Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/06BE29Hrnjakovic

ANALIZA I OBRADA TEKSTA POMOĆU RAZLIČITIH MODELA TEMA

TOPIC MODELS FOR TEXT ANALYSIS

Olivera Hrnjaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Ovaj rad opisuje trenutne mogućnosti i ograničenja postojećih algoritama za izdvajanju tema iz teksta. Dat je teorijski prikaz popularnih modela tema uz sve neophodne korake analize i obrade teksta koji se izvršavaju pre slanja podataka na ulaz modela. Praktičan deo rada je izdvajanje tema iz pitanja sa sajta Stack overflow. Upoređeni su LSA, PLSA i LDA pristup, a evaluacija modela je izvršena određivanjem koherentnosti tema odgovarajućim merama, imenovanjem tema i analizom njihove vizuelizacije u prostoru. Kako modeli tema unapred zahtevaju navođenje broja tema koje će biti izdvojene iz teksta, deo rada posećen je optimizaciji hiperparametara. Izabrani model za modelovanje tema jeste LDA sa 6 tema. Da bi se dobila numerička procena performansi modela 30 pitanja je ručno označeno imenima dobijenih tema i simuliran je klasifikacioni model. Ova pitanja su korišćena kao test skup podataka u kreiranom LDA klasifikacionom modelu. Postignuta je uspešnost od 77% tačnosti.*

Ključne reči: *modelovanje tema, analiza teksta, LDA*

Abstract – *This paper describes the current capabilities and limitations of existing topic modeling algorithms. A theoretical overview of popular topic models was given, along with all the necessary analysis and text processing steps that should be performed on the input data. The practical part of the paper is to extract topics from questions from the Stack overflow site. LSA, PLSA and LDA approaches were used and evaluated using coherence, perplexity, naming techniques and topic visualization in space. To get the best performance for topic modeling, we estimated the best topic number. The results showed that the best model is LDA and the best topic number is 6. In order to obtain a numerical evaluation of the model performance, 30 questions were manually annotated with the names of the topics acquired. In that way, we simulated a classification model. These questions were used as a test data set in the created LDA classification model. The accuracy of the classification model was 77%.*

Keywords: *Topic modeling, text analysis, LDA*

1. UVOD

Ekspanzija podataka i nemogućnost njegove obrade na tradicionalan način dovela je do porasta količine neobrađenih podataka čime se potencijalno gube informacije. Kao rešenje ovog problema pojavile su se tehnike veštačke inteligencije koje mogu da rade sa podacima brže od čoveka. Nestruktuirani podaci u vidu teksta predstavljaju većinu neobrađenih podataka. Oblast veštačke inteligencije koja radi sa analizom teksta na osnovu jezika naziva se obrada prirodnog jezika. Modelovanje tema je deo obrade prirodnog jezika i jedan je način za određivanje značenja teksta. Poznavanje semantike teksta pruža mogućnosti za upoređivanje tekstova i dalju primenu u sistemima za preporuku, komunikaciju, prevođenje i slično. U ovom radu biće opisane trenutne mogućnosti postojećih algoritama za izdvajanje tema iz teksta.

U narednoj sekciji izložen je teorijski prikaz obrade prirodnog jezika i modelovanja tema. Treća sekcija sadrži opis problema koji je obrađivan u radu i pregled postojeće literature. Četvrta sekcija sadrži metodologiju predloženog rešenja. U petoj sekciji objašnjena je eksperimentalna evaluacija rezultata, dok poslednja sekcija zaključuje rad i predlaže pravce potencijalnog daljeg istraživanja.

2. OBRADA PRIRODNOG JEZIKA I MODELOVANJE TEMA

Računari su davno prevazišli čoveka u mnogim veštinama poput matematičkog računa i memorisanja podataka. U nekim oblastima dosegli su jednakost sa čovekom, ali i dalje je izazov napraviti sistem koji bi mogao da nadmaši čoveka u rešavanju problema koji iziskuju kreativnost i reagovanje na do sada neviđene probleme na način svojstven čoveku.

Primer ovakvog sistema jeste sistem koji radi sa jezikom. Jezik je "živa" materija, menja se zajedno sa ljudima, deli se u dijalekte, vremenom se pojavljuju nove reči, izrazi i idiomi, a postojeći se gube. Ovim problemima bavi se obrada prirodnog jezika.

Modeli tema (engl. *Topic models*) su klasa probabilističkih latentnih promenljivih modela za tekstualne podatke koji služe da predstave tekstualne podatke u obliku raspodele tema [1]. Ovi modeli kreiraju neku vrstu sumarizacije ili filtriranja teksta u formatu koji se može interpretirati kao tema (engl. topic). Modeli tema spadaju u nenadgledani oblik učenja i predstavljaju neku vrstu klasterovanja "sličnih" reči. Dakle, formiraju se apstraktne teme koje u stvari predstavljaju grupu povezanih reči.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac, van. prof.

Postoje dve vrste modela tema:

- metode zasnovane na sadržaju teksta i
- prediktivne metode.

Razlika između ovih metoda jeste u tome što metode koje su bazirane na sadržaju teksta rade na osnovu statistike i prebrojavanja. Sračunava se koliko često se neka reč pojavljuje u kombinaciji sa susednom rečju u velikom tekstualnom korpusu, a potom se taj broj mapira na vektor koji predstavlja svaku reč.

Prediktivni modeli direktno predviđaju reč na osnovu njoj susednih reči koristeći naučene vektore za ugradnju. U ovom radu izloženi su algoritmi koji analiziraju sadržaj teksta. Izloženi su LSA, PLSA i LDA koji su trenutno aktuelni algoritmi u oblasti modelovanja tema. Pored njih opisan je i *Deep learning* pristup koji predstavlja osnovu za budući razvoj ovih algoritama.

3. OPIS PROBLEMA

Stack overflow je sajt na kome ljudi koji se bave programiranjem mogu da postavljaju pitanja i daju odgovore. Sajt posećuje veliki broj korisnika, što znači da se na njemu svakodnevno postavlja veliki broj pitanja i daje veliki broj odgovora. Svako pitanje ima naslov u kome skraćeno treba objasniti njegovu suštinu i sam tekst pitanja koji sadrži detaljno objašnjenje pitanja, često uz primere i delove koda. Cilj sajta jeste da na što brži način omogući korisnicima da reše problem. Zato je neophodno da pitanja budu semantički povezana i grupisana sa sebi međusobno sličnim pitanjima.

Postojanje sistema koji obrađuje tekstove u naslovu i tekstu komentara i na osnovu toga povezuje slična pitanja veoma je važno za ovakav sajt. Analiziranje sadržaja pitanja je značajno, jer će opisi sličnih problema u vidu značenja tekstova biti slični. U ovom radu biće prezentovan sistem koji izvlači teme iz teksta i na taj način predstavlja njegovo značenje. Ovakva reprezentacija teksta mogla bi poslužiti kao deo sistema za preporuku ili deo sistema koji eliminiše duplikate tj. upućuje korisnike na slična pitanja.

3.1. Opis skupa podataka

Korišćen je skup podataka *Python Questions from Stack Overflow* [2] preuzet sa sajta *Kaggle*. Ukupan broj pitanja u skupu podataka je 607282, a za potrebe ovog projekta iskorišćen je deo skupa podataka od 100000 pitanja. Tekstovi pitanja pisani su na engleskom jeziku. Skup podataka obuhvata 3 fajla, fajl sa pitanjima, sa odgovorima i sa tagovima. Za potrebe projekta korišćen je fajl sa pitanjima, fajl sa tagovima je eksperimentalno korišćen radi kreiranja potencijalne metrike za evaluaciju, dok fajl sa odgovorima nije korišćen. Korišćeni su atributi: *Title* (Naslov pitanja) i *Body* (Telo odnosno tekst pitanja) koji su spojeni u jedan atribut *Text* koji je dalje korišćen u radu.

3.2. Pregled postojeće literature

Veštačka inteligencija je jedna od najpopularnijih oblasti u kompjuterskim naukama u današnje vreme. Jedan od pokazatelja njene popularnosti je velik broj objavljene stručne literature, naučnih radova i članaka. Modelovanje tema je dostiglo visok nivo, a pristupi koji se koriste se konstantno unapređuju radi poboljšanja performansi.

Tehnike probabilističkog modelovanja tema za formiranje sistema za preporuku naučnih radova iskorišćene su u [3]. Korišćen je skup podataka sa sajta *CiteULik*e koji je sadržao veliki broj citata naučnih radova. Za svaki rad izdvojeni su naslov i abstrakt koji su potom transformisani u odgovarajući oblik da bi bili pogodan ulaz u model. Na osnovu ovoga formirano je obeležje koje je korišćeno u ovom radu kao osnovno obeležje na osnovu koga je vršena analiza. Obrada tekstualnog sadržaja sa sajta *Stack overflow* bila je tema rada [4]. Ideja rada bila je izdvojiti i grupisati članke i diskusije na osnovu tema koje se javljaju u sadržaju.

Za ovo je korišćen LDA kao mehanizam modelovanja tema. Pre nego što je primenjen LDA, urađeno je pretprocesiranje teksta. Najpre je iz teksta odbačeno sve što nije rečenica, odnosno izbačeni su delovi koda. Delove koda je jednostavno izbaciti iz teksta zato što su na sajtu označeni html tagom <code>.

Takođe, svi drugi HTML tagovi koji opisuju izgled sadržaja na sajtu su izbačeni. Nakon toga obavljeno je uklanjanje najčešćih engleskih reči poput “a”, “the” “is”, koje ne utiču na izvlačenje tema iz teksta. Kako je reč o istom skupu podataka, pretprocesiranje u ovom radu je urađeno po ugledu na dati rad.

Analiza teksta i modelovanje tema rađeno je u radu [5] koji se bavi poređenjem LDA i LSA modela kako bi se formirao automatski sistem za preporuku filmova. Ovaj rad korišćen je zbog upoređivanja tehnika modelovanja tema i metode evaluacija modela. Sistem za preporuku zamišljen je da funkcioniše bez bilo kakve informacije o korisniku, odnosno u pitanju je sistem za preporuku baziran na sadržaju filmova. Metod evaluacije bio je eksperimentalna ručna evaluacija: 30 korisnika je ocenilo rad sistema za preporuku. Rezultati su pokazali da je LSA bio uspešniji u davanju preporuka od LDA.

4. METODOLOGIJA

Tekst je najpre pripremljen za ulaz u modele pretprocesiranjem. Algoritmi za modelovanje tema očekuju ulazne tekstualne podatke u tačno određenom formatu. Kao ulaz u većinu modela potrebno je proslediti rečnik i *bag-of-words* zbirku reči. Prvi korak pretprocesiranja bila je tokenizacija. Tokenizacija je proces dobijanja tokena iz reči. Za dobijanje tokena korišćene su funkcije iz biblioteke *gensim*. Nakon što je dobijen token nad njim su izvršeni stemovanje i lematizacija kako bi se dobio koren reči i model ne bi bio zavisan od vrste reči. Nakon stemovanja i lematizacije formirani su rečnik i *bag-of-words* model (tekstualni korpus). Za ove korake u pretprocesiranju takođe postoji podrška u *gensim*-u za engleski jezik.

Iako je često teško odrediti optimalan broj tema kada se formiraju modeli tema, postoji nekoliko metoda koje služe kao smernice u ovom poslu. Čovek najtačnije određuje koliko su dobijene teme povezane sa tekstom od koga smo pošli, ali bi za ovakav posao bilo potrebno angažovanje

velikog broja ljudi, a posao bi bio izuzetno vremenski zahtevan. Zbog toga je gotovo nemoguće odrediti uspešnost modela na ovaj način. Korišćene su *coherence* [6] i *perplexity* [7] mere, čijom kombinacijom je određeno da je 6 optimalan broj tema za LSA i LDA modele. Rezultati dobijeni merom nedoumice (*perplexity*) se vide u tabeli 1. Bolji model ima manju meru nedoumice, a najmanja mera postignuta je za 6 i 7 tema, za TFIDF korpus.

Tabela 1. Vrednost mere nedoumice za LDA model

	5 tema	6 tema	7 tema
Običan korpus	-7.030	-7.025	-7.021
TFIDF korpus	-7.28	-7.32	-7.32

Da bi se donela konačna odluka o optimalnom broju tema nad LDA i LSA modelom ispitana je mera koherencije koja je pokazala da je 6 optimalan broj tema za oba modela. Za PLSA model ne postoji kompletna implementacija zbog čega nije moguće ispitati uspešnost modela u odnosu na ove mere.

Korišćen je običan tekstualni korpus ali i korpus dobijen TFIDF metodom, a na kraju su upoređeni dobijeni rezultati. TFID ili tf–idf je metoda koja se koristi u pretraživanju informacija kako bi se utvrdila važnost reči za dokument u odnosu na neki tekstualni korpus [8].

Upoređena su 3 modela: LSA, PLSA i LDA. Za kreiranje LSA i LDA modela korišćen je paket *models* iz *gensim* biblioteke. Model se kreira tako što se konstruiše *LsiModel* ili *LdaModel* objekat kojima se prosleđuju parametri od kojih je minimalno proslediti tekstualni korpus, rečnik i broj tema.

Kada je reč o kreiranju PLSA modela, koriste se dva pristupa. Prvi pristup koristi latentne promenljive i verovatnoću i algoritam maksimizacije očekivane verodostojnosti - odnosno EM algoritam. Mane ovog pristupa su sporo izvršavanje i teško rukovanje sa modelom kada je reč o obradi novih dokumenata [9]. Drugi pristup koristi faktorizaciju matrice, odnosno NMF algoritam (nenegativnu faktorizaciju matrice).

Ovaj način simulacije PLSA je znatno brži u odnosu na model latentnih promenljivih i jednostavniji je za implementaciju s obzirom da postoji gotova implementacija u biblioteci *sklearn*.

5. ANALIZA REZULTATA

Kao glavni metod evaluacije modela odabrana je ručna analiza tema, a u obzir su uzeti i rezultati dobijeni na osnovu mera nedoumice i koherentnosti [10]. Svaka tema prikazana je preko svojih 10 najfrekventnijih reči. Najpre su definisane metode imenovanja tema na osnovu kojih su teme imenovane [11]. U modelima tema počeli smo od formiranja latentnog sloja kako bi dobili upotrebljiviji model. Međutim u fazi imenovanja tema i evaluacije modela cilj je razbiti u delove latentni sloj kako bi shvatili njegovo značenje. Reči koje se javljaju u pronađenim temama treba objediniti tako da opišu semantiku tih tema. LDA i LSA su posebno pogodni za ovakav pristup pošto su u temama reči navedene zajedno sa svojom frekvencijom ponavljanja čime je predstavljen značaj svake reči u temi. Analizom dobijenih tema metodom PLSA pokazano je da se reči iz tema ponavljaju u više tema i da skup reči ne predstavlja celinu koju je moguće opisati ili imenovati.

Najbolji model na osnovu mera nedoumice i koherencije pokazao se LDA model. Analizom reči u temama za modele LSA i LDA zaključeno je da je za LDA model moguće imenovati 5 tema, a za LSA 4.

Zbog toga je LDA model proglašen za najbolji model za dati skup podataka. Prikaz dobijenih imenovanih tema vidi se na slici 1.

	Stringovi i tekst	Kolekcije	Objektno programiranje	Veb programiranje	Topic 5	Fajl sistem i komandna linija
0	string	list	class	django	datetim	subprocess
1	charact	valu	method	file	pyqt	command
2	encod	function	decor	instal	typeerror	script
3	unicod	string	object	error	slice	stdout
4	regex	file	attribut	script	twitter	popen
5	text	array	instanc	server	excel	file
6	match	dictionari	function	user	player	scrapi
7	lxml	code	inherit	work	oauth	shell
8	express	number	subclass	code	sud	cython
9	html	lik	defin	window	chart	swig

Slika 1. *Imenovane teme za najbolji model (LDA)*

Konačna evaluacija je izvršena ručno, analizom nekoliko nasumično izabranih dokumenata odosno pitanja iz test skupa podataka. Za test skup podataka je izdvojeno 10000 pitanja. Ručno je labelirano 30 dokumenata imenima tema za LDA model. Svaki dokument označen je sa najmanje jednom a najviše šest oznaka. Korišćena je metrika koja je brojala koliko označenih tema je model pogodio. Dakle to znači da su u obzir uzeti samo *true positive*s, odnosno model nije "kažnjavan" za pogrešno izdvojene teme. Rezultati LDA modela pokazali su 30 pogođenih labela od ukupnog 39 označenih, što znači da je model za ovaj uzorak imao uspešnost od 77%.

6. ZAKLJUČAK

U ovom radu su predstavljeni osnovni koncepti modelovanja tema kroz teorijski i praktični prikaz. Definisano je modelovanje tema kroz obradu prirodnog jezika, objašnjene su matematičke osnove na kojima se ova oblast zasniva kao i aktuelno stanje u ovoj oblasti i potencijalni pravci budućeg razvoja. Poseban akcenat stavljen je na transformaciju teksta, eksploratornu analizu i vizuelizaciju podataka, jer su to koraci koji prave razliku između dobrih i loših modela. poređeni su LSA, PLSA i LDA. Korišćeni skup podataka su pitanja sa sajta *Stack overflow*.

Za određivanje performansi modela kombinovano je nekoliko metoda: mere koherentnosti i nedoumice, vizuelizacija tema u prostoru i analiza koherentnosti ručnim imenovanjem tema. Utvrđeno je da su korišćenjem LDA modela dobijene najbolje teme.

Da bi se dobila numerička procena performansi modela 30 pitanja je ručno označeno imenima dobijenih tema i simuliran je klasifikacioni model.

Ova pitanja su korišćena kao test skup podataka u simuliranom LDA klasifikacionom modelu. Postignuta je uspešnost od 77% tačnosti.
Praktično rešenje predstavljeno u ovom radu je polazna tačka u rešavanju opisanog problema. Da bi se ovaj predlog rešenja unapredio najpre bi trebalo za obučavanje modela koristiti veću količinu podataka. Tačnije bilo bi potrebno pronaći optimalnu količinu podataka za dati problem.

Još jedno unapređenje rešenja mogli bi biti dobijeno ako se broj reči ili karaktera u pitanjima ograniči nekim brojem. Ovako nešto moglo bi se uraditi tehnikama za sumarizaciju teksta i time bi se potencijalno izbalansirao skup podataka. Konačno neophodno bi bilo unaprediti metriku za evaluaciju modela kako bi se modeli mogli egzaktnije uporediti. To bi moglo biti postignuto ručnim označavanjem pitanja imenima tema, modifikacijom tagova koji već postoje za pitanja ili kombinacijom ove dve metode.

7. LITERATURA

[1] Topic model. In Wikipedia, The Free Encyclopedia. Retrieved August, 2019, from https://en.wikipedia.org/wiki/Topic_model

[2] Python Questions from Stack Overflow Retrieved from https://www.kaggle.com/stackoverflow/pythonquestions

[3] Wang, Chong, and David M. Blei. "Collaborative topic modeling for recommending scientific articles." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.

[4] Barua, Anton, Stephen W. Thomas, and Ahmed E. Hassan. "What are developers talking about? an analysis of topics and trends in stack overflow." Empirical Software Engineering 19.3 (2014): 619-654.

[5] Bergamaschi, Sonia, Laura Po, and Serena Sorrentino. "Comparing Topic Models for a Movie Recommendation System." WEBIST (2). 2014.

[6] Mimno, David, et al. "Optimizing semantic coherence in topic models." Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, 2011.

[7] Perplexity To Evaluate Topic Models Retrieved from http://qpleple.com/perplexity-to-evaluate-topic-models/

[8] tf–idf. In Wikipedia, The Free Encyclopedia. Retrieved August, 2019, from https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[9] Source code. https://github.com/laserwave/plsa/blob/master/plsa.py

[10] Evaluate Topic Models: Latent Dirichlet Allocation (LDA) Retrieved from https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0

[11] Binkley, David, et al. "Understanding LDA in source code analysis." Proceedings of the 22nd international conference on program comprehension. ACM, 2014.

Kratka biografija:

Olivera Hrnjaković rođena je u Novom Sadu 1995. god. Osnovne studije je upisala 2014. godine na Fakultet tehničkih nauka, odsek Računarstvo i automatika. Osnovne studije je završila 2018. godine, nakon čega upisuje master akademske studije na Fakultetu tehničkih nauka, smer Inteligentni sistemi.