Inspiratron.org - Natural language processing, machine learning and cybersecurity

Klasifikator sentimenta za srpski jezik

by Nikola Miloševi? - Saturday, October 20, 2012

https://inspiratron.org/blog/2012/10/20/klasifikator-sentimenta-za-srpski-jezik/

S obzirom da ulazim u poslednju fazu mog master rada, koji se bavi mašinskom sentiment analizom srpskog jezika, mislim da bi bilo dobro da napišem par re?i o tome ovde. Sentiment analizator je zapravo i razlog zakupa ovog domena. Evo o ?emu se radi:

Analiza sentimenta je oblast mašinskog procesuiranja prirodnog ?ovekovog govora ?iji je cilj otkriti sentiment odre?enog teksta ili re?enice, odnosno subjektivan ose?aj tog dela teksta. Najprostije re?eno da li re?enica ima pozitivan ili negativan kontekst. Radi se o klasifikaciji teksta u dve klase u ovom slu?aju - pozitivnu i negativnu. Klasifikacija može biti i u više klasa (mnogo pozitivno, pozitivno, neutralo, negativno, mnogo negativno i sl.), me?utim ovde ?emo se baviti samo binarnom klasifikacijom. Prilikom ove klasifikacije postoje odre?eni izazovi, koji je razlikuju od obi?ne klasifikacije teksta po temama. Recimo, negacija uti?e mnogo više na sentiment re?enice nego na njegovu temu. Pa tako re?enica "Ovo je dobar ?ovek" i "Ovo nije dobar ?ovek" po klasifikaciji teme su jako sli?ni, ali mnogo razli?iti kada se radi o sentimentu. Naravno, postoje odre?eni problemi, koji još u nau?nim krugovima su predmet istraživanja i koji nisu adekvatno rešeni, poput ironije.

Klacifikacija teksta i analiza sentimenta su naro?ito bitni u mnogim sociološkim i poslovnim procesima. Tema je postala naro?ito interesantna nakon 2001. godine, kada je ekspanzija interneta, blogova, foruma i socijalnih mreža donela velike mogu?nosti za eksploataciju analize sentimenta i klasifikaciju uopšte. Danas, analiza sentimenta je bitna kako u poslovnoj sferi, gde se želi saznati utisak o odre?enom proizvodu na osnovu podataka prikupljenih sa interneta, preko politike gde se želi saznati kakav je utisak javnosti o odre?enom kandidatu, do socijologije koja može imati veliki benefit od razvoja ove oblasti. Tako?e klasifikacija se koristi u razli?itim servisima za online oglašavanje, ?esto i zajedno sa sentiment analizom, jer ukoliko je tekst o odre?enom proizvodu napisan u negativnom kontekstu ne bi bilo dobro pored teksta postaviti reklamu sa baš tim proizvodom. Naravno navedeni na?ini koriš?enja su mnogo brži i jeftiniji od klasi?nog istraživanja javnosti ili pak donose dodatnu vrednost u oglašavanju.

Za ura?eni klasifikator je iskoriš?en Naive Bayes algoritam i posebno razvijen stemer za srpski jezik. Prema mojim informacijama radi se o jednom od prvih sentiment analizatora za srpski jezik, kao i jedan od od prvih ura?enih stemera za srpski jezik sa malim brojem pravila (oko

300, raniji stemer za koji znam je stemer Danka Šipke i Vlade Kešenja sa preko 1000 pravila i manjom ta?noš?u, bar za task analize sentimenta).

O stemeru ?u pisati posebno, tako da ?u sad objasniti kako radi Naive Bayes klasifikator.

Naive Bayes je algoritam supervizovanog mašinskog u?enja, što zna?i da postoji trening set podataka koji su labelisani i gde su odre?ene klase dokumenata. Na osnovu ovih dokumenata i re?i u njima se radi dalja statisti?ka analiza i odre?uje klasa novih, nepoznatih dokumenata.

Naive Bayes klasifikator je zasnovan na Bayesovom pravilu koje glasi:

$$p(C|F_1,\ldots,F_n) = \frac{p(C)\ p(F_1,\ldots,F_n|C)}{p(F_1,\ldots,F_n)}.$$

Gde je C klasa, a F feature, odnosno re?. Interpretacija glasi da je verovatno?a da re?i F1 do Fn pripadaju klasi C, jednaka verovatno?i klase C pomnožene sa verovatno?om da se u klasi C nalaze feature-i F1 do Fn, podeljenom sa verovatno?om featur-a. U praksi interesuje nas samo gornji deo, jer donji ne zavisi od C. Pa se tako jedna?ina može napisati kao:

$$\begin{aligned}
&p(C, F_1, \ldots, F_n) \\
&\qquad \propto p(C)\ p(F_1, \ldots, F_n|C) \\
&\qquad \propto p(C)\ p(F_1|C)\ p(F_2, \ldots, F_n|C, F_1) \\
&\qquad \propto p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3, \ldots, F_n|C, F_1, F_2) \\
&\qquad \propto p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2)\ p(F_4, \ldots, F_n|C, F_1, F_2, F_3) \\
&\propto p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3|C, F_1, F_2)\ \ldots p(F_n|C, F_1, F_2, F_3, \ldots, F_{n-1}).
\end{aligned}$$

Ovde su uvodi naivni predpostavka da featur-i, odnosno re?i F1...Fn ne zavise jedni od drugih, odnosno da su potpuno nezavisni. Tako jedna?ina dobija oblik:

$$\begin{aligned}
p(C, F_1, \ldots, F_n) &\propto p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots \\
&\propto p(C) \prod_{i=1}^{n} p(F_i|C).
\end{aligned}$$

Odnosno:

$$p(C|F_1,\ldots,F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

Odnosno najverovatnija klasa je ona ?ija je verovatno?a najve?a. Ra?una se nad istim re?ima i za pozitivnu i za negativnu, uporede, i klasa je ona ?ija je verovatno?a ve?a.

Verovatno?a klase se ra?una kao zbir svih dokumenata odre?ene klase, podeljena sa zbirom svih dokumenata. Verovatno?a re?i u klasi se ra?una kao zbir pojavljivanja odre?ene re?i u toj klasi podeljen za zbirom pojavljivanja svih re?i u trening setu.

Problem u ovakvom izra?unavanju postoji sa re?ima koje ne postoje u trening setu. Njihova verovatno?a bi u ovom slu?aju bila 0, što bi poremetilo ceo proces odre?ivanja klase. Zato se uvodi poravnavanje koje predpostavlja da su i ti dokumenti javljeni jednom, kao i da svi ostali dokumenti su se javili za jedan više put nego što zaista jesu. Ovakvo poravnavanje se zove Laplasovo poravnavanje (Laplace smooting). Na ovaj na?in je mogu?e dobiti kvalitetno klase i za nove podatke koji se nisu javili u trening setu.

Pre ubacivanja teksta u Naive Bayes klasifikator, bilo da se radi o u?enju ili da se radi o odre?ivanju sentimenta novih re?enica potrebno je uraditi nekoliko obrada. Prvo je potrebno uraditi stemming, odnosno uklanjanje sufiksa, kako bi re?i razli?itih fleksija imali isti oblik. Potrebno je tako?e dovesti sva slova na istu veli?inu, odnosno smanjiti velika slova da budu mala. Tako?e ura?ena je i obrada negacija. Nakon pojavljivanja re?ice ne ili glagolske negacije glagola jesam (nije) ili hteti (ne?u), dodat je prefix NE_ svim re?ima do znaka interpunkcije. Na taj na?in je obezbe?eno da te re?i iz re?enice negativnog konteksta ne budu zajedno sabirani sa re?enicama pozitivnog konteksta u bazi.

Detaljnije o svemu ovome, strukturi baze i samom algoritmu, kada odbranim rad.

Nikola Miloševi?
