena.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Quality Of Life	24060	0.312	0.354	**0.695**
Socioeconomic Factors	19673	0.176	0.139	**0.318**
Exercise	12840	0.214	0.253	**0.493**
Health Status	11083	0.100	0.102	**0.275**
Age Distribution	10561	0.059	0.063	**0.122**
Social Support	9614	0.115	0.111	**0.455**
Patient Education As Topic	9250	0.070	0.087	**0.281**
Clinical Competence	9222	0.162	0.171	**0.346**
Activities Of Daily Living	9165	0.158	0.161	**0.370**
Sex Distribution	8762	0.052	0.052	**0.109**

Tabela A.11: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Technology and Food and Beverages.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Polymers	11945	0.167	0.138	**0.419**
Biocompatible Materials	8904	0.192	0.157	**0.355**
Nanoparticles	8004	0.219	0.239	**0.572**
Polyethylene Glycols	7938	0.131	0.133	**0.586**
Dietary Supplements	6950	0.171	0.204	**0.417**
Liposomes	6769	0.227	0.300	**0.728**
Quality Control	6511	0.051	0.057	**0.205**
Milk	6441	0.239	0.274	**0.680**
Dietary Fats	6309	0.169	0.159	**0.447**
Animal Feed	6005	0.342	0.297	**0.498**

Tabela A.12: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Information Science.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Molecular Sequence Data	122899	0.437	0.356	**0.598**
Amino Acid Sequence	82792	0.414	0.338	**0.613**
Base Sequence	78955	0.374	0.295	**0.537**
Surveys And Questionnaires	65081	0.293	0.278	**0.514**
Prevalence	40316	0.199	0.191	**0.498**
Algorithms	38568	0.316	0.272	**0.510**
Incidence	35215	0.147	0.141	**0.367**
Severity Of Illness Index	35013	0.101	0.106	**0.206**
Computer Simulation	30830	0.208	0.190	**0.369**
Phylogeny	29987	0.454	0.360	**0.690**

Tabela A.13: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Persons.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Adult	650107	0.513	0.504	**0.784**
Middle Aged	575125	0.559	0.544	**0.809**
Aged	403756	0.467	0.466	**0.738**
Adolescent	256854	0.316	0.326	**0.611**
Child	183420	0.350	0.390	**0.730**
Aged, 80 And Over	131569	0.223	0.235	**0.472**
Child, Preschool	109873	0.268	0.299	**0.650**
Young Adult	101411	0.142	0.156	**0.327**
Infant	84343	0.246	0.277	**0.608**
Infant, Newborn	64556	0.370	0.381	**0.665**

Tabela A.14: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Health Care.

Pripis MeSH	Št. primerov	DBOW	DMm	CNN
Treatment Outcome	131585	0.288	0.273	**0.422**
Retrospective Studies	112052	0.325	0.288	**0.703**
Risk Factors	109942	0.277	0.281	**0.501**
Follow-Up Studies	92767	0.200	0.195	**0.352**
Prospective Studies	80495	0.197	0.166	**0.642**
Surveys And Questionnaires	65081	0.293	0.278	**0.514**
Reproducibility Of Results	59860	0.215	0.191	**0.349**
Sensitivity And Specificity	56584	0.205	0.201	**0.354**
Age Factors	55759	0.109	0.124	**0.219**
Cross-Sectional Studies	42302	0.244	0.183	**0.640**

Tabela A.15: Povprečne točnosti modelov za najpogostejše pripise MeSH iz skupine Geographic Locations.

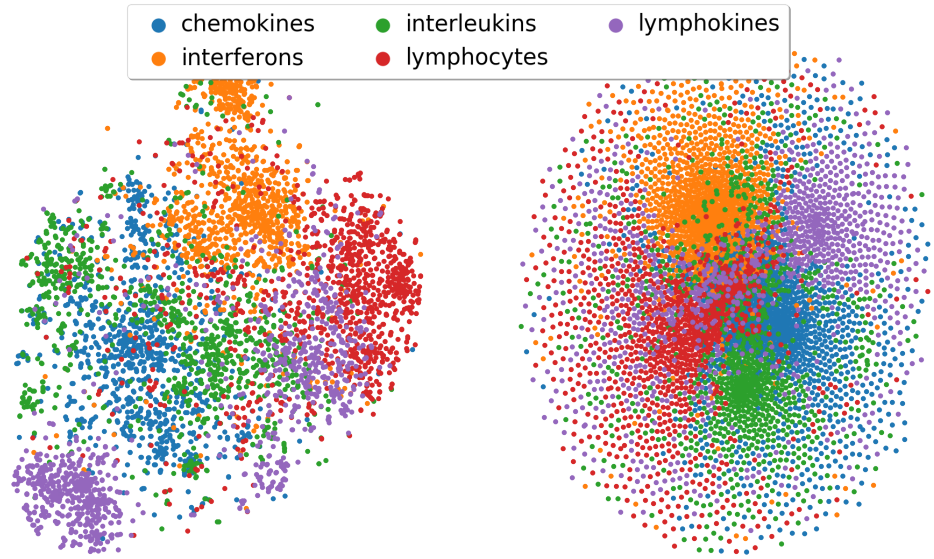
Pripis MeSH	Št. primerov	DBOW	DMm	CNN
United States	64275	0.263	0.229	**0.547**
China	19076	0.257	0.233	**0.721**
United Kingdom	15023	0.116	0.116	**0.449**
Japan	14400	0.192	0.201	**0.596**
Germany	11479	0.133	0.117	**0.476**
Brazil	10832	0.252	0.160	**0.740**
India	10404	0.147	0.150	**0.705**
Italy	10097	0.121	0.088	**0.534**
Europe	9991	0.095	0.098	**0.390**
Australia	9990	0.120	0.116	**0.504**

Dodatek B

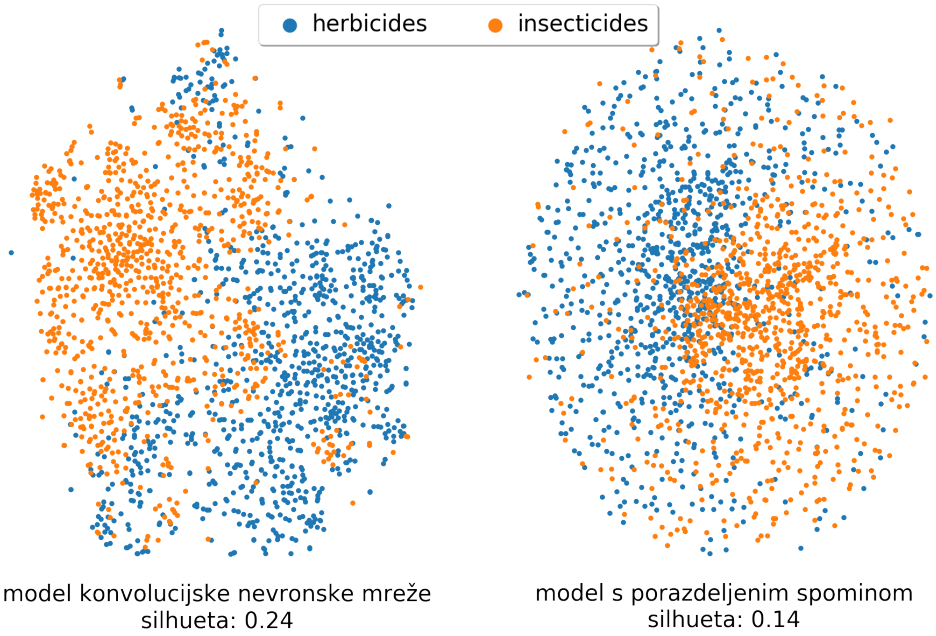
Dodatne projekcije t-SNE

Prilagamo dodatne projekcije t-SNE povzetkov člankov, ki so zgrajene na osnovi vektorskih predstavitev modela konvolucijske nevronske mreže in modela s porazdeljenim spominom, ki uporablja povprečje kontekstnih vektorjev. Projekcije prikazujejo ločevanje povzetkov, ki zadevajo:

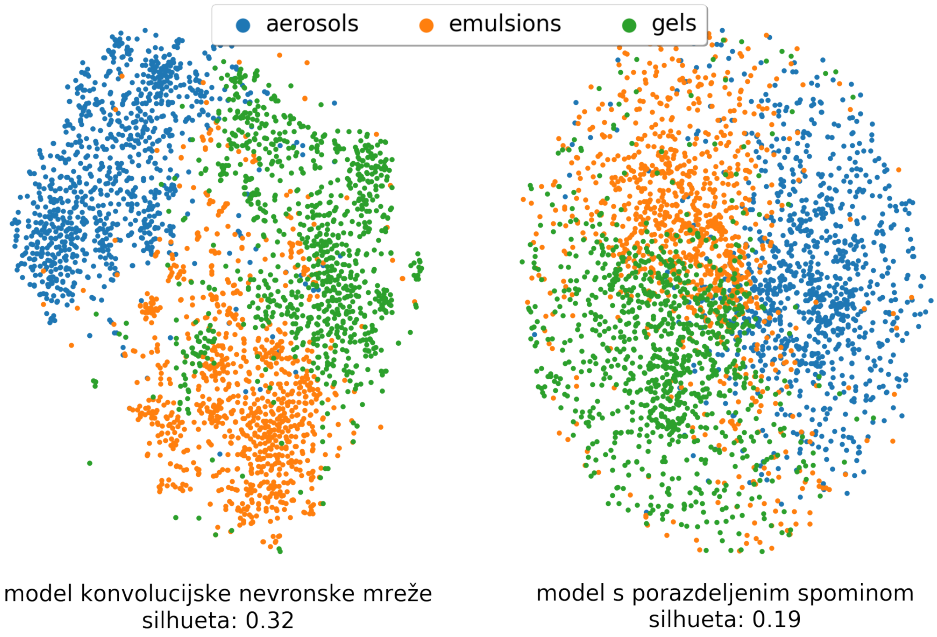
- različne citokine (slika B.1),
- različne pesticide (slika B.2),
- različne koloide (slika B.3),
- različne endoskopije (slika B.4),
- in različne okoljske onesnaževalce (slika B.5).



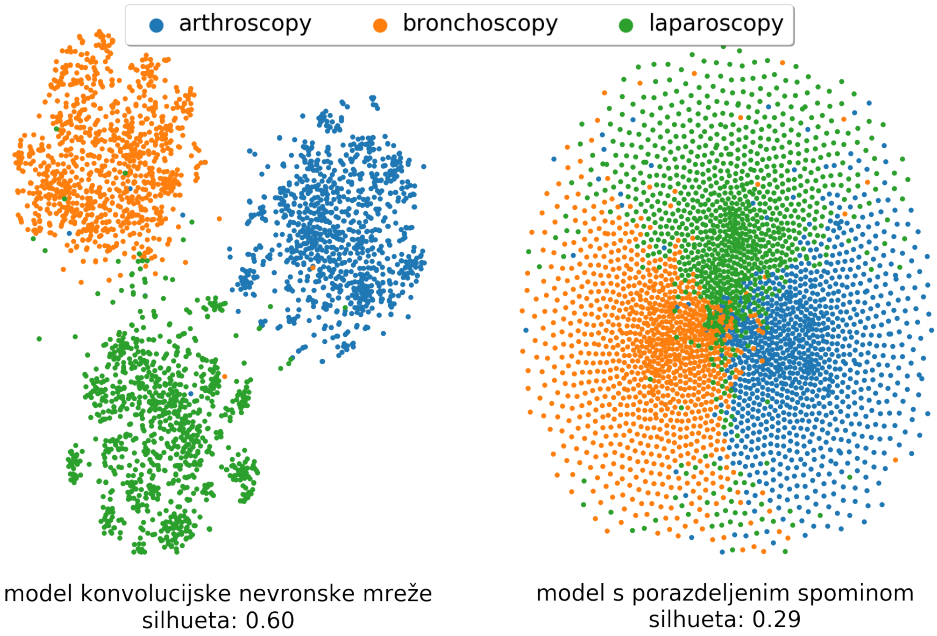
Slika B.1: Projekcije t-SNE povzetkov člankov o različnih citokinih na osnovi dveh različnih vektorskih predstavitev.



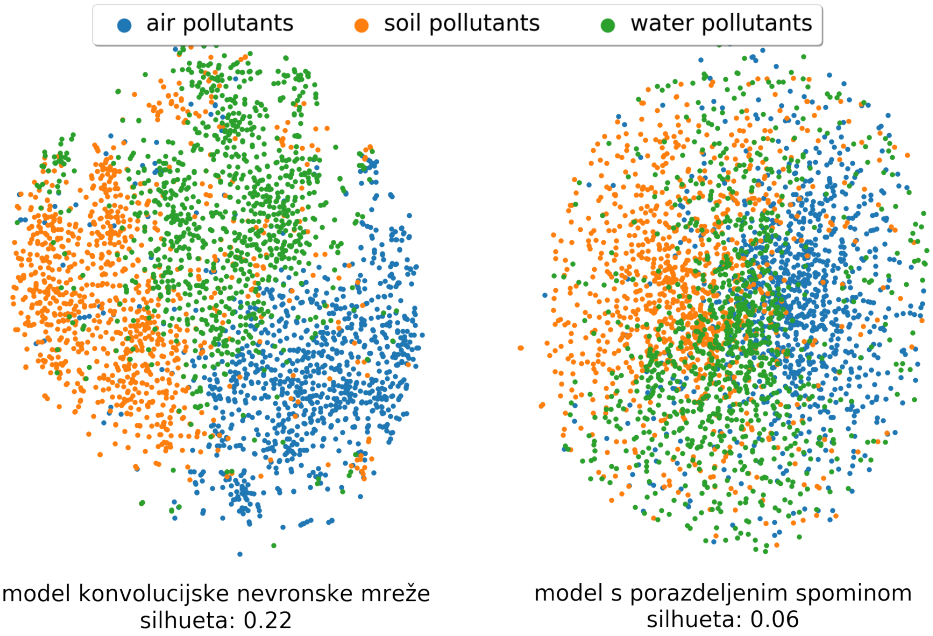
Slika B.2: Projekcija t-SNE povzetkov člankov o različnih pesticidih na osnovi dveh različnih vektorskih predstavitev.



Slika B.3: Projekcije t-SNE povzetkov člankov o različnih koloidih na osnovi dveh različnih vektorskih predstavitev.



Slika B.4: Projekcija t-SNE povzetkov člankov o različnih endoskopijah na osnovi dveh različnih vektorskih predstavitev.



Slika B.5: Projekcija t-SNE povzetkov člankov o različnih okoljskih onesnaževalcih na osnovi dveh različnih vektorskih predstavitev.

Dodatek C

Implementacije modelov

Prilagamo implementaciji večrazredne logistične regresije in konvolucijske nevronske mreže, ki sta napisani v programskem jeziku Python in s programsko knjižnico Keras [4].

C.1 Večrazredna logistična regresija

Implementacija večrazredne logistične regresije za napovedovanje pripisov MeSH iz besedil v vektorskih predstavitvah modelov doc2vec.

```
from keras.models import Model
from keras.layers import Input, Dense
from keras.constraints import max_norm
from keras.optimizers import Adam

input_layer = Input(shape=(1000,))
output_layer = Dense(2890, activation='sigmoid',
                     kernel_constraint=max_norm(3))(input_layer)

model = Model(inputs=input_layer, outputs=output_layer)
model.compile(loss='binary_crossentropy', optimizer=Adam())
```

C.2 Konvolucijska nevronska mreža

Implementacija konvolucijske nevronske mreže za vektorsko predstavitev besedil in napovedovanje pripisov MeSH.

```
from keras.models import Model
from keras.layers import Input, Dense
from keras.layers.embeddings import Embedding
from keras.layers.convolutional import Conv1D
from keras.layers.pooling import GlobalMaxPooling1D
from keras.optimizers import Adam

input_layer = Input(shape=(500,))
embed_layer = Embedding(415253, 300)(input_layer)
conv_layer = Conv1D(1000, 4, activation='sigmoid')(embed_layer)
pool_layer = GlobalMaxPooling1D()(conv_layer)
text_vector = Dense(2890, activation='sigmoid')(pool_layer)

model = Model(inputs=input_layer, outputs=text_vector)
model.compile(loss='binary_crossentropy', optimizer=Adam())
```

Dodatek D

Uporaba modela v programskem okolju Orange3

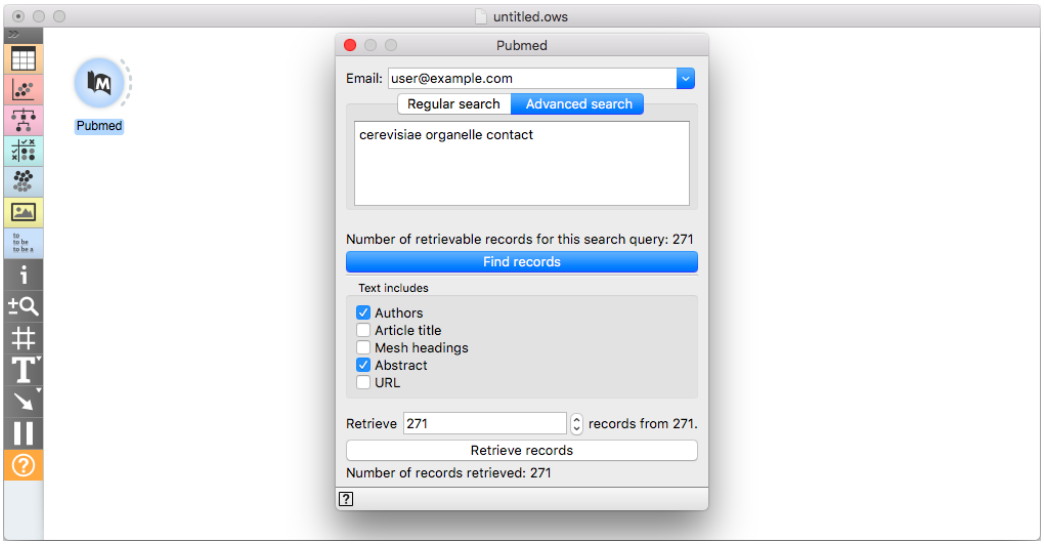
Za enostavno vektorsko predstavitev besedil z modelom konvolucijske nevronske mreže smo razvili komponento za programsko okolje Orange[1] z imenom Text Embedding, ki je prosto dostopna v repozitoriju GitHub[2]. V poglavju predstavimo dva primera uporabe komponente. V prvem primeru predstavimo gručenje besedil, v drugem pa postopek za vizualizacijo besedil. V obeh primerih uporabimo povzetke znanstvenih in strokovnih člankov s področja ved o življenju, ki jih pridobimo s komponento PubMed iz razširitve Orange3 Text[3].

D.1 Gručenje vektorjev besedil

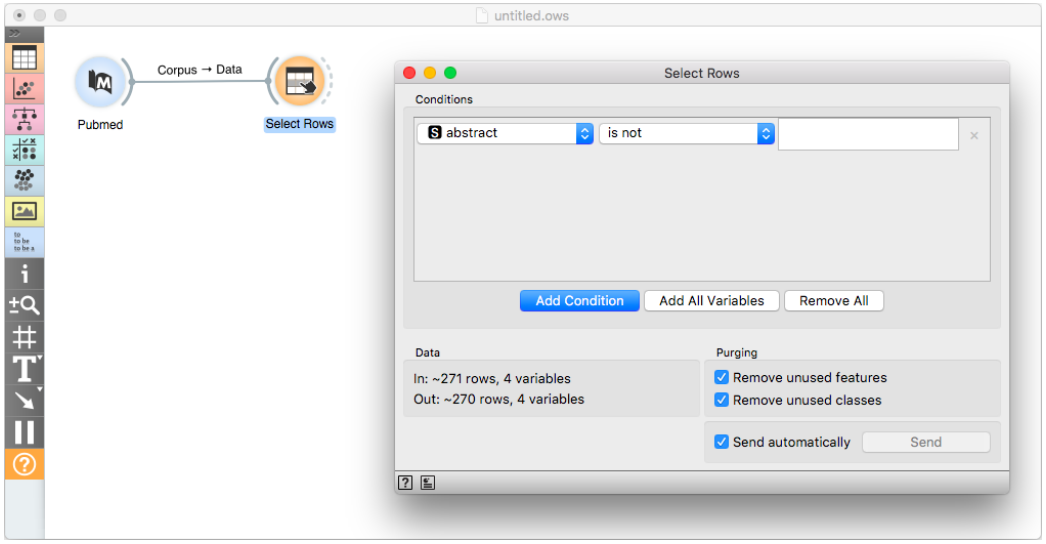
V programskem okolju Orange izberemo komponento PubMed in jo nastavimo tako, kot je prikazano na sliki D.1. V nastavitvenem oknu komponente vnesemo iskalni niz in nato s pritiskom na gumb *Find records* pridobimo članke, ki ustrezajo iskalnemu nizu. Zatem v istem oknu označimo, da od člankov želimo pridobiti imena avtorjev in povzetke. Da pridobimo želene

[1] https://github.com/biolab/orange3
[2] https://github.com/tomislijepcevic/orange3-textembedding
[3] https://github.com/biolab/orange3-text

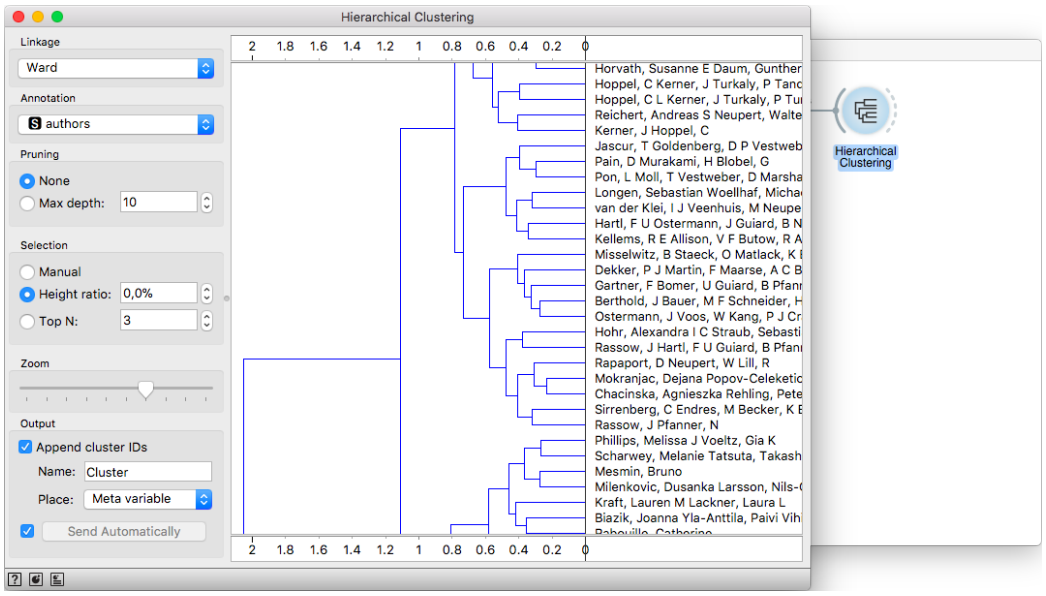
podatke je potrebno pritisniti še na gumb *Retrieve records*. Pridobljeni članki nimajo nujno povzetek, zato s komponento Select Rows odstranimo vse brez povzetkov (slika D.2). Povzetke člankov nato vektorsko predstavimo z novo komponento Text Embedding. Vektorje povzetkov bomo gručili glede na medsebojne kosinusne razdalje, zato s komponento Distances izračunamo medsebojne razdalje in nato s komponento Hierarhical Clustering poženemo hierarhično gručenje vektorjev. Rezultat gručenja je prikazan na sliki D.3.



Slika D.1: Pridobivanje znanstvenih in strokovnih člankov s področja ved o življenju s komponento PubMed.



Slika D.2: Odstranjevanje člankov brez povzetkov s komponento Select Rows.



Slika D.3: Hierarhično gručenje vektorjev povzetkov glede na medsebojne kosinusne razdalje s komponento Hierhical Clustering.

D.2 Vizualizacija vektorjev besedil

V tem primeru ustvarimo projekcijo t-SNE povzetkov člankov na osnovi vektorske predstavitve našega modela. Za povzetke enkrat uporabimo iskalni niz *cerevisiae organelle contact* in drugič *cerevisiae biofuel lignocellulose*. Pridobljene članke združimo v eno podatkovno množico s komponento Concatenate in nato s komponento Select Rows odstranimo članke brez povzetkov, tako kot v prejšnjem primeru. Nato povzetke člankov vektorsko predstavimo z našo komponento Text Embedding. Za izdelavo projekcije t-SNE uporabimo komponento Manifold Learning. Na koncu uporabimo komponento Scatter Plot, ki zgradi razsevni diagram projekcije, kot je prikazan na sliki D.4.

Slika D.4: Projekcije t-SNE povzetkov člankov na osnovi vektorske predstavitve našega modela konvolucijske nevronske mreže.