Implementacija algoritma za etiketiranje reči u srpskom jeziku u programskom jeziku Python

Miloš Kosanović, Igor Đurović, Dimitrije Dimitrijević, Slavimir Stošović
Visoka Tehnička Škola Niš
Niš
milos.kosanovic@vtsnis.edu.rs, djurovic.igoor@gmail.com, mitad92@gmail.com,
slavimir.stosovic@vtsnis.edu.rs

Sažetak – **U radu je opisan proces automatskog osnovnog morfološko-sintaksičkog etiketiranja, kao jednog od osnovnih problema obrade prirodnih jezika, korišćenjem algoritma zasnovanog na skrivenim lancima Markova koji je implementiran u programskom jeziku *Python*. Objašnjen je način implementacije algoritma sa svim koracima od pribavljanja digitalnog izvora do dobijanja etiketiranog teksta. Prikazani su rezultati tri eksperimenta koji su sprovedeni za srpski i engleski jezik korišćenjem *INTERA* i *MULTEXT-East* anotiranih korpusa reči. Objašnjen je način na koji je izvršen eksperiment, a zatim su rezultati eksperimenta prikazani i obrazloženi.**

Ključne reči: etiketiranje reči, token, korpus reči, NLTK

I. UVOD – OBRADA PRIRODNIH JEZIKA

Problemi koji se javljaju prilikom obrade prirodnog jezika (*NLP, Natural Language Processing*), kao jedne od oblasti računske lingvistike (*Computational Linguistics*), podrazumevaju najčešće veoma složeni postupak koji čine pre svega: leksička analiza, morfološko-sintaksička analiza, semantička analiza i ponekad analiza pragmatike. Leksička analiza podrazumeva segmentaciju govora ili teksta, gde se najpre određuje početak i kraj rečenica i reči, kao i osnovne leksičke kategorije: brojevi, interpunkcija, reči, HTML etikete i sl. Rezultat leksičke analize je skup tokena. Morfološko-sintaksička analiza proučava strukturu reči, rečenica i teksta kao i njihovu međusobnu zavisnost. Proces vršenja ove analize naziva se još i parsiranje, a kao rezultat dobija se tzv. stablo parsiranja koje pokazuje sintaktičke veze između reči u rečenici. Semantička analiza se dalje bavi značenjem i smislom ovih reči, kao i njihovim međusobnim odnosima. Jedan od primera bio bi razlikovanje homonima (*Word Sense Disambiguation* – WSD), kao i razlikovanje hipernima, sinonima i dr. Analiza pragmatike bavi se načinom izgovora i naglasaka nad određenom reči, čime se menja suštinsko značenje reči ili rečenice. Jedan od značajnih koraka u složenom procesu obrade prirodnih jezika predstavlja i etiketiranje reči. [1]

Etiketiranje govora ili teksta (*POS tagging*) jedan je od zadataka kojim se bavi prirodna obrada jezika u kome se reči anotiraju odgovarajućim gramatičkim kategorijama kao što su imenica, glagol, zamenica, pridev, prilog, predlog, rečca, usklik ili uzvik i dr. Često se naziva još i morfološko ili morfološko-sintaksičko etiketiranje (*MSD tagging*), naročito ukoliko se pored osnovnih gramatičkih kategorija anotiraju i dodatne potkategorije kao što su pol, broj, padež ili vremenski oblik. Proces etiketiranja dobro je poznat zadatak i važan korak u procesu obrade prirodnih jezika. Često prethodi procesu lematizacije, koji predstavlja svođenje reči na njihov osnovni oblik ili leksemu.

Postoji više pristupa za implementaciju PoS programa i oni se najčešće dele na: algoritme zasnovane na pravilima (*rule-based tagger*) [2], statističke algoritme zasnovane na verovatnoći (*HMM tagger* i *maximum entropy tagger*) [2][3], algoritme zasnovane na transformacijama (*Transformation based Learning – TBL*) [4] i algoritme za učenje stablom odlučivanja (*tree tagger*) [1]. U ovom radu koristili smo statistički algoritam koji određuje etiketu reči pomoću skrivenih lanaca Markova.

II. PRETHODNO POSTIGNUTI REZULTATI

Zadatak etiketiranja rečenica na engleskom jeziku generalno se smatra zatvorenim pitanjem. U radu Kristofera Meninga [5] odlično se sumira trenutno stanje kao i preostali problemi koje treba rešiti, te da li ih je uopšte moguće rešiti kako bi se preciznost sa trenutnih 97% popela na ciljanih 100%. Etiketiranje reči nije potpuno deterministički određeno pravilima i u nekim slučajevima se ni stručnjaci iz oblasti lingvistike ne slažu prilikom određivanja etikete reči. Takođe, visoko inflektivni jezici i jezici u kojima redosled reči u rečenici nije strogo definisan, poput srpskog jezika, naročito su problematični. U literaturi se pominje da prosečna ljudska greška prilikom etiketiranja iznosi i do 3–5 procenata.

Većina radova na srpskom jeziku predstavlja ostvarene rezultate etiketiranja reči na nedovoljno transparentan način. Korpusi reči i tekstovi koji su korišćeni za treniranje i testiranje modela nisu navedeni ili često nisu javno dostupni. Objavljeni rezultati uglavnom ne sadrže objašnjenje o načinu računanja i evaluacije algoritama, pa je jako teško rezultate uporediti ili replicirati eksperimente. Standardne mere i načine za ocenu kvaliteta modela i algoritama kao što su tačnost, preciznost, F1 mera i tabela pogrešno dodeljenih etiketa (*confusion matrix*) nedostaju gotovo svim radovima koje su autori ovog rada razmatrali. Rad koji najadekvatnije opisuje trenutno stanje u ovoj oblasti je rad [6], u kom se pominje da je postignuta preciznost etiketiranja za srpski jezik 96.46%, ali nije navedena preciznost etiketiranja nepoznatih reči. U radu [1] prijavljeni su rezultati od 93% i 60% za poznate i nepoznate reči kod etiketiranja korišćenjem TnT (*Trigrams'n'Tag*) algoritma i 93% i 44.45% kod lematizacije gde je korišćen MBT - *Memory-Based Taggger* – TiMBL algoritam.

III. N-GRAM MODEL

Termin ***N-gram*** označava sekvencu od *n* jedinica nečega, u tekstu ili u govoru. U zavisnosti od toga za šta se primenjuju, te jedinice mogu biti slogovi, foneme, reči ili

slova. Broj jedinica nečega što čini *N-gram* naziva se dužina *N-grama*. Dužina prestavlja jako bitan faktor i često se pominje, zato se *N-grami* različitih dužina obično drugačije i nazivaju. Tako se sekvenca dužine 1 naziva unigram (eng. *unigram*), dužine 2 bigram (eng. *bigram*) a dužine 3 trigram (eng. *trigram*). *N-grami* većih dužina nazivaju se četvorogramima (dužine 4) i petogramima (dužine 5) i oni se dosta ređe koriste. Primeri *N-grama* slova, različitih dužina, reči „abrakadabra“ su:

- unigrami: a, b, r, a, k, a, d, a, b, r, a
- bigrami: ab, br, ra, ak, ka, ad, da, ab, br, ra
- trigrami: abr, bra, rak, aka, kad, ada, dab, abr, bra
- četvorogrami: abra, brak, raka, akad, kada, adab, dabr, abra
- petogrami: abrak, braka, rakad, akada, kadab, adabr, dabra

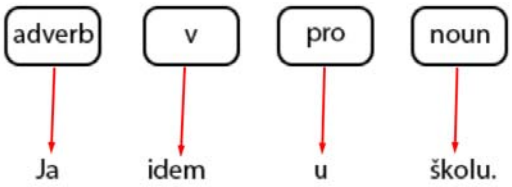
Za slučaj kada se koriste jedinice *N-grama* koje su slova postoji još nekoliko stvari koje treba definisati. Treba razmatrati da li kao slova treba uzeti i blanko znake na početku i kraju svake reči, jer se time u sam *N-gram* model uvodi i informacija o tome koja se slova često nalaze na počecima i krajevima reči nekog jezika. [7]

N-grami se koriste da bi se napravili ***N-gram* modeli** koji modeluju, na određeni način, statistička svojstva pojave-*grama* u nekom tekstu, napisanom ili izgovorenom. *N-gram* model se sastoji od vrednosti *N-grama* i nekih brojnih vrednosti koje se pridružuju svakoj vrednosti *N-grama*. Jedan primer takve brojne vrednosti je broj pojavljivanja N-*grama* reči u tekstu.

IV. Skriveni lanci Markova

Korišćenje verovatnoće u problemu etiketiranja reči prilično je staro i primenjuje se još od 1965. godine. U nastavku je navedeno kako funkcioniše algoritam koji koristi skrivene lance Markova (*Hidden Markov Model* ili HMM) koji predstavlja specijalni slučaj Bajesove klasifikacije (*Bayesian classification).*

Skriveni lanac Markova (u daljem tekstu HMM) deli događaje, odnosno stanja i odgovarajuće promenljive, u skrivena X stanja i posmatrana Y stanja. Na primer, skriveni događaj može biti fonema koja je izgovorena u govornom signalu ili neki od njegovih parametara, ili leksema i etiketa reči u rečenici tekstuelnog ulaza (odnosno etiketa), a posmatrani događaj je onda ili etiketa (za PoS), ili prepoznata reč u rečenici (za WSD). Niz posmatranih stanja zavisi samo od jednog trenutnog stanja u nizu skrivenih stanja, u smislu uslovnih verovatnoća, a svako skriveno stanje zavisi samo od trenutnog i prethodnih u nizu. Svakoj etiketi ili reči u rečniku može biti posvećen poseban lanac sa skrivenim stanjima koja odgovaraju prethodnim rečima ili etiketama, a mogu se i paralelno obrađivati. Jezički model može biti zadat grafovima prelaska stanja u vremenu, slično grafovima kontekstno slobodnih gramatika, čiji su lukovi, tzv. bigrami, obeleženi verovatnoćom. U slučaju da prethodna dva stanja utiču na izbor novog, računaju se trigrami, itd. Uopšte, *N-grami* kao podnizovi niza leksema koje čine ulazni tekst sa svojim frekvencijama u korpusu, važan su alat za dobijanje verovatnoća i efikasno obučavanje HMM-a.



Slika 1. Izdvajanje bigrama

Kod bilo kog problema klasifikacije prisutan je polazni skup uzoraka, a zadatak je da se ustanovi kojoj klasi ili grupi pripada svaki od tih uzoraka. U našem slučaju skup uzoraka je sekvenca uzastopnih reči u rečenici kojima treba dodeliti odgovarajuće etikete. HMM-i računaju verovatnoću distribucije etiketa na sekvenci reči u rečenici i biraju onu koja ima najveću verovatnoću po nekom algoritmu.

Literatura [2] objašnjava da se najneverovatnija sekvenca etiketa i^n_1 za neku sekvencu reči w^n_1 može izračunati tako što će se izračunati proizvod dve verovatnoće za svaku sekvencu etiketa, i izabrati najveći. Dve verovatnoće koje se računaju su verovatnoća prethodne sekvence etiketa (***trigram or transition probability***) $P(t^n_1)$ i verovatnoća pojavljivanja neke reči ***(emission probabilities*** ili ***state observation likelihood***) $P(w^n_1| t^n_1)$:

$$t^n_1 = \text{argmax } P(w^n_1| t^n_1)* P(t^n_1) \quad (1)$$

Ove verovatnoće je još uvek teško izračunati, pa je ovakve algoritme često potrebno dalje uprostiti. Prva pretpostavka je da verovatnoća pojavljivanja neke reči zavisi samo od njene etikete, odnosno da ne zavisi od ostalih reči oko nje. Na primer, ako je posmatrana reč veznik, može se pretpostaviti da je velika verovatnoća da to bude reč „*i*“ jer je on najčešći veznik u srpskom jeziku. Ova verovatnoća se može izračunati tako što se izbroji koliko puta se reč „i“ javlja kao veznik, pa se ta vrednost podeli sa ukupnim brojem reči koje su etiketirane kao veznici.

$$P(wn1|tn1) = \prod_{i=1}^{n} P(w_i|t_i) \quad (2)$$

Druga pretpostavka je da verovatnoća pojavljivanja etikete zavisi samo od prethodne etikete čime ona postaje verovatnoća **tranzicije etikete** (*tag transition probability*). Na primer, u engleskom jeziku zamenice se vrlo često nalaze ispred prideva ili imenica. Pogleda li se sekvenca reči *that(P) flight(N) and the(P) yellow(A) hat(N),* očekivano je da verovatnoće P(N|P) i P(A|P) budu velike. S druge strane, pridevi ne prethode zamenicima, pa će verovatnoća P(P|A) biti mala.

$$P(tn1) = \prod_{i=1}^{n} P(t_i|t_{i-1}) \quad (3)$$

Verovatnoća tranzicije etikete može se izračunati tako što se uzme korpus i izračuna se koliko puta se etiketa N nalazi nakon etikete P, odnosno koliko ima (N,P) bigrama, pa se to podeli sa ukupnim brojem pojavljivanja etikete P. U slučaju trigrama koristi se sličan obrazac. Konačan izbor

etikete biće onaj koji ima maksimalnu vrednost $t^n_1 = \prod_{i=1}^{n} P(w_i|t_i) * \prod_{i=1}^{n} P(t_i|t_{i-1})$. Vrednosti koje se na ovaj način dobijaju mogu biti jako male i često su manje od 10^{-10}. Zbog toga se često pri računanju ovih verovatnoće uzima njihova logaritamska vrednost sa osnovom 2. U ovom radu emisiona i tranziciona verovatnoća izračunata je na ovaj način. Ukoliko se sve navedeno uzme u obzir, konačna formula za računanje verovatnoće da reč w ima etiketu t, ukoliko se kao merilo verovatnoće uzme broj bigrama, je:

$$t = \log_2 \operatorname{argmax} \left(\frac{N(w)}{N(t)} * \frac{N(t-1,t)}{N(t)}\right) \quad (4)$$

U radu je ilustrovano u kakvom su odnosu verovatnoće koje se računaju sa izborom odgovarajuće etikete, kao i na koji način se ove verovatnoće mogu računati. Različite varijacije ovog algoritma mogu koristiti drugačije načine računanja verovatnoća. Za detalje možete pročitati poglavlja 5 i 6 u knjizi [2].

V. ALGORITAM ZA ETIKETIRANJE REČI

Postupak razvoja i testiranja algoritma za etiketiranje reči obuhvata sledeće korake:

1. akviziciju digitalnog izvora: korpusa anotiranih reči za treniranje modela i testiranje algoritma;
2. konverziju u standardni format, probleme izbora skupa karaktera, struktura, navodnika i drugih znakova, zaglavlja, itd.;
3. segmentaciju: rečenice, tokenizaciju – tipove, veličinu slova, skraćenice, složenice, brojeve;
4. morfološku analizu: kreiranje unigram, bigram i trigram matrice pomoću *nltk Python* biblioteke, određivanje skupa etiketa, određivanje tranzicionih i emisionih verovatnoća, optimizaciju korišćenjem „retkih“ reči;
5. treniranje modela pomoću *viterbi* algoritma i *Scikit-learn Python* biblioteke (taj model se kasnije koristi za etiketiranje teksta?;
6. etiketiranje (obradu) programom za etiketiranje;
7. evaluaciju modela, evaluaciju rezultata, poboljšavanje algoritma za etiketiranje, poboljšavanje modela.

Za razvoj bilo kakvog algoritma za etiketiranje ili lematizaciju reči neophodno je prvo pribaviti **digitalni korpus anotiranih reči** nekog jezika koji se može koristiti za **treniranje modela**, a zatim i za **evaluaciju algoritama**. Nažalost, kreiranje ovakvih rečnika vremenski je zahtevno i skupo, jer zahteva angažovanje lingvističkog stručnjaka. Takođe, proces pripreme, obrade i konverzije teksta predstavlja vremenski zahtevan zadatak i vrlo često istraživač u ovoj oblasti provede i do 80% vremena pripremajući materijale za treniranje, testiranje i evaluaciju sistema.

Kao pomoć pri razvoju algoritma korišćena su predavanja i uputstva kursa „*Natural languge proceeesing*“ sa sajta *coursera*[8]. U prvom koraku razvoja algoritma korišćena su dva javno dostupna korpusa ručno anotiranih reči: *INERA* korpus [9] i *MULTEXT-East* anotirani korpus 4.0 [10]. Rečnici se smeju koristiti samo u akademske svrhe i autori ovog rada izražavaju autorima rečnika na dozvoli za korišćenje. Za engleski jezik korišćen je Braunov korpus reči koji sadrži oko 1M anotiranih reči, a primer klase etiketa, skup oznaka koji se koristi za označavanje imenica, glagola itd., je *Penn Tree Bank Tagset*

Ovi korpusi zapisani su u različitim formatima tako da je u drugom koraku našeg algoritma potrebno dobijene korpuse prilagoditi formatu koji razume naš algoritam, a to se radi sledećom funkcijom:

```
SrpLemKor_to_train(DATA_PATH + srcFile,
DATA_PATH + trainFile)
```

U trećem koraku potrebno je korpus podeliti na rečenice, a zatim rečenice dodatno podeliti na reči, odnosno tokene, što se radi funkcijom:

```
brown_words, brown_tags =
split_wordtags(brown_train)
```

U koraku 4 potrebno je odrediti tranzicionu i emisionu verovatnoću. Za računanje broja bigrama i trigrama, kao i za neke pomoćne radnje sa tekstom, korišćen je *nltk Python* biblioteka. NLTK je vodeća platforma za izgradnju *Python* programa za rad sa podacima koji se odnose na ljudski jezik. On pruža jednostavno korišćenje interfejsa za preko 50 korpusa, kao što je *WORDNET*, zajedno sa paketom za obradu teksta, klasifikaciju, tokenizaciju, pronalaženje osnove reči, analizu i semantičko rasuđivanje.

NLTK ima veoma dobru API dokumentaciju i samim tim pogodan je za inženjere, studente, predavače i pojedince generalno. Pored toga, dostupan je za različite operativne sisteme, licenciran je kao *OpenSource* i besplatan je. [11]

```
q_values = calc_trigrams(brown_tags)
```

Radi optimizacije pristupilo se uklanjanju retkih reči, odnosno onih reči koje se premalo puta ponavljaju da bi statistički bile značajne za računanje verovatnoće. Zbog toga su reči zamenjene simbolom _RARE_, čime se postiže optimizacija algoritma i poboljšava preciznost etiketiranja. Algoritam je testiran tako što su sve reči koje se pojavljuju 5 ili manje puta zamenjene ovim simbolom.

```
brown_words_rare = replace_rare(brown_words,
known_words)
```

Kod ovog koraka takođe je potrebno izračunati i emisionu verovatnoću. Ova verovatnoća označava koliko puta se neka nepoznata reč nađe i prepozna, kao npr. imenica u odnosu na celokupan broj imenica. Računamo je kao:

```
e_values, taglist =
calc_emission(brown_words_rare, brown_tags)
```

U petom koraku kreira se model koji će program za etiketiranje kasnije koristiti pomoću *Viterbi* algoritma i *scikit-learn* biblioteke. *Scikit-learn* je *opensource* biblioteka za *Python* koja poseduje različite algoritme za klasifikaciju, regresiju i druge algoritme iz domena mašinskog učenja, i prilagođena je za rad sa ostalim *Python* bibliotekama kao što su „NumPy“ i „SciPy“. *Viterbi* algoritam nalazi verovatnoću najverovatnijeg niza posmatranja, ali pre svega pronalazi niz

odgovarajućih skrivenih stanja lanca Markova, tzv. Viterbijevu putanju, dinamičkim programiranjem.

```
viterbi_tagged = viterbi(brown_dev_words,
taglist, known_words, q_values, e_values)
```

U šestom koraku dobijeni model je iskorišćen za etiketiranje teksta. Tekst koji koristimo za testiranje se prosleđuje algoritmu koji vrši etiketiranje. Dobijeni izlazni fajl ima sledeći oblik:

```
At/ADP that/DET time/NOUN highway/NOUN –
token/etiketa
```

Sve ovo je potrebno kako bi kasnije rezutat, tj. etiketirani tekst, bio upoređen sa originalnim etiketama, i kako bi se dobili rezultati u procentima o tačnosti etiketiranja. Svim znacima interpunkcije dodeljena je etiketa *PUNCT* nakon izvršenog etiketiranja.

U sedmom koraku naš model je evaluiran kroz tri testiranja. Rezultati testiranja navedeni su u sledećem poglavlju. Kao rezultat testiranja dobijeni su pomoćni fajlovi koju su korišćeni prilikom razvoja algoritma. Fajlovi su nazvani od B2 do B6. U fajlu B2 upisuju se sortirani trigrami sa izračunatom tranzicionom verovatnoćom, te on izgleda ovako:

```
TRIGRAM ADV V A -3.84612168402
TRIGRAM ADV V ADV -3.90978032124
TRIGRAM ADV V CONJ -3.45726811655
TRIGRAM ADV V N -3.08854581018
```

U fajlu B3 upisuje se kompletan trening fajl, ali su pritom reči koje se pojavljuju manje od 5 puta zamenjene simbolom RARE, te fajl izgleda ovako:

```
Bio je _RARE_ i _RARE_ _RARE_ dan ; na _RARE_
je _RARE_ _RARE_ .
```

U B4 fajlu upisuje se emisiona verovatnoća za svaki par reč/etiketa i to izgleda ovako:

```
! PUNCT -6.01877487974
! SENT -7.09705215761
" PUNCT -2.31501633096
```

I na kraju, u fajl B5 smešta se konačno etiketiran tekst, odnosno par reč/etiketa.

```
"/PUNCT Dopada/V mi/PRO se/PAR ustrojstvo/V
tvog/A intelekta/N ./SENT
```

VI. Rezultati eksperimenta

U okviru ovog rada realizovana su 3 različita testa: prvi test sproveden je za engleski jezik; drugi je izvršen nad korpusom *MULTEXT-East* za srpski jezik; i treći u kome je korpus *MULTEXT-East* korišćen za treniranje. Odabrani tekstovi iz korpusa *INTERA* korišćeni su za testiranje i evaluaciju algoritma. Za svaki test izvršena je posebna analiza za poznate i nepoznate reči. Pod nepoznatim rečima smatraju se one reči koje nisu korišćene prilikom treniranja modela. Zatim je sračunata matrica pogrešno etiketiranih reči i na osnovu nje tačnost, preciznost i F1 mera za svaku od etiketa. Ukupna tačnost etiketiranja računata je kao količnik broja tačno etiketiranih tokena i ukupnog broja tokena koji se etiketiraju.

A. *Test 1*

U testu 1 korišćen je deo Braunovog korpusa od 265 hiljada reči za treniranje modela, i deo od 30 hiljada reči za evaluaciju modela za etiketiranje reči. Postignuta je tačnost od 89.59%. Broj nepoznatih reči bio je 2262 a tačnost nepoznatih reči 56.27%.

TABELA I – REZULTATI PRVOG TESTA

Broj Testa	Ukupno reči	Tačnost (%)	Tačnost nepoznatih reči (%)	Broj nepoznatih reči
1	30000	89.59	56.27	2262

B. *Test 2*

U drugom testu korišćen je ručno anotirani korpus reči *MULTEXT-East*. Test predstavlja standardni način evaluacije rezultata etiketiranja i sprovodi se tako što se vrši 10 različitih testiranja tako što se anotirani korpus reči podeli na 10 manjih jednakih delova. Veći deo od 9 delova korišćen je za treniranje, a prvi deo ovog teksta za testiranje i evaluaciju modela. Veći deo uvek ima oko 90 hiljada reči a manji deo oko 10000 reči. Test je ponovljen 10 puta tako što je izvršena rotacija delova korpusa koji su korišćeni za testiranje i treniranje (u testu 1 – prvi deo fajla koristi se za testiranje, a ostalo za treniranje, u testu 2 – drugi deo fajla koristi se za testiranje a ostalo za treniranje i tako redom...). Tabela 2 opisuje rezultate ovog testiranja:

TABELA II – REZULTATI DRUGOG TESTA

Broj Testa	Tačnost(%)	Tačnost nepoznatih reči (%)	Broj nepoznatih reči
1	83.65	56.71	1400
2	83.46	53.80	1433
3	84.82	52.29	1394
4	85.62	54.77	1172
5	84.81	54.10	1255
6	84.88	53.62	1298
7	82.4	51.16	1630
8	86.87	54.18	1135
9	87.03	53.86	1101
10	83.76	51.59	1446

TABELA III – MATRICA POGREŠNO ETIKETIRANIH REČI

	PAR	A	ADV	P	V	CONJ	NUM	PREP	PRO	N
PAR	532	11	15	1	33	37	1	4	47	20
A	1	327	27	0	**135**	0	1	1	33	84
ADV	5	16	392	0	3	13	0	12	9	8
P	0	0	0	2571	0	0	0	0	0	0
V	1	**70**	**60**	0	1889	3	2	4	62	**160**
CONJ	33	0	19	0	5	733	0	0	4	0
NUM	0	0	4	0	1	0	82	0	0	1
PREP	0	0	5	0	0	0	0	620	0	0
PRO	6	1	4	0	0	1	0	0	916	0
N	0	**109**	44	0	**257**	0	6	1	25	1413

Procenat reči koje su uvek bile tačno etiketirane je između **84% i 87%,** a procenat tačno etiketiranih nepoznatih reči je oko **53%**. Kako srpski jezik predstavlja jezik sa visokom fleksijom, za razliku od engleskog jezika,

primećuje se pad u procentu tačnosti. Kao primer za dodatnu analizu odabran je test broj 9. Ukupan broj reči u ovom fajlu bio je 10886. U tabelama 3 i 4 prikazan je presek rezultata za taj test. U 1889 slučajeva naš model je glagol prepoznao tačno, ali je 160 puta glagol etiketirao kao imenicu ili je u 60 slučajeva glagol prepoznao kao prilog. Kao što se iz tabele 3 može videti najčešće greške se javljaju na relaciji imenica-glagol, glagol-prilog i prilog-glagol.

TABELA IV – VREDNOSTI PARAMETARA ZA EVALUACIJU. REDOM TAČNOST, PRECIZNOST, ODZIV I F1 MERA

	Acc	P	R	F1
PAR	0.92	0.759	0.92	0.832
A	**0.612**	**0.536**	**0.612**	**0.572**
ADV	0.688	0.856	0.688	0.763
PUNCT	1	1	1	1
V	0.813	0.839	0.813	0.826
CONJ	0.931	0.923	0.931	0.927
NUM	0.891	0.932	0.891	0.911
PREP	0.966	0.99	0.966	0.978
PRO	0.836	0.987	0.836	0.905
N	0.838	0.76	0.838	0.797

C. Test 3

U ovom testu je za treniranje modela korišćen korpus reči *MULTEXT-East*, ali su za evaluaciju korišćeni tekstovi iz različitih domena iz korpusa *INTERA*. Korišćeni tekstovi su podeljeni u edukativne, pravničke i finansijske kategorije. Treba napomenuti da su, po ustanovljenoj praksi, svi testovi korišćeni za testiranje viđeni samo jednom, odnosno autori ih nisu koristili u toku razvoja i validacije algoritma kako bi se postigla objektivnost pri testiranju algoritma za etiketiranje. Rezultati su prikazani u tabeli 5. Prirodno, najveći procenat tačnosti postignut je za tekst koji je iz istog domena kao i trening korpus, dok je za ostale domene procenat tačnosti etiketiranja nešto manji.

TABELA V. – REZULTAT TREĆEG TESTA

Naziv teksta	Tačno etik. Nep. reči (%)	Tačno etik. reči (%)	Broj pogrešno etik. reči	Broj nep. reči	Ukupan broj reči
1984_V3_test ostalo.txt	50,88	83.00	1504	1244	7513
Kvalitet-SR _Edu.txt	55.29	74.29	4513	6586	15286
Radionica-SR Fin.txt	54.92	75.9%	8687	12083	31846
PravaRoma-SR_Leg.txt	50.32	73.05	6680	8592	21405
Radiodif-SR_Leg.txt	45.50	70.22	5102	6189	15005
Strelec-SR_Leg.txt	41.58	70.44	5530	5694	16135
Telekom-SR_Leg.txt	53.83	73.05	5796	7796	18742

VII. ZAKLJUČAK – DISKUSIJA REZULTATA

U ovom radu predstavljeni su rezultati testiranja jednostavnog algoritma za etiketiranje reči. Algoritam je zasnovan na modelu skrivenih lanaca Markova, a za računanje verovatnoće korišćene su odgovarajuće vrednosti bigrama i trigrama reči. Sam algoritam pisan je u programskom jeziku *Python* i pored standardnih koristi i *nltk* i *scikit-learn* biblioteke.

U testu 2 sproveden je unakrsni validacioni test nad jednim korpusom reči, odnosno nad istim domenom. Prosečna tačnost postignuta ovom prilikom je 84% za sve reči, odnosno 53% za nepoznate reči. Najveći broj grešaka primećen je u pogrešno etiketiranim parovima imenica-glagol, glagol-prilog, prilog-glagol. Generalno se može uočiti da prepoznavanje priloga predstavlja najveći problem.

U testu 3 sprovedeno je testiranje algoritma nad tekstovima iz različitih domena što trenutno predstavlja aktivnu oblast istraživanja. Očekivano, najveću tačnost od 83%, uz 51% za nepoznate reči, imao je tekst iz domena na kome je model i treniran. Na ostalim domenima algoritam se ponašao nešto lošije. Ukupna tačnost kretala se između 75% i 70%, a tačnost za nepoznate reči je bila između 55% i 45%.

Najbolji rezultati prijavljeni u drugim radovima, na istom domenu, postignuti su za jezike sa niskom fleksijom, kao što su engleski i nemački. Tačnost iznosi oko 97% za sve reči i oko 89% za nepoznate reči. Što se tiče rezultata za srpski jezik, postupak testiranja i dobijanja rezultata u ovoj oblasti za srpski jezik nije na dovoljno transparentan način obavljen, pa samim tim rezultati teško mogu biti međusobno poređeni. Za dobijanje rezultata u ovom radu korišćeni su javno dostupni anotirani rečnici za treniranje i testiranje algoritma. Zbog toga će u budućnosti biti moguće uporediti neke buduće rezultate i poboljšanja s njima. Samim tim oni predstavljaju preduslov i neophodnu osnovu za dalji razvoj algoritama za etiketiranje reči na srpskom jeziku. Posebno interesantna oblast istraživanja predstavlja kreiranje takozvanih *cross-domain* algoritama za etiketiranje, čemu će autori rada posvetiti posebnu pažnju. Dalji koraci istraživanja bi obuhvatali i uključivanje dodatnih osobina koje bi povećale tačnost etiketiranja nepoznatih reči, kao i povećanje tačnosti etiketiranja problematičnih parova, navedenih u rezultatima testiranja. Još jedan od koraka biće evaluacija već postojećih algoritama nad pomenutim korpusima reči, kao što je Stenfordov *MaxEnt* algoritam.

VIII. LITERATURA

[1] [1] Popović Zoran, „Evaluacija programa za obeležavanje (etiketiranje) teksta na srpskom jezku“, Matematički fakultet Univerziteta u Beogradu, 2008

[2] [2] Martin, James H., and Daniel Jurafsky. "Speech and language processing."International Edition, 2000

[3] Primena Stenfordovog PoS tagera i LemmaGen lematizatora na SrpLemKor korpus, Zbornik radova Visoke Tehničke škole u Nišu, http://www.vtsnis.edu.rs/naucno_istrazivacki_rad/zbornik_vtsnis/zbornik_2015/09_Zbornik_2015_Mil_Kosanovic_S_Stosovic.pdf, 2015

[4] Vlado Delić, Milan Sečujski, Aleksandra Kupusinac, Transformation-based part-of-speech tagging for Serbian language, Recent Advances in Computational Intelligence, Man-machine systems and cybernetics, pages 98-103, ISBN: 978-960-474-144-1,

[5] Christopher D. Manning, Part-of-Speech Tagging from 97% to 100%:Is It Time for Some Linguistics?, 2011

[6] Zeljko Agić, Nikola Ljubešić, and Danijela Merkler. 2013a. Lemmatization and morphosyntactic tagging of Croatian and Serbian.

In Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing, pages 48–57, Sofia, Bulgaria, August. Association for Computational Linguistics, 2013
[7] Miljan Ivković, N gram modeli jezika i reči, Završni rad, Elektronski fakultet u Nišu, 2014
[8] „Natural Language Processing“, Dragomir R. Radev, Ph.D. University of Michigan, online course, Coursera,, https://www.coursera.org/course/nlpintro, 2015
[9] INTERA korpus, http://metashare.ilsp.gr:8080/repository/browse/intera-corpus/61ad85b044c411e29be4842b2b6a04d7c7f3888afa194f75af0084b96a9c58b6/, 2015
[10] Tomaž Erjavec (2012): MULTEXT-East: Morphosyntactic Resources for Central and Eastern European Languages. Language Resources and Evaluation, 46/1, pp. 131-142.
[11] http://www.nltk.org/, 2015
[12] Utvić, M. Annotating the Corpus of Contemporary Serbian. INFOtheca 12, 2, 36a-47a, 2011

ABSTRACT

In this paper we describe the process of the non-supervised part of speech tagging (POS), the one of the basic natural language processing tasks. We base the algorithm on hidden Markov chains and the N-gram language model, and implement it in Python programming language. We elaborate the implementation of the algorithm step by step, from obtaining the digital source to producing the final tagged document. We conduct three experiments on Brown, INTERA and MULTEXT-East annotated corpora for Serbian and English language, and then discuss the obtained results.

IMPLEMENTATION OF POS TAGGING ALGORITHM IN PYTHON FOR SERBIAN LANGUAGE

Milos Kosanovic, Igor Djurovic, Dimitrije Dimitrijevic, Slavimir Stosovic