Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/04TI01Ivanovic

ANALIZA STACK EXCHANGE MREŽE: VIZUALIZACIJA GRAFA POJMOVA I KLASIFIKACIJA TEKSTA PO PROGRAMSKIM JEZICIMA

ANALYSIS OF THE STACK EXCHANGE NETWORK: CONCEPT GRAPH VISUALIZATION AND TEXT CLASSIFICATION BY PROGRAMMING LANGUAGES

Jovan Ivanović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – INFORMACIONI INŽENJERING

Kratak sadržaj – *U radu je prikazana analiza skupa podataka preuzetog sa Stack Exchange, mreže veb stranica posvećenih pitanjima i odgovorima. Rad ispituje osobine ovog skupa prvo kroz implementaciju metode vizualizacije pojmova i tema u skupu podataka, a zatim kroz izgradnju klasifikatora pitanja po programskim jezicima.*

Ključne reči: *Stack Exchange, Sistemi za istraživanje i analizu podataka, Vizualizacija podataka, Klasifikacija teksta*

Abstract – *This paper presents an analysis of the data set retrieved from Stack Exchange, a network of questions and answers (Q&A) websites. The paper investigates the properties of the data set first by implementing a method of visualization of concepts and topics found in the data set, and then by building a classifier to classify questions by programming languages.*

Keywords: *Stack Exchange, Data mining, Data visualization, Text classification*

1. UVOD

Postojanje Interneta i Veba je omogućilo drastično pojednostavljenje razmene i akumulacije znanja. Jedno od značajnih društvenih čvorišta u ovom domenu je *Stack Exchange* mreža veb stranica [1]. Ona je posvećena javnom postavljanju domenskih pitanja, diskusiji o njima i odgovaranju na njih.

Ovaj rad je motivisan javnom dostupnošću znanja sadržanog u ovoj mreži. U mreži se nalazi preko 170 stranica posvećenih posebnim domenima znanja. Najposećenija i najstarija stranica je *Stack Overflow*, posvećena domenu računarskog programiranja [2]. Pored ovog domena, postoje stranice i za razne druge, kao što su matematika (stranica *Mathematics*), kuvanje (stranica *Seasoned Advice*) i video igre (stranica *Arqade*).

Ključna osobina organizacije podataka u ovoj mreži je postojanje tagova koji predstavljaju ključne pojmove za svako pitanje. Postojanje tagova je u ovom radu iskorišćeno na dva načina.

Prvo, iskorišćeno je za formiranje grafa pojmova (na nivou jedne stranice u mreži), gde svaki tag predstavlja jedan pojam, a gde je povezanost pojmova uslovljena njihovim zajedničkim pojavljivanjem u pitanjima. Zatim je vršena vizualizacija delova ovog grafa.

Vizualizacija je zatim proširena grupisanjem pojmova po temama. U ovu svrhu su isprobani LSA (*Latent Semantic Analysis*) [3], NMF (*Non-negative Matrix Factorization*) [4] i LDA (*Latent Dirichlet Allocation*) [5] modeli za modelovanje tema.

Tagovi su takođe uzeti i kao osnova za izgradnju klasifikatora teksta. Mogućnosti klasifikacije su isprobane na primeru klasifikacije tekstova po programskim jezicima na koje se odnose. U ovu svrhu su isprobani naivni Bajesovski klasifikator, linearni SVM (*Support Vector Machine*), nekoliko drugih linearnih modela i neuronska mreža sa *word embeddings* [6] slojem.

2. METODOLOGIJA

Sva programska rešenja implementirana za potrebe rada su izrađena u programskom jeziku *Python*.

Prvo programsko rešenje je alat za preuzimanje podataka sa *Stack Exchange* mreže. Od podataka su korišćena samo pitanja sa tagovima, dok odgovori i komentari na pitanja i odgovore nisu korišćeni. Pitanja su preuzeta sa više stranica u mreži, a najviše (175000 pitanja) je preuzeto sa *Stack Overflow* stranice.

Zatim su nad skupom podataka isprobane mogućnosti vizualizacije grafa pojmova i izgradnje klasifikatora teksta.

2.1. Vizualizacija grafa pojmova

U cilju jednostavnog pregleda pojmova iz određenog domena (tj. sa određene stranice u mreži) realizovana je izgradnja i vizualizacija grafa pojmova. Svaki čvor u grafu predstavlja jedan pojam u domenu, tj. jedan tag iz skupa podataka sa određene stranice. Grane između pojmova su zasnovane na njihovom zajedničkom pojavljivanju u pitanjima, a težina grane je određena brojem zajedničkih pojavljivanja.

Radi preglednosti vizualiacije, bilo je potrebno ograničiti broj grana i broj čvorova. Broj grana je ograničavan brisanjem manje značajnih grana. Svaka grana je pretvarana u dve usmerene grane, gde jedna polazi iz prvog, a jedna iz drugog čvora. Zatim je na nivou svakog čvora u grafu nalažena suma težina izlaznih grana, i

---

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bila dr Jelena Slivka, docent.

zadržavane su samo grane čiji je udeo u toj sumi bio iznad određenog procenta (npr. iznad 1%).

Broj čvorova je zatim ograničavan fokusiranjem samo na određene aspekte grafa jednim od sledećih pristupa:

- prikazivanjem samo odabranih čvorova
- prikazivanjem odabranih čvorova i njima susednih čvorova
- prikazivanjem čvorova koji predstavljaju ključne pojmove za teme izmodelovane nad tagovima
- prikazivanjem čvorova koji predstavljaju ključne pojmove za jednu od tema izmodelovanih nad tagovima

Treći i četvrti pristup selekciji čvorova se oslanjaju na modelovanje tema. Modelovanje tema je, u svrhu proširenja vizualizacije grafa pojmova, vršeno nad tagovima pitanja. Takođe, u svrhu validacije tema nad tagovima pitanja, vršeno je i modelovanje tema nad tekstovima istih pitanja.

U oba slučaja (modelovanje nad tagovima i modelovanje nad tekstovima) tokeni su prvo pretprocesirani TF-IDF (*Term Frequency - Inverse Document Frequency*) statistikom.

Nakon ovoga su isprobane tri tehnike za modelovanje tema: LSA (*Latent Semantic Analysis*), LDA (*Latent Dirichlet Allocation*) i NMF (*Non-negative Matrix Factorization*). Teme dobijene modelovanjem nad tagovima je zatim bilo moguće koristiti pri vizualizaciji grafa pojmova.

2.2. Klasifikacija tekstova po programskim jezicima

Pored vizualizacije grafa pojmova, u ovom radu je ispitana i mogućnost obučavanja klasifikatora nad posmatranim skupom podataka, i to nad primerom klasifikacije tekstova po programskim jezicima.

Tekstovi za obučavanje ovog klasifikatora su preuzeti sa stranice *Stack Overflow*, a kategorisani su po sledećim jezicima: C, C#, C++, *Clojure*, F#, *Go*, *Haskell*, *Java*, *JavaScript*, *Kotlin*, *Objective-C*, PHP, *Python*, R, *Ruby*, *Rust*, *Scala*, *Swift*, *TypeScript* i VB.NET.

Pri obučavanju su korišćena pitanja sa tačno jednim od ovih jezika među svojim tagovima, a klasifikatori su obučavani da prepoznaju tačno jedan jezik po tekstu. Tekstovi pitanja tipično sadrže i prirodni jezik i programski kod.

Isprobani klasifikatori su naivni Bajesovski klasifikator, linearni SVM (*Support Vector Machine*), nekoliko drugih linearnih modela i neuronska mreža.

Kompletan skup podataka je iznosio 105000 pitanja, izbalansiranih po klasama. 20% pitanja je zatim izdvojeno u test skup, a optimizacija hiperparametara je zatim vršena petostrukom unakrsnom validacijom nad preostalim podacima (za naivni Bajesovski klasifikator i linearne klasifikatore) i nad validacionim skupom od 20% (za neuronsku mrežu).

Pri pretprocesiranju, tekst je tokenizovan po sledećem regularnom izrazu:

```
[><=!~:\-
\+\*\/\&\|#$@;]{1,3}|[\[\](){}]{1,2}|[
\"\']|[A-Za-z\#\+\-\_]+
```

Ovaj regularni izraz pored slova pronalazi i određene kombinacije drugih karaktera specifične za programski kod, kao što su `>=`, `++` i `:=`.

Frekventne reči nisu uklanjane kako bi se izbeglo slučajno uklanjanje tokena značajnih za sintaksu nekih od jezika (npr. `if` u mnogim jezicima). Velika slova su zadržana zbog zapažanja da mogu biti od pomoći u razlikovanju sličnih tokena (npr. `If` u jeziku VB.NET naspram `if` u brojnim drugim jezicima). Stematizacija i lematizacija nisu rađene, pod pretpostavkom da nisu značajne za domen programskih jezika.

U slučaju naivnog Bajesovskog klasifikatora i linearnih modela, tokeni su zatim grupisani u unigrame, bigrame i trigrame, i na kraju im je određivana TF-IDF statistika. Za TF-IDF je korišćena L2 regularizacija. Vrste ngrama i regularizacije za TF-IDF su određene optimizacijom hiperparametara za svaki od modela.

Isprobano je nekoliko linearnih modela, uključujući linearni SVM i logističku regresiju, gde se linearni SVM pokazao kao najuspešniji. Za regularizaciju je optimizacijom odabrana *elastic net* [6] regularizacija sa jednakim udelom L1 i L2 regularizacije.

Višeklasna klasifikacija linearnim modelima je implementirana tzv. *one-versus-rest* klasifikacijom, tj. svaka klasa je imala svoj klasifikator, a za konačnu klasu nekog teksta je birana ona čiji klasifikator je bio najpouzdaniji u svoj pozitivan rezultat.

Klasifikacija je isprobana i upotrebom neuronske mreže. Finalna verzija mreže je sadržala sledeće slojeve:

1. *word embeddings* sloj: dimenzionalnost vektora 150)
2. 1D konvolucioni sloj: 128 filtera, veličina konvolucije 5, ReLU (*Rectified Linear Unit*) aktivacija
3. globalni 1D *max pooling* sloj
4. običan neuronski sloj: 20 neurona, *softmax* aktivacija

Pre ulaza u mrežu, formiran je vokabular od 20000 tokena. Dimenzije vokabulara i slojeva su određene eksperimentalno, tako da povećanje van pronađene granice daje samo zanemarljiva poboljšanja u rezultatima mreže.

Konvolucioni sloj je dodat sa namerom da obuhvati određene lokalne šablone tokena u tekstu, i doveo je do manjeg poboljšanja rezultata. Dalje dodavanje konvolucionih slojeva nije poboljšalo rezultate.

Pored navedenih slojeva, isprobana je i upotreba rekurentnih slojeva u vidu LSTM (*Long Short-Term Memory*) [6] i GRU (*Gated Recurrent Unit*) [6] slojeva, kao i dodavanje običnih neuronskih slojeva. Ovi pristupi nisu doveli do poboljšanja rezultata.

Uvođenje regularizacije u vidu *dropout* regularizacije [6] nad *word embeddings* i konvolutivnim slojem i u vidu L2 regularizacije nad *word embeddings* slojem nije dovelo do poboljšanja rezultata nad validacionim skupom.

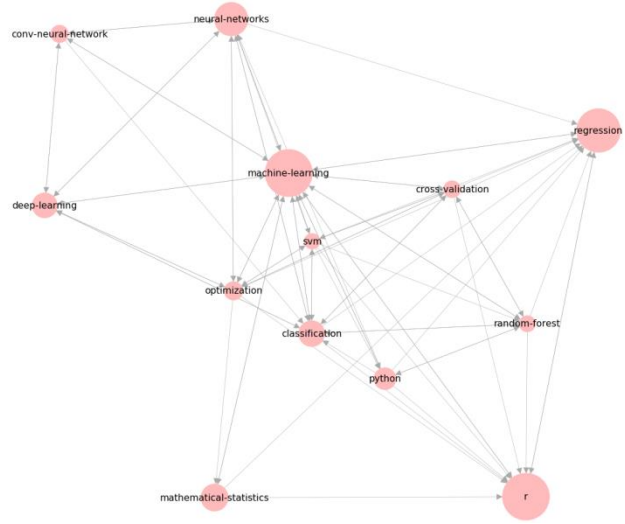
Za funkciju gubitka finalne mreže je korišćena kategorijska unakrsna entropija (eng. *categorical crossentropy*), a za optimizaciju je upotrebljen *Adam* algoritam (*adaptive moment estimation*) [6].

3. EVALUACIJA I REZULTATI

Evaluacija će posebno biti izneta za vizualizaciju grafa pojmova, za modelovanje tema i za klasifikaciju teksta.

3.1. Vizualizacija grafa pojmova

Na slici 2 je prikazan primer vizualizacije grafa pojmova za okolinu pojma (taga) *machine-learning* nad podacima sa stranice *Cross Validated* (posvećene statistici, istraživanju podataka i mašinskom učenju). Prikazane su samo grane sa udelom od bar 1.5% u ukupnoj sumi težina izlaznih grana po čvorovima. Prikazani pojmovi, kao što su klasifikacija i neuronske mreže, imaju jasne veze sa konceptom mašinskog učenja.



Slika 2. *Vizualizacija okoline pojma machine-learning za podatke sa stranice Cross Validated*

3.2. Modelovanje tema

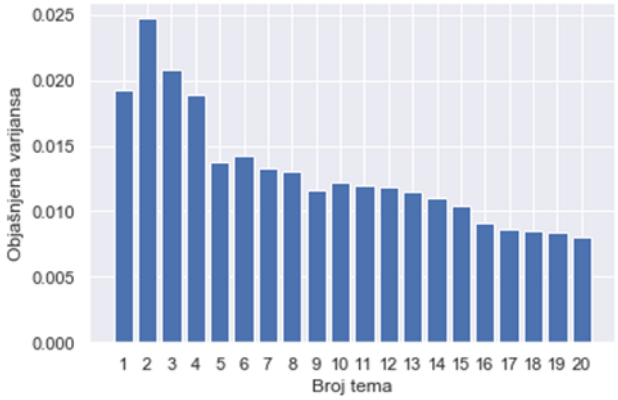
Rezultati modelovanja tema su ocenjeni tumačenjem adekvatnosti ključnih reči po temama za različite domene, i procenjeno je da je NMF davao najpreciznije rezultate. U tabeli 1 se nalazi primer rezultata NMF za *Mathematics* skup podataka.

Tabela 1. *Teme pronađene upotrebom NMF nad podacima sa stranice Mathematics*

Tema	Ključne reči
1	*calculus, integration, limits, definite-integrals, derivatives*
2	*linear-algebra, matrices, eigenvalues-eigenvectors, vector-spaces, linear-transformations*
3	*probability, statistics, probability-theory, probability-distributions, random-variables*
4	*real-analysis, sequences-and-series, general-topology, analysis, proof-verification*
5	*geometry, trigonometry, euclidean-geometry, triangle, circle*
6	*combinatorics, number-theory, elementary-number-theory, discrete-mathematics, permutations*

Od dostupnih metrika ovih modela, kao najkorisnija pri izboru tema se pokazala mera objašnjene varijanse za LSA. Uzevši u obzir da je NMF davao najbolje rezultate, a ne LSA, razmotrena je opravdanost upotrebe mere objašnjene varijanse LSA kao heuristike za izbor tema i za NMF. Ovo je potencijalno opravdano činjenicom da NMF koristi za inicijalizaciju isti algoritam na kojem je LSA zasnovana, a to je SVD (*Singular Value Decomposition*). Ova sličnost je verovatan razlog za generalno isti redosled najznačajnijih pojmova po temama koje su NMF i LSA pronalazile (na *Mathematics* primeru ova dva pristupa dele najznačajniji pojam za svaku od tema).

Na grafiku 1 je prikazana mera objašnjene varijanse za LSA na primeru podataka sa stranice *Mathematics*. Prve četiri teme se izdvajaju po značaju.



Grafik 1. *Mera objašnjene varijanse po temi LSA modela nad podacima sa stranice Mathematics*

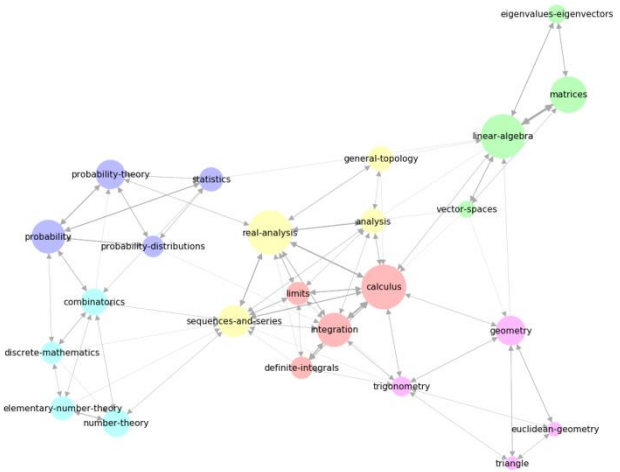
Teme pronađene nad tagovima su takođe upoređene i sa temama pronađenim nad tekstovima pitanja. U tabeli 2 su prikazani rezultati upotrebe NMF nad tagovima i tekstovima sa stranice *Computer Science*. Primećuju se iste teme (ali sa različitim redosledom), uključujući grafove, Tjuringove mašine i teoriju kompleksnosti.

Tabela 2. *Teme pronađene upotrebom NMF nad tagovima i tekstovima sa stranice Computer Science*

Tema	Ključne reči: tagovi	Ključne reči: tekstovi
1	*algorithms, optimization, algorithm-analysis*	*log, array, data*
2	*complexity-theory, time-complexity, np-complete*	*turing, machine, tm*
3	*formal-languages, finite-automata, regular-languages*	*np, complete, polynomial*
4	*graphs, graph-theory, shortest-path*	*edges, edge, tree*
5	*turing-machines, computability, undecidability*	*regular, grammar, context*

Konačan rezultat modelovanja tema nad tagovima jeste integracija sa vizualizacijom grafa pojmova. Na slici 3 je prikazan rezultat modelovanja 6 tema nad podacima sa stranice *Mathematics*.

Prikazana su po četiri najznačajnija pojma po temi, a teme su prikazane kroz boje čvorova. Način grupisanja čvorova sa istim temama dodatno potvrđuje validnost pronađenih tema, a bliskost određenih tema na vizualizaciji pruža dodatnu perspektivu koja nije bila dostupna prostim posmatranjem ključnih pojmova.



Slika 3. *Primer vizualizacije tema pronađenih u podacima sa stranice Mathematics*

3.3. Klasifikacija tekstova po programskim jezicima

Od isprobanih klasifikatora će biti razmatrani naivni Bajesovski klasifikator, linearni SVM (kao najuspešniji linearni klasifikator) i neuronska mreža sa *word embeddings* slojem.

Naivni Bajesovski klasifikator se sa F1 merom od 86% nije pokazao kao najuspešniji model, ali se zato pokazao kao koristan za razmatranje značajnih n-grama po jezicima (npr. C među značajnim tokenima sadrži "`printf`", Go sadrži "`:=`", dok Haskell sadrži "`->`"). Elementi sintakse jezika su se pokazali kao najznačajniji tokeni.

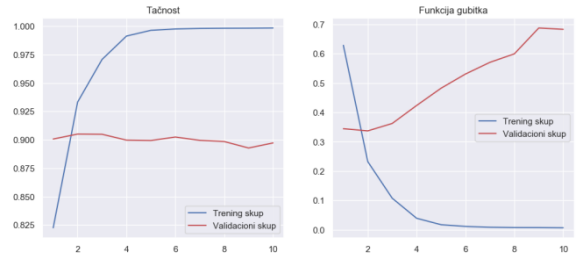
I SVM i neuronska mreža su dali F1 meru od 91%, što ih čini najuspešnijim modelima. U tabeli 3 su prikazane F1 mere po jezicima za SVM. Rezultati za neuronsku mrežu su veoma slični, i neće biti posebno prikazani.

Tabela 3. *Rezultati klasifikacije konačnog SVM klasifikatora nad test skupom*

Jezik	F1	Jezik	F1	Jezik	F1
C	88%	Java	87%	Ruby	94%
C#	87%	JavaScript	90%	Rust	97%
C++	85%	Kotlin	94%	Scala	95%
Clojure	96%	Objective-C	84%	Swift	86%
F#	96%	PHP	91%	TypeScript	92%
Go	96%	Python	91%	VB.NET	91%
Haskell	95%	R	95%		

Razmatranje uzroka zabune ukazuje na jezike koje imaju određene sličnosti, gde se npr. pitanja za jezik *Swift* greškom klasifikuju kao *Objective-C* (oba su jezici kompanije Apple), ili se C++ pitanja klasifikuju kao C pitanja (C je značajan uzor za C++).

Neuronska mreža je dala generalno veoma slične rezultate kao i SVM. Za neuronsku mrežu je upadljivo da je već posle dve epohe obučavanja dolazila do najboljeg rezultata nad validacionim skupom (grafik 2). Dodavanje regularizacije je uspelo da smanji nagli rast funkcije gubitka nad validacionim skupom, ali nije uspelo da popravi najbolji rezultat.



Grafik 2. *Tačnost i vrednost funkcije gubitka po epohama obučavanja neuronske mreže*

4. ZAKLJUČAK

U radu su prikazani rezultati analize podataka sa *Stack Exchange* mreže veb stranica. Fokus analize je bio dvostruk, prvo na vizualizaciji grafa pojmova, a zatim na klasifikaciji teksta.

Vizualizacija grafa pojmova se pokazala kao koristan način za orijentisanje unutar određenog domena, npr. kroz predlaganje sličnih pojmova na osnovu pojmova već poznatih korisniku, ili kroz davanje opšteg pregleda određenog domena. Zajedno sa vizualizacijom pojmova je isprobano i modelovanje tema nad pojmovima, a kao najuspešniji od isprobanih modela se pokazao NMF.

Klasifikacija teksta je isprobana na primeru klasifikacije pitanja sa stranice *Stack Overflow* po programskim jezicima na koje se odnose. Najuspešniji modeli su bili linearni SVM i neuronska mreža sa *word embeddings* slojem, oba sa prosečnom F1 merom od 91% duž 20 razmatranih jezika. Ovi klasifikatori bi mogli npr. poslužiti za tagovanje programskog koda ili pitanja na stranici *Stack Overflow*. Dalji smer za razvoj klasifikatora bi mogao biti pronalaženje više tagova po tekstu.

Pored navedenog, ovaj rad demonstrira i značaj postojanja slobodno dostupnih skupova podataka kao što je ovaj.

5. LITERATURA

[1] https://stackexchange.com (pristupljeno u januaru 2019.)

[2] https://stackoverflow.com (pristupljeno u januaru 2019.)

[3] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze (2008), Introduction to Information Retrieval, Cambridge University Press, chapter 18: Matrix decompositions & latent semantic indexing

[4] Lee, Daniel D., and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." *Advances in neural information processing systems*. 2001.

[5] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.

[6] Goodfellow, Ian, et al. *Deep learning*. Vol. 1. Cambridge: MIT press, 2016.

Kratka biografija:

Jovan Ivanović rođen je u Novom Sadu 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Informacioni inženjeringa odbranio je 2019. god.
kontakt: jovan.ivanovic3@gmail.com