Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Semantička sličnost parova rečenica

Projekat iz predmeta Pronalaženje skrivenog znanja

Profesor: Dr Veljko Milutinović

Student: Vuk Mirović
Broj indeksa: 3025/2013

Januar 2015.

Sadržaj

Tekst projekta

Tekst zadatka

Semantička sličnost predstavlja koncept dodeljivanja metrike skupovima izraza ili dokumenata zasnovane na sličnosti njihovog značenja. Ovaj koncept jedan je od ključnih za razumevanje prirodnih jezika, jer omogućava pravljenje smislenih poređenja i zaključivanja. Zbog toga određivanje semantičke sličnosti igra važnu ulogu u automatskoj kategorizaciji i sumarizaciji teksta, mašinskom prevođenju, pronalaženju informacija i drugim oblastima veštačke inteligencije. Problem semantičkog poređenja kratkih tekstova (*Short Text Semantic Similarity* – STSS) ima poseban značaj, jer su kratki tekstovi u širokoj upotrebi na Internetu, u formi natpisa i opisa proizvoda, anotacija slika i veb stranica, kratkih novinskih naslova i vesti, itd. Takođe, ovaj problem igra važnu ulogu u pitanjima vezanim za obrazovanje i učenje, kao što su automatsko testiranje i ocenjivanje zadataka.

Zadatak određivanja semantičke sličnosti para rečenica predstavlja dodeljivanje binarne ocene svakom paru, gde ocena jedan ukazuje da su rečenice slične, a nula obratno. Za određivanje semantičke sličnosti potrebno je, na osnovu analize teksta, prvo izračunati vrednosti unapred zatadih atributa. Ovi atributi koriste se za treniranje *data minig* (DM) modela, a zatim se vrši evaluacija performanisi dobijenog modela pomoću validacionog skupa. Dostavljeni skup podataka unapred je podeljen na deo predviđen za treniranje (trening skup), koji iznosi 70% početnog skupa, i preostali deo za evaluaciju (verifikacioni skup). U nastavku je dat opis skupa podataka koji će se koristiti za treniranje i verifikaciju, kao i opis zahtevane realizacije.

Dodatna literatura

- Furlan B., Batanović V., Nikolić B., "Semantic Similarity of Short Texts in Languages with a Deficient Natural Language Processing Support," *Decision Support Systems*, (ISSN)0167-9236, Vol. 55, Issue 3, pp. 710–719, June 2013. PDF

- Furlan B., Sivački V., Jovanović D., Nikolić B. "Comparable Evaluation of Contemporary Corpus-Based and Knowledge-Based Semantic Similarity Measures of Short Texts," *JITA*, vol. 1, no. 1, ISSN 2233-0194 (online), pp. 65-71, June 2011. PDF

- Jovanović D., Furlan B., Nikolić B., "A Software System for Measuring the Semantic Similarity of Short Texts," In ETRAN, Banja Vrućica (Teslić), R. Srpska, BIH, June 6-9, 2011. PDF

Opis podataka

Za evaluaciju realizovanog sistema koristiti MSRPC korpus (*Microsoft Research Paraphrase Corpus*) [1]: MSRPC je najveći korpus parafraza za engleski jezik koji se sastoji od 5801 para rečenica. Svaki par rečenica je ocenjen, tj. data je ocena semantičke sličnosti od strane dvoje sudija. Pri tom, dodeljivane ocene su binarne, gde ocena jedan ukazuje da su rečenice slične, a nula obratno. Slučajeve u kojima je došlo do neslaganja dodeljenih ocena rešavao je treći sudija. Od ukupno 5801 para rečenica, njih 3900 (67%), je proglašeno semantički sličnim, dok je ostatak predstavljao semantički različite parove. Konačno, korelacija u ocenama između troje sudija iznosi 83%, što predstavlja i gornju granicu preciznosti koju sistem može postići nad ovim korpusom.

Ukupan korpus je slučajnim izborom podeljen na dva dela. Prvi deo sadrži približno 70% (4076 parova) i predstavlja trenirening skup, dok drugi deo sadrži preostalih 1725 parova i namenjen je verifikaciji.

Zadaci

Projekat koji izrađuje svaki student sastoji se iz zadataka opisanih u nastavku. Štampane materijale pripremiti prema uputstvima datim u zadacima, a sve zajedno na sledeći način:

1. Na naslovnoj strani jasno napisati naziv predmeta, prezime i ime studenta, broj indeksa i adresu e-pošte.
2. Sve zajedno čvrsto povezati u jednu celinu, tako da se listovi ne mogu rasipati (najbolje spiralom).

Zadatak 1 – priprema podataka (20 poena)

Nad ulaznim tekstualnim fajlovima izvršiti uklanjanje specijalnih karaktera iz rečenica, a zatim izvršiti izračunavanje vrednosti atributa za određivanje sličnosti opisanih u nastavku.

Atributi za određivanje sličnosti - Upotrebom *Stanford CoreNLP* [2] za svaki par rečenica generisati listu atributa na osnovu koje će se dalje vršiti ocenjivanje njihove ukupne sličnosti. Atributi se mogu podeliti u tri grupe:

1. atributi zasnovani na vrstama reči (plitko parsiranje);
2. atributi zasnovani na relacijama među rečima (duboko parsiranje);
3. opšti atributi.

Na kraju, reči koje nisu prepoznate od strane alata označiti sa *UNKNOWN* i izostaviti iz dalje analize. Takođe, ovo uraditi i za kardinalne brojeve (CD).

Atributi zasnovani na vrstama reči

Za svaku vrstu reči koju kao rezultat daje *Stanford POS tagger* (tabela 1) određuju se:

a) Razlika broja reči između prve i druge rečenice za datu vrstu reči. Ovi atributi označavaju se prefiksom *diff_Tag_* praćenim oznakom vrste reči. Npr. atribut *diff_Tag_NNP* označava razliku u broju ličnih imenica jednine. Da bi ovi atributi bili neosetljivi na promenu redosleda rečenica u paru, računa se njihova apsolutna vrednost.

b) Semantička sličnost skupova reči date vrste iz jedne i druge rečenice. Ovi atributi označavaju se prefiksom *semSim_Tag_* praćenim oznakom vrste reči. Npr. atribut *semSim_Tag_NN* označava semantičku sličnost skupova imenica jednine. S obzirom da se upotrebom navedenih alata može odrediti semantička sličnost samo između imenica i glagola, ovi atributi se određuju za ove vrste reči.

c) Leksička sličnost skupova reči date vrste iz jedne i druge rečenice. Ovi atributi označavaju se prefiksom *lexSim_Tag_* praćenim oznakom vrste reči. Npr. atribut *lexSim_Tag_JJ* označava leksičku sličnost skupova atributa.

Pored opisanih, dodati su specijalni atributi koji grupišu imenice i glagole. Razlika u broju svih imenica (opštih, ličnih, jednine, množine) obeležava se *diffNouns*, dok se semantička i leksička sličnost obeležavaju sa *semSimNouns* i *lexSimNouns* respektivno. Glagoli svih oblika na isti način su grupisani i daju atribute *diffVerbs, semSimVerbs* i *lexSimVerbs*.

Oznaka	Vrsta reči	Oznaka	Vrsta reči
CC	Veznik	PRP$	Prisvojna zamenica
CD	Kardinalni broj	RB	Prilog
DT	Determinator	RBR	Prilog – komparativ
EX	Egzistencijalno „*there*“	RBS	Prilog – superlativ
FW	Strana reč	RP	Rečca
IN	Predlog	SYM	Simbol
JJ	Pridev	TO	Predlog „to“
JJR	Pridev – komparativ	UH	Uzvik
JJS	Pridev – superlativ	VB	Glagol – osnovni oblik
LS	Označavač stavke u listi	VBD	Glagol – prošlo vreme
MD	Modalni glagol	VBG	Glagol – gerundiv ili prezent particip
NN	Imenica – jednina ili zbirna	VBN	Glagol – past particip
NNS	Imenica – množina	VBP	Glagol – prezent ne-trećeg lica jednine
NNP	Lična imenica – jednina	VBZ	Glagol – prezent trećeg lica jednine
NNPS	Lična imenica – množina	WDT	Wh-determinator (npr. „*which*“)
PDT	Predeterminator	WP$	Prisvojna wh-zamenica
POS	Prisvojni završetak („*'s*“)	WRB	Wh-prilog (npr. „*how*“, „*where*“, „*why*“)
PRP	Lična zamenica		

Tabela 1 – vrste reči i njihove oznake

Atributi zasnovani na relacijama među rečima

Za svaku relaciju koju kao rezultat daje *Stanford parser* odrediti:

a) Razliku broja datih relacija između prve i druge rečenice. Ovi atributi označavaju se prefiksom *diff_Dep_* praćenim skraćenim nazivom relacije. Npr. atribut *diff_Dep_dobj* označava razliku u broju direktnih objekata. Da bi ovi atributi bili neosetljivi na promenu redosleda rečenica u paru, računa se njihova apsolutna vrednost.

b) Semantičku sličnost dobijenih relacija iste vrste iz jedne i druge rečenice. Ovi atributi označavaju se prefiksom *semSim_Dep_* praćenim skraćenim nazivom relacije. Npr. atribut *semSim_Dep_dobj* označava izračunatu semantičku sličnost skupova direktnih objekata.

c) Leksičku sličnost skupova datih relacija iz jedne i druge rečenice. Ovi atributi označavaju se prefiksom *lexSim_Dep_* praćenim skraćenim nazivom relacije. Npr. atribut *lexSim_Dep_dobj* označava izračunatu leksičku sličnost skupova direktnih objekata.

Opšti atributi

Opšti atributi nisu zasnovani na rezultatima parsiranja, već na skupovima svih reči u rečenicama. Prvi atribut je apsolutna razlika ukupnog broja reči između prve i druge rečenice i naziva se *diff_All*. Drugi atributi je ukupna leksička sličnost *overallLexsim* koja uzima u obzir sve reči iz jedne i druge rečenice i računa njihovu leksičku sličnost na način opisan u nastavku bez uzimanja u obzir kojoj vrsti reči pripada određena reč.

Semantička i leksička sličnost

Atributi zasnovani na vrstama reči (plitko parsiranje) - Za određivanje vrednosti atributa koji predstavljaju semantičku i leksičku sličnost skupova reči pojedinih vrsta koristi se sledeći algoritam (pojednostavljena verzija algoritma [3]):

1. Rečenice *r1* i r2, koje sačinjavaju jedan par čija se sličnost određuje, formiraju dva skupa reči, odnosno dva skupa lema reči koje ove rečenice sadrže. Za svaku vrstu reči *t*, iz ova dva skupa izdvajaju se dva podskupa koji sadrže samo ove vrste reči. Na dalje to su skupovi st_1 i st_2:

   $$st_1 = \{w_{11}, w_{12}, \dots, w_{1n}\}$$
   $$st_2 = \{w_{21}, w_{22}, \dots, w_{2m}\}$$

   n i *m* predstavljaju broj reči date vrste *t* u rečenicama *r1* i *r2* respektivno. Unija ova dva skupa smešta se u novi:

   $$st = \{w_1, w_2, \dots, w_r\}$$

   gde je *r* broj reči u skupu *st*.

2. Formira se vektor *v1* dužine *r*, u kome je vrednost svakog *i*-tog člana jednaka maksimalnoj semantičkoj ili leksičkoj sličnost između *i*-tog člana niza *st* i svih članova niza *st1*.

   $$v1 = \{sim_{max}(w_1, st_1), sim_{max}(w_2, st_1), \dots, sim_{max}(w_r, st_1)\}$$

   vrednost sličnosti *sim* računa se u zavisnosti od toga da li je u pitanju semantičko ili leksičko poređenje.

3. Na isti način na koji je kreiran vektor v_1, kreira se vektor v_2 u kome će vrednost svakog *i*-tog člana biti maksimalna semantička ili leksička sličnost između *i*-tog člana niza *st* i svih članova niza st_2.

4. Kosinusna sličnost vektora *v1* i *v2* predstavlja rezultujuću sličnost između skupova reči st_1 i st_2:

   $$sim(st_1, st_2) = \frac{v_1 \cdot v_2}{\| v_1 \| * \| v_2 \|}$$

Atributi zasnovani na relacijama među rečima (duboko parsiranje) - Za određivanje semantičke i leksičke sličnost skupova pojedinih relacija koje su iste vrste *r*, vrednost sličnosti dve relacije rel_1 i rel_2 računati na sledeći način:

$$sim_{dep}(rel_1, rel_2) = sim_{word}(g_1, g_2) \cdot \mathbf{2}^{\boldsymbol{sim_{word}(d_1, d_2)-1}}$$

gde su g_1 i d_1 vodeća i zavisna reč u relaciji rel_1, a g_2 i d_2 vodeća i zavisna reč u relaciji rel_2. Funkcija sim_{word} predstavlja jedan od pristupa opisanih u nastvaku za računanje semantičke, odnosno leksičke sličnost, između dve reči.

1. Algoritam (tj. pojednostavljena verzija algoritma [4]) polazi od skupova relacija sr_1 i sr_2 određenog tipa *r*:

$$sr_1 = \{rel_{11}, rel_{12}, \dots, rel_{1n}\}$$
$$sr_2 = \{ rel_{21}, rel_{22}, \dots, rel_{2m}\}$$

i konstruiše matricu $n \times m$ gde su *n* i *m* broj relacija datog tipa u rečenicama.

$$M = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mn} \end{bmatrix}$$

Elementi matrice predstavljaju sličnosti pojedinačnih relacija:

$$\alpha_{ij} = sim_{dep}(rel_{1i}, rel_{2j}), \text{ gde je } 0 \leq \alpha_{ij} \leq 1.$$

2. Nakon konstrukcije matrice pronalazi se maksimalan element αij i dodaje se u listu β, a svim elementima matrice iz vrste *i* i kolone *j* dodeljuje se vrednost 0. Operacija se ponavlja sve dok u matrici ima elemenata većih od 0. Na kraju se dobija lista β sa maksimalnim elementima po vrstama i kolonama,
3. Za dobijanje rezultujuće sličnosti koristi se sledeća jednačina:

$$sim(sr_1, sr_2) = \frac{(\sum_{i=1}^{|\beta|} \beta_i) * (m+n)}{2mn}$$

Leksička sličnost para reči

Za određivanje leksičke sličnosti koristi se algoritam *Normalized Maximal Consecutive Longest Common Subsequence starting at any character n* (*NMCLCSn*), implementiran pomoću metode *longestCommonContiguousSubstring* dostupne unutar *Stanford CoreNLP* paketa, na sledeći način:

```
import edu.stanford.nlp.util.StringUtils;
/**
* Normalized Maximal Consecutive Longest Common Subsequence starting at
* character n
* http://www.site.uottawa.ca/~mdislam/publications/tkdd.pdf
* @param s1
* @param s2
* @return
*/
public static double NMCLCSn(String s1, String s2) {
        if(s1==null || s2==null || s1.length()==0 || s2.length()==0) return 0;
        return Math.pow(StringUtils.longestCommonContiguousSubstring(s1, s2),2) /
        (s1.length() * s2.length());
}
```

Semantička sličnost para reči

Za određivanje semantičke sličnosti para reči koristiti algoritam *lin* [5] dostupan u alatu *Java WordNet::Similarity* [6]. Pogledati TestExamples.java kao primer upotrebe.
Za vrste reči za koje nije podržano određivanje semantičke sličnosti odrediti samo leksičku sličnost.

Zadatak 2 – treniranje modela (20 poena)

Učitati u bazu podataka skupove podataka za treniranje i validaciju. Realizovati tri DM model upotrebom sledećih algoritma: (1) stabla odlučivanja (*Decision Trees*) i (2) dva proizvoljna DM algoritama.

Izvršiti evaluaciju dobijenih modela nad validacionim skupom na sledeći način. Rezultati validacije predstavljaju binarne vrednosti koje se mogu predstaviti na način prikazan u tabeli 2.

Ishod verifikacije (predviđena sličnost)	Stvarna sličnost	
	DA (*true*)	NE (*false*)
DA (*true*)	TP (*true positive*)	FP (*false positive*)
NE (*false*)	FN (*false negative*)	TN (*true negative*)

Tabela 2 – predstavljanje rezultata testiranja

Izraz „stvarno pozitivni“ (*True Positives* – TP) odnosi se na one parove rečenica koji predstavljaju parafraze i pravilno su označeni kao takvi od strane algoritma. „Stvarno negativni“ (*True Negatives* – TN) predstavljaju neslične parove rečenica koje je algoritam pravilno prepoznao. „Lažno pozitivni“ (*False Positives* – FP) čine parovi semantički različitih rečeničnca, koji su pogrešno označeni kao parafraze. Konačno „lažno negativni“ (*False Negatives* – FN) su parovi rečenica koji jesu parafraze, ali su pogrešno ocenjeni kao semantički različiti. Na osnovu ovih vrednosti odrediti vrednosti parametara tačnosti (*Accuracy* – A), preciznosti (*precision* – P), osetljivosti (*recall* – R) i F-meru (F). Ove mere intenzivno se koriste u teoriji pretraživanja informacija i računaju se na sledeći način:

$$A = \frac{TP + TN}{TP + FP + FN + TN}$$

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

U kontekstu STSS, tačnost predstavlja odnos broja pravilno ocenjenih parova i ukupnog broja parova rečenica u korpusu nad kojim se vrši evaluacija. Preciznost se može shvatiti kao odnos broja pravilno identifikovanih parova parafraza i ukupnog broja parova označenih parafrazama od strane algoritma. Osetljivost predstavlja odnos između tačno identifikovanih parova parafraza i stvanog

broja parova parafraza u korpusu. F-mera se računa kao harmonijska sredina preciznosti i osetljivosti.

Zadatak 3 – prošireni model (10 poena)

Predložiti dodatne atribute koji će poboljšati učinak dobijenih DM modela. Detaljno dokumentovati ove atribute i prikazati dobijene vrednosti parametara A, P, R i F.

Priprema podataka

Treniranje i validacija sistema se vrši pomocu MSRPC korpusa parafraza za engleski jezik. Taj korpus rečenica je podeljen na dva dela u dva fajla: *msr_paraphrase_train.txt* i *msr_paraphrase_test.txt*. Glavni zadatak pripreme podataka se svodi u tome da je za svaki par rečenica izračunavaj vrednosti atributa koji će se kasnije koristiti u evaluaciji sličnosti rečenica i da se sačuvaju u pogodan oblik za dalju obradu.

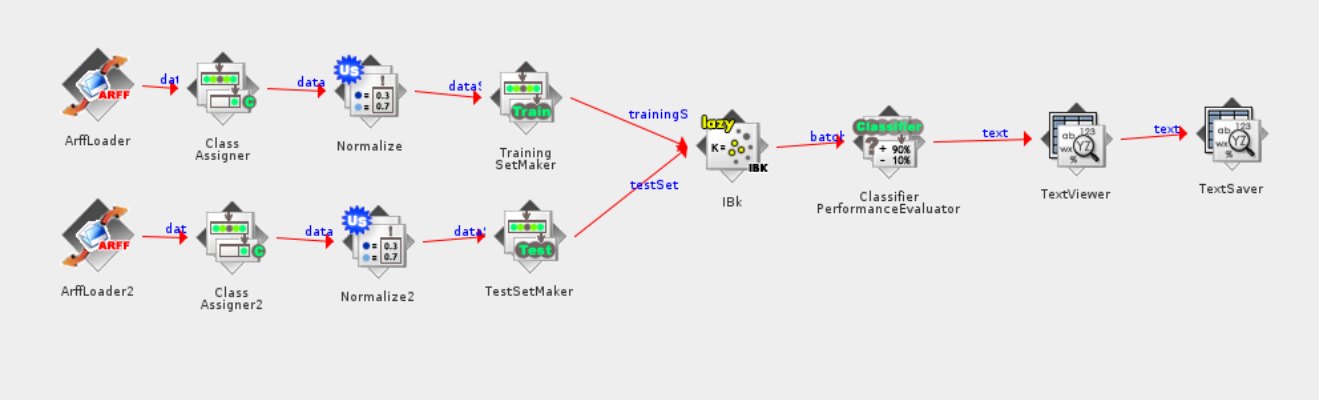
Svaka linija MSRPC korpus fajla sadrži sledeće podatke razdvojene tabulatorom: vrednost ocene sličnosti parafraza, identifikator prve rečenice, identifikator druge rečenice, sadržaj prve rečenice i sadržaj druge rečenice. Nas zanimaju samo vrednost ocene izraza i sadržaji rečenica. Preprocesiranje .txt fajlova i ekstrakcija samo relevantnih podataka i priprema u pogodniji oblik za obradu se nalazi u Python skripti scripts/clean_data.py. Dalje procesiranje se radi u Java programu u direktorijumu *src*, potrebne zavisnosti se nalaze u direktoijumu *deps* (*Stanford CoreNLP* i *WordNet Similarity* paketa). Podaci nad kojima radi program se nalaze u direktorijumu *data*.

Određivanje vrste reči i relacija među rečima u rečenica je realizovano u funkciji *processSentence* u klasi Data. Ova funkcija kao ulaz prima sadržaj rečenice i pomocu paketa Stanford CoreNLP se vrši određivanje vrste reči i relacija koje postoje među rečima. Procesirane podatke vraća u instanci klase *ProcessedSentence.* Posle se poziva funkcija *calculateAllMetrics* koja izračunava sve potrebne metrike tražene u projektu i vraća instancu klase *AllMetrics* koja ima sve potrebne metrike i zatim sačuva metrike u format pogodan za dalju obradu: *ARFF*. Funkcija *calculateAllMetrics* pozivaza svaku metriku odgovarajuću funkciju iz klase *Data* i prosleđuje joj potrebne podatke. Sam paket *Stanford CoreNLP* ima mogućnost za određivanje osnovnih(*Basic*) zavisnostima među rečima i zavisnostima koje dodatno sadrže informacije o predlozima i konjuktivima(*Collapsed*) i one zavisnosti koje to ne sadrže(*Non-collapsed*). Za potrebe ovog projekta su korišćene osnovne relacije (*Collapsed dependecies*) među rečima. ARFF format fajla je specifičan za alat *Weka* koji je korišćen za realizaciju tri DM modela. U zaglavlju takvog fajla se za svaki atribut nalazi definicija: naziv i tip vrednosti tog atributa. Nakon zaglavlja se nalazi *data* deo gde su u svakom redu vrednosti svih atributa za jedan par rečenica. Poslednji atribut u tom fajlu je ciljani(*target*) atribut koji ima naziv *class*.

Treniranje i evaluacija modela

Za realizaciju tri DM modela koriććen je alat otvorenog koda *Weka* verzije 3.7.12. Ovaj alat za generisanje DM modela je odabran zbog svoje jednostavnosti i intuitivnosti pri korišćenju.

Za izgradnju DM modela korišćen je podprogram *KnowledgeFlow* iz Weka paketa koji omogućava da se grafički nacrta i prikaže tok podataka za DM model. Za svaki od tri DM modela postoji napravljen poseban grafički prikaz. Za grafički prikaz DM modela su korišćeni komponente: *ArffLoader*, *Normalize*, *ClassAssigner*, *TrainingSetMaker*, *TestSetMaker*, *TextViewer*, *TextSaver*, *ClassifierPerformanceEvoulator*, i komponente koje predstaljaju relizaciju nekog od tri odabrana DM algoritma. Komponenta *ArffLoader* služi za učitavanje *.arff* fajla. Postoje dve instance *ArffLoader*-a pošto je potrebno učitati dva fajla: jedan koji sadrži trening skup i jedan koji sadrži skup za validaciju. U komponenti *ClassAssigner* se konfigurise atribut koji predstavlja ciljani(*target*) atribut. *Normalize* normalizuje atribute min-max normalizacijom na opseg 0-1. *TrainingSetMaker* i *TestSetMaker* su komponente koje se samo koriste kako bi se ova dva skupa razdvojila pri povezivanju na komponentu algoritma. *ClassifierPerformanceEvoulator* je komponenta koja služi da isprocesira rezultate validacije i da ih pretvori u tekstualno razumljiv oblik koji će koristiti naredna komponenta u lancu – *TextViewer*. Komponenta *TextViewer* omogućava da se vidi rezultat validacije sa vrednostima atributa koje obezbeđuje sam alat *Weka*. *TextSaver* je komponenta kojoj se zadaje putanja do fajla do mesta gde treba da se snimi fajl koji predstavlja rezultat validacije. Kada se snimi rezultat validacije u tekstualni fajl, naredni korak je procesiranje tog fajla Python skriptom u *scripts/calc_metrics.py*. koji isparsira rezultat modela iz Weka-e i izračunava vrednosti parametara tačnosti(A), preciznosti(P), osetljivosti(R) i F-mera(F).Primer izgleda toka podataka za jedan od DM modela se može videti na slici 1.



Slika 1: KnowledgeFlow za algoritam kNN

DM modeli su izgrađeni pomoću tri algoritma: *C4.5*, *K-NearestNeighbour* i *CART*. Ovi algoritmi su odabrani jer spadaju u deset najčešće korišćenih algoritama koji se ističu i po velikoj tačnosti.

Algoritam *C4.5* je *decision tree* koji služi za određivanje vrednosti ciljnog atributa na osnovu izgradnje stabla odlučivanja. Algoritam u alatu *Weka* se nalazi pod nazivom *J4.8*. Ovom algoritmu je moguće zadavati nekoliko parametara od kojih su najbitniji parametri: *minNumObj* i *confidenceFactor*. Parametar *minNumObj* označava broj minimalnih instanci trening skupa koje se moraju naći u svakom listu stabla. *ConfidenceFactor* je parametar koji se koristi pri odsecanju tako

da manje vrednosti ovog faktora forsiraju više odsecanja. Kombinacijama ovih parametara su dobijane različite vrednosti tačnosti modela(*Accuracy*) nad validacionim skupom što se može videti u tabelama 1 i 2. U tabeli 1 je variacija *minObj* za zakucani *confidenceFactor* 0.1, a u tabeli 2 variacija *confidenceFactor*-a za zakucan *minObj* 100. Statistički gledano najbolju preciznost DM model izgrađen na bazi *C4.5* algortima je postigao kada se radi odsecanje stabla sa *confidenceFactor* od 0.1 i kada je *minObj* 100.

minObj	**Accuracy**
10	72.81
50	73.51
100	74.78
150	74.37
200	73.15

confidenceFacor	**Accuracy**
0.05	74.78
0.1	74.78
0.15	74.20
0.25	74.25
0.5	73.73
0.75	73.73

U tabeli 3. mogu se videti vrednosti parametara tačnosti, preciznosti, osetljivosti i F-mera.

Tačnost(A)	73.51%
Preciznost(P)	76.26%
Osetljivost(R)	87.36%
F-mera(F)	81.43%

Algoritam *K-NearestNeighbour* je primer algortima baziranog na izgrađivanju modela na osnovu instanci trening skupa tako što se svaka instanca validacionog skupa poredi sa instancama trening skupa i glasanjem se dobije vrednost ciljanog atributa. Algoritam u alatu *Weka* se nalazi pod nazivom *IBk*. Ovom algoritmu je moguće zadavati nekoliko parametara od kojih su najbitniji parametri *kNN* i *distanceWeighting*. Parametar kNN predstavlja broj suseda koji se uzima u obzir pri glasanju. Bitno je da ovaj parametar bude neparan broj kada ciljani atribut može imati samo dve vrednosti da se ne bi desila "nerešena" situacija pri glasanju. Za pronalaženje najbolje tačnosti ovog modela, ovaj parametar nije uzimao vrednost 1 iz razloga da se ne bi došlo do pretreniranosti modela i situacije da se izgubi generalizam trening skupa. Parametar *distanceWeighting* pokazuje da li se za biranje k najboljih suseda koristi težinsko glasanje kako bi bliži susedi imali više uticaja na krajnje glasanje. Uzeta je u obyir evaluacija bez težinskog glasanja i sa invertnim težinskim glasanjem. Statistički gledano najbolju preciznost DM model izgrađen na bazi *K-NearestNeighbour* algortima je postigao kada se u obzir pri glasanju uzima 19 najbližih suseda i kada je težina pri glasanju jednaka recipročnoj vrednosti razdaljine do suseda.

kNN	**distanceWeighting**	**Accuracy**
5	No	67 %
13		69,31 %
19		70.49 %
21		70.26 %
23		70,08 %

kNN	**distanceWeighting**	**Accuracy**
5	1/distance	67,53 %
13		69,62 %
19		70.78 %
21		70.60 %
23		70,43 %

U tabeli 5. mogu se videti vrednosti parametara tačnosti, preciznosti, osetljivosti i F-mera.

Tačnost(A)	69.62 %
Preciznost(P)	72.62%
Osetljivost(R)	87.18%
F-mera(F)	79.24%

Algoritam *CART* je *decision tree* koji služi za određivanje vrednosti ciljnog atributa na osnovu izgradnje stabla odlučivanja. Algoritam u alatu *Weka* se nalazi pod nazivom *SimpleCART*. Ovom algoritmu je moguće zadavati nekoliko parametara od kojih su najbitniji parametri: *minNumObj* i *numFolds*. Parametar *minNumObj* označava broj minimalnih instanci trening skupa koje se moraju naći u svakom listu stabla. Kombinacijama ovih parametara su dobijane različite vrednosti tačnosti modela(*Accuracy*) nad validacionim skupom što se može videti u tabeli 6. U tabeli 6 je variacija *minObj* za zakucani *numFolds* 5, a u tabeli 7 variacija *numFolds*-a za zakucan *minObj* 10. Statistički gledano najbolju preciznost DM model izgrađen na bazi *CART* algortima je postigao za parametre *numFolds* 5 i kada je *minObj* 10.

minObj	**Accuracy**
5	73.68
8	73.68
10	73.68
15	72.40
20	72.33

numFolds	**Accuracy**
3	72.95
5	73.68
7	73.33
10	73.04

U tabeli 8. vrednosti parametara tačnosti, preciznosti, osetljivosti i F-mera.

Tačnost(A)	73,68 %
Preciznost(P)	77,87 %
Osetljivost(R)	84,39 %
F-mera(F)	81,00 %

Proširivanje modela novim atributima

Da bi se poboljšao učinak dobijenih DM modela dodato je jedan atributa: *sameWords*. Vrednost atributa *sameWords* predstavlja broj reči koje su zajedničke za obe rečenice u odnosu na prosečnu dužinu rečenica.

U tabeli 9. se mogu videti da su poboljšane vrednosti parametara atributa tačnosti, preciznosti, osetljivosti i F-mera kada se tri ranije predstavljena modela izgrade i validiraju na osnovu ovog dodatnog atributa.

	C4.5	**C4.5 new**	**Gain**
A	73.51	73.91	0.54%
P	76.26	76.76	0.65%
R	87.36	87.27	-0.1%
F	81.43	81.68	0.31%
	kNN	**kNN new**	**Gain**
A	69.62	69.74	0.17%
P	72.62	72.76	0.19%
R	87.18	87.1	-0.09%
F	79.24	79.29	0.06%
	CART	**CART new**	**Gain**
A	70.14	71.88	2.48%
P	75.65	77.27	2,14%
R	81.26	81.78	0.6%
F	78.35	79.46	-1.14%

Literatura

[1] “Microsoft Research Paraphrase Corpus,” 2005. [Online]. Available: http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042/

[2] The Stanford Natural Language Processing Group, “Stanford CoreNLP.” [Online]. Available: http://nlp.stanford.edu/software/corenlp.shtml

[3] L. Li, Y. Zhou, B. Yuan, J. Wang, and X. Hu, “Sentence similarity measurement based on shallow parsing,” in *Fuzzy Systems and Knowledge Discovery*, 2009, pp. 487–491.

[4] A. Islam and D. Inkpen, “Semantic text similarity using corpus-based word similarity and string similarity,” *ACM Trans. Knowl. Discov. Data*, vol. 2, no. 2, pp. 1–25, Jul. 2008.

[5] T. Pedersen, S. Patwardhan, and J. Michelizzi, “WordNet:: Similarity: measuring the relatedness of concepts,” in *HLT-NAACL*, 2004.

[6] D. Hope, “Java WordNet::Similarity.” [Online]. Available: http://www.sussex.ac.uk/Users/drh21/Java WordNet Similarity beta version 11.01.source.zip

[7] I. H. Witten, E. Frank, Mark A. Hall , “Data Mining – Practical machine learning tools and tehniques,” 3rd Edition, 2011, pp. 403–583.