Сврставање појмова на континуалној скали позитивно-негативно према поларитету осећања засновано на емотиконима

УДК 811.163.41’322.2

Михаило Шкорић
miks@tesla.rcub.bg.ac.rs
Универзитет у Београду

САЖЕТАК: Циљ овог рада је скретање пажње на могућност коришћења текста на вебу обележеног емотиконима и другим одређивачким нискама у сентименталној анализи језички независног корпуса. Он уводи неколико новина у постојећу истраживачку окосницу и тестира њихову успешност. Такође, приказује софтверски алат специјално направљен у ту сврху, објашњава систем његовог рада у изради базе података са појмовима који одражавају одређени степен осећања и даје упуство за његово коришћење. Осим тога, приказује и софтверски алат за тестирање базе података и даје примере анализе добијених резултата.

КЉУЧНЕ РЕЧИ: истраживање података, поларитет осећања, екстракција информација, емотикони, текст на вебу.

РАД ПРИМЉЕН: 24. јануар 2017.
РАД ПРИХВАЋЕН: 25. март 2017.

1. Увод

Приликом израде софтвера за разумевање природног језика, широко су прихваћена два приступа:

– Софтвер који нема дубоко разумевање значења текста, већ само граматике језика којим је он написан, што омогућава ширу примену.

– Софтвер који има дубље разумевање самог значења текста, често ограничен на једну или само неколико области које покушава да разуме и претежно се користи за класификацију текстова.

Системи засновани на анализи осећања додељују тексту вредности над параметрима осећања, где већи број параметара значи много већу комплексност дефинисања.

Под претпоставком да се приликом сравњивања свих могућих параметара на поларитет осећања не долази до великог губитка информација, и да машина може да одлучи између више избора мерењем једног јединог параметра, задатак се своди на одређивање тога шта је позитивно, а шта негативно.

У циљу што економичнијег прављења интелигентног система, не треба да се занемари иједан ресурс који стоји на располагању и који може потенцијално да буде од користи. Основна идеја овог истраживања је коришћење метода истраживања података за проналажење метаподатака – одређивача, које корисници друштвених мрежа ненамерно остављају у својим порукама (у облику емотикона или језички-универзалних устаљених фраза) и додељивање вредности позитивности појмовима у скупу у којем се они налазе. Како су одређивачи језички независни, и овај систем би био језички независтан. Могао би, уколико се испостави као валидан, да омогући коришћење метода машинског учења над огромним корпусом текстова унапред обележених одређивачима.

1.1 Осврт на прећашња слична истраживања

Године 2005. објављена је серија експеримената са класификацијом расположења постова на интернет блоговима, која је послужила као основа за многа будућа истраживања (Mishne, 2005). Ови експерименти састојали су се од евиденције појмова који су се појављивали у постовима за које су сами аутори тврдили да су одражавали одређено расположење док су их писали. Направљени су индекси појмова и израчуната је њихова фреквенција у постовима који су поистовећивани са једним од девет различитих расположења: забављеност, умор, срећа, радост, досада, осећај успеха, поспаност, задовољство и узбуђење. Нови постови су затим тестирани на фреквенцију речи како би се утврдило расположење аутора који их је писао. Резултати су упоређени са људском проценом истих постова и закључено је да машина процењује расположење аутора само мало лошије од човека.

На Универизитету у Токију је 2009. године објављено истраживање у оквиру кога се у тексту анализирало девет расположења, коришћењем комплексних коначних аутомата који препознају граматичке структуре текста (Neviarouskaya et. al., 2009). Исте године истраживачи са Универзитета Станфорд у Калифорнији представили су нови систем анализе веб-постова коришћењем алгоритама који су тренирани да препознају емотиконе и који су додељивали позитивну или негативну вредност расположења порукама на *Twitter* платоформи (Go et. al., 2009). Циљ истраживања био је да се направи систем за класификацију постова, како би потрошачи могли да истраже став претходних купаца пре куповине неког производа. Неколико различитих алгоритама за машинско учење тренирано је са осам емотикона (пет позитивних и три негативна), а резултати су показали да је њихова прецизност изнад 80% у погађању расположења у постовима.

Истраживачи са Хебрејског универзитета у Јерусалиму су 2010. године спровели још једно слично истраживање расположења израженог у постовима на платформи *Twitter*, узимајући у обзир, поред 15 емотикона, и 50 тагова,[1] што је њихов оригинални допринос (Davidov et. al., 2010). Алгоритми који су тренирани на таговима, такође су показили успешност у препознавању расположења поста.

У поменутим истраживањима емотикони се у тексту третирају као ниске карактера експлицитног значења. Другачији приступ предложили су 2010. године истраживачи са Хокаидо универзитета у Сапору. Незадовољни дотадашњим базама емотикона и њиховим вредностима, бавили су се пре свега тиме како да њихову вредност утврде што тачније. Идеја је била да се емотикони третирају као структура састављена од засебних елемената који представљају очи и уста. Композитни делови посебно су обрађивани како би се израчунала њихова вредност. Када су направили своју базу у њој су се емотикони оцењивали према десет могућих осећања: бес, негодовање, узбуђење, страх, допадање, срећа, олакшање, срамота, туга и изненађење (Ptaszynski et. al., 2010). Ово истраживање је касније проширено, па су 2011. године објавили рад на тему истраживања емотикона у којем су емотикони дефинисани као делови природног језика, те је предложено да њихово истраживање буде укључено у истраживања природног језика (Ptaszynski et. al., 2011).

[1] Корисници платформе *Twitter* имају опцију да своје постове додатно обележе таговима како би постови који говоре о некој теми могли лакше да буду пронађени.

У оквиру овог истраживања спроведена је анкета која је показала да су емотикони други по реду носиоци осећања у реченици после самих лексичких израза.

1.2 Основне информације о експерименту

Овај експеримент има циљ да испроба нови приступ истраживању текста екстракцијом емотикона на нови начин, комбинацијом следеће три идеје:

- Емотиконима који ће се користити у експерименту неће се доделити дискретне вредности попут позитивно или негативно већ ће им бити додељена вредност на скали од најпозитивније до најнегативније. Ово ће се одразити и на појмове чија ће вредност, такође, бити додељена на овој скали.
- Користиће се искључиво универзалне одређивачке ниске независне од специфичног језика. Циљ је да се направи у потпуности језички независтан систем што би умногоме проширило могући корпус учења.
- Уместо засебних порука, попут оних на платформи *Twitter*, користиће се поруке које су део конверзације. Циљ овога је да се тестира могућност деловања одређивача не само на поруку у којој се налазе, већ и на поруке у непосредној близини.

Додатни резултат овог експеримента су и два софтверска алата, специјално направљена ради обављања експеримента. Ови програми ће омогућити будућим истраживачима да тестирају сличне идеје, а детаљно објашњење рада софтвера може помоћи и при изради потпуно новог, бољег и потпунијег алата.

Основа идеја овог експеримента јесте да докаже да је могуће:

- језички независно сачинити инвертовани индекс појмова над корпусом текстова који садржи скуп познатих одређивача.
- помоћу тих одређивача аутоматски доделити појмовима вредности на скали позитивно-негативно, који одражава став људи о тим појмовима.

Експеримент се састоји од два дела:

- аутоматска израда базе података која садржи инвертовани индекс појмова и њихових вредности, коришћењем софтверског алата специјално направљеног за овај експеримент.

– тестирање вредности појмова из базе података упоређивањем са исказима људи и коришћењем софтверског алата за тестирање који је наменски направљен за базе података које су продукт првог дела експеримента.

2. Припрема за израду базе података

База података, продукт овог експеримента, окупља на једном месту појмове (речи које се јављају у разговорима) и вредности расположења које они одражавају (уколико постоје) у бројчаном облику. Вредности су израчунате у односу на то у којој близини у тексту у односу на посматрани појам се јављају одређивачи вредности, који је облик тих одређивача и која је њихова вредност. Одређивачи вредности могу бити емотикони или устаљене фразе које су у разговору јављају, а које по природи нису универзалног значења и одражавају позитиван или негативан став, замењујући тако у писаном тексту изразе лица и интонацију.

Интензитет вредности одређивача директно утиче на интензитет вредности коју он преноси појму. Такође, што је одређивач ближи појму, то ће се његова вредност више одразити на вредност самог појма.

Како би формирање овакве базе било успешно и њен коначни исход задовољавајући, потребно је задовољити три предуслова:

– корпус прикупљених текстова мора да буде организован на одређен начин;
– прикупљени текстови и поруке морају да садрже одређиваче који би допринели додели вредности појмовима који се уз њих налазе;
– одређивачи морају имати унапред одређене вредности.

У наставку текста биће детаљно објашњено како је направљена база података за ово истраживање, од прикупљања самог корпуса тесктова до извоза коначне и потпуне базе података, која се онда може користити на више начина.

2.1 Прикупљање корпуса текстова

Полазна идеја је била да се база као резултат овог истраживања заснива на корпусу текстова који садрже преписку и у оквиру ње одређиваче да изразе позитивне или негативне ставове саговорника.

За овај експеримнет коришћене су датотеке које садрже историју дописивања корисника на друштвеној мрежи *Facebook*, за чију је припремну обраду припремљена одговарајућа XSL трансформација (објашњено у одељку 3.1). Неколицина добровољаца је ове датотеке преузела са званичне веб-странице[2] и проследила их аутору овог рада како би биле укључене у корпус и употребљене у истраживању.

Прикупљено је и коришћено шест датотека (од шест корисника) величине од 1,85MB до 167,09MB. Оне заједно садрже 3.884 разговора 2019 различитих корисника, и 1.843.826 порука које су ти корисници међусобно разменили. Садржај ових порука послужио је приликом прављења базе за ово истраживање. Тиме су испуњена прва два претходно наведена предуслова.

2.2 Додељивање вредности одређивачима

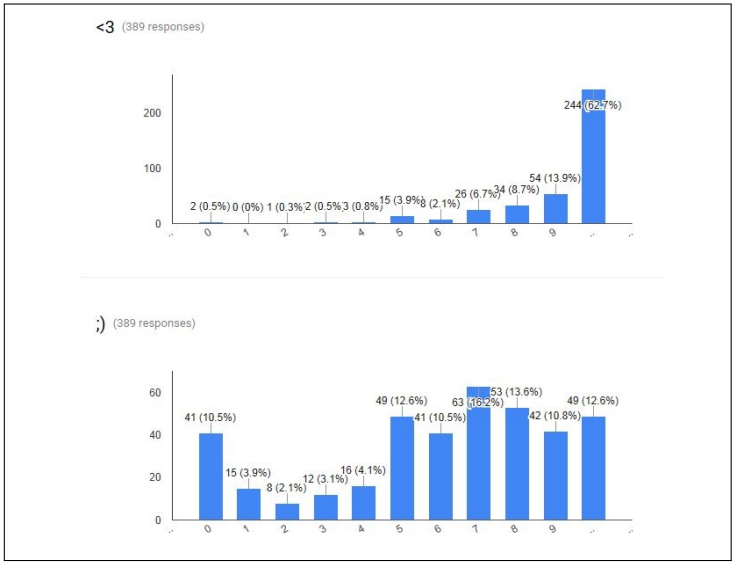
Вредности одређивача добијене су методом анкете која је спроведена на интернет платформи *Google forms*, којој су људи могли да приступе путем хиперлинка објављеног на друштвеним мрежама или прослеђеног од стране других учесника. Учесници анкете су добили задатак да задатим емотиконима и фразама доделе вредност између 0 и 10, где 0 представља највећи интензитет негативног расположења, а 10 највећи интензитет позитивног расположења. Било им је наложено да се при оцењивању воде тиме како одређивач одражава њихово осећање када га шаљу или како га тумаче у примљеној поруци. Учесници анкете су такође имали могућност да сами предложе додатне одређиваче за које су сматрали да су релевантни и предложе њихове вредности.

У периоду од 30 дана 389 учесника анкете оценило је 19 предложених одређивача и предложило додатна 22, од којих је 9 прихваћено.[3] Резултати су се разликовали од одређивача до одређивача – негде су резултати били уједначенији, док негде нису (слика 1).

[2] Друштвена мрежа *Facebook* омогућава свим својим корисницима да путем странице подешавања на свом профилу преузму датотеку која садржи све њихове тренутне мултимедијалне датотеке, као и историју ћаскања у облику јединствене ZIP датотеке.

[3] Укупно 143 корисника су предложила неки одређивач, а потребан број предлагача да би се нови одређивач уврстио био је 48, тј. више од једне трећине предлагача. Уколико је довољан број корисника предложио неки одређивач његова вредност је израчуната као средња вредност свих предложених вредности и уврштен је у истраживање.

По завршетку испитивања, за вредност сваког одређивача узета је аритметичка средина скупа вредности које су испитаници проследили (или предложили за 9 накнадно додатих одређивача).



Слика 1. Пример различите усаглашености корисника над значењем одређивача *<3* (велика сагласност – преко 60% испитаника се слаже око једног одговора) и значењем одређивача *;)* (лоша сагласност – чак пет најчешће одабраних одговора у распону од само 5.7%).

Како би вредности боље осликале оно што треба да представљају – тачку на скали негативно-позитивно – оне су пресликане из скупа $(0, 10)$ на скуп $(-1, 1)$, што је учињено применом формуле:

$$x = (x - 5)/5 \tag{1}$$

Тако је вредност која одговара негативном расположењу највећег интензитета постала -1, док је вредност која одговара позитивном расположењу највећег интензитета постала 1. Израчунат је, такође, универзални фактор корекције којим су све вредности помножене. За овај фактор узет је количник највеће вредности 1 и вредности

одређивача са највећим позитивним интензитетом $0,81$, тако да одређивач *<3*, који према учесницима у анкети одражава позитивно расположење највећег интензитета, добио максималну вредност 1. Тиме што су свим одређивачима додељене одговарајуће вредности испуњен је трећи услов за формирање базе.

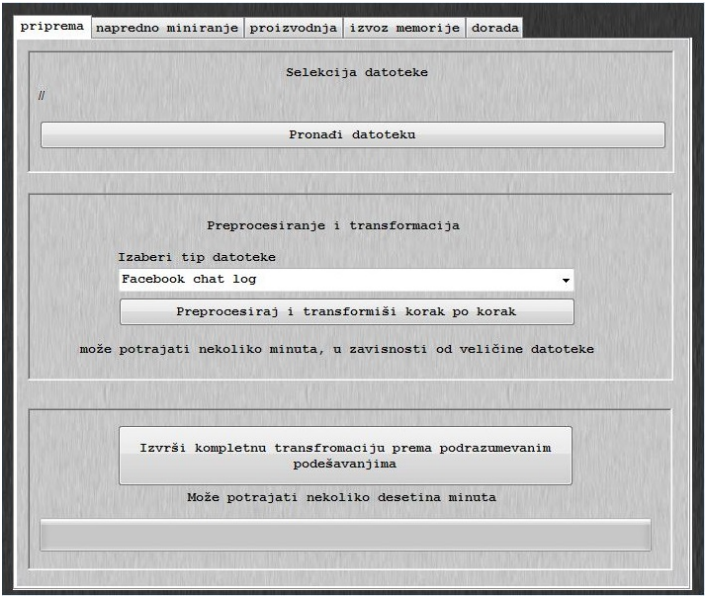
одређивач	значење	вредност	одређивач	значење	вредност
:))	смешак	0.56	:)	смешак	0.26
:D	осмех	0.91	:P	плажење	0.35
:p	плажење	0.24	xD	церење	0.59
:o	чуђење	-0.22	:O	чуђење	-0.29
:((	мрштење	-0.86	:(	мрштење	-0.66
:/	негодовање	-0.48	:\	негодовање	-0.48
:**	пољупци	0.88	:*	пољубац	0.80
hahaha	смејање	0.72	haha	смејање	0.15
<3	срце	1.00	;)	миг	0.26
:'(	плач	-0.74	lol	смејање	0.04
^^	радост	0.95	-.-	без речи	-0.45
:3	мачка	0.94	*.*	сјај у очима	0.95
:S	негодовање	-0.05	:'D	церекање	0.95
o.o	неверица	-0.10	...	без речи	-0.18

Табела 1. Коначна листа одређивача и њихових вредности.

3. Израда базе

За израду базе података задужен је био софтвер, направљен специјално за потребе овог истраживања. Написан је на програмском језику C# и може се покренути на *Windows* платформи, користећи било коју верзију овог оперативног система која ради на 64 бита. Корисничко окружење је предусретљиво и у потпуности на српском језику. Циљ при конципирању и изради овог софтвера био је да било који истраживач који говори српски може да га самостално користи, креира нове базе и употреби га за нова будућа истраживања.

Задатак софтвера је да улазну датотеку, која је у одговарајућем облику, кроз седам корака трансформише у базу података која садржи



Слика 2. Изглед корисничког окружења софтвера за истраживање података.

појмове који су у тој датотеци пронађени и њихове вредности на скали негативно-позитивно зависно од одређивача који се налазе у одговарајућој близини појма. У наредном поглављу биће објашњени кораци аутоматске израде базе података.

3.1 Припремна обрада

Припремна обрада обавља се над датотеком коју корисник сам бира притиском на дугме *Пронађи датотеку*. Тада се отвара *Windows Explorer* каталог у којем корисник на уобичајен начин бира датотеку са локалног рачунара. Алат тренутно подржава само датотеке које садрже историју дописивања корисника на друштвеној мрежи *Facebook*, и једини формат датотеке који корисник може да одабере је *htm*. Корисник наставља даље притиском на дугме *Препроцесирај и трансформиши корак по корак*[4] (слика 2).

[4] Алтернативно, може се користити дугме *Изврши комплетну трансформацију према подразумеваним подешавањима*, након чега

```
▼<html>
  ▼<head>
     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
     <title>Korisnik 1 - Messages</title>
     <link rel="stylesheet" href="../html/style.css" type="text/css"/>
   </head>
  ▼<body>
    ▼<div class="nav">
       <img src="../photos/profile.jpg"/>
      ▶<ul>...</ul>
     </div>
    ▼<div class="contents">
       <h1>Korisnik 1</h1>
      ▼<div>
        ▼<div class="thread">
           Korisnik 1, Korisnik 2
          ▼<div class="message">
            ▼<div class="message_header">
               <span class="user">Korisnik 1</span>
               <span class="meta">Tuesday, February 9, 2016 at 2:44pm UTC+01</span>
             </div>
            </div>
            <p>Tekst poruke</p>
          ▼<div class="message">
            ▼<div class="message_header">
               <span class="user">Korisnik 2</span>
               <span class="meta">Tuesday, February 9, 2016 at 2:39pm UTC+01</span>
             </div>
            </div>
            <p>Tekst poruke</p>
```

Слика 3. Одломак улазне датотеке.

Како софтвер користи XSL трансформације, основни предуслов је да његов улаз буде добро формирана XML датотека. Датотека коју је корисник одабрао (слика 3) најпре се очисти од свих знакова који могу сметати при провери њене добре формираности. То се обавља коришћењем функције *Regex.Replace* и регуларног израза *[\u0000-\u001F]*, који проналази у датотеци појављивање прва 32 карактера *ASCII* скупа и замењује их празном ниском. Ради додатног обезбеђивања добре формираности нове датотеке карактер *&* се такође замењује размаком и уводи се нови корени елемент.

Припремна обрада се завршава трансформацијом структуре датотеке из почетне у жељену, која је обрадива (слика 4). Овом

ће се извршити комплетна трансформација према подразумеваним подешавањима или подешавањима коришћеним приликом последње трансформације корак по корак.

Слика 4. Одломак датотеке након припремне обраде.

приликом се одстрањују сви чворови чија деца не садрже поруке који су корисници друштвене мреже разменили. Једини чвор који се чува је * * (слика 3) у којем су садржане поруке. Врши се преструктурирање ових чворова, нова датотека се чува и тиме се припремна обрада завршава.

3.2 Екстракција вредности из датотеке

Екстракција одређивача обавља се из два дела. Први део екстракције се састоји из задавања одређивача који ће се претражити и вредности које они означавају. Одређивачи и вредности се могу задати један по један, а овај корак се може и прескочити. Уколико се прескочи, софтвер ће аутоматски учитати подразумеване одређиваче и њихове вредности (табела 2.2).

У каталогу за напредно ископавање (text mining)(слика 5) изложени су сви подразумевани одређивачи и њихове вредности. Вредности се мењају ручно, путем текстуалног поља поред сваког од израза. Уколико је неки одређивач непожељан, његову вредност треба заменити карактером "/" и он неће бити коришћен приликом ископавања. У случају да корисник жели да тестира неки нови одређивач, може да користи дугме *додај емотикон* при дну странице, након што је попунио обавезна поља за израз и вредност. Уколико унета вредност одређивача није у опсегу од -1 до 1 програм је неће прихватити и корисник ће добити поруку о неуспешном додавању новог одређивача. Максимални број одређивача који се могу користити је 36.

Слика 5. Пример подешавања екстракције вредности из текста.

Напредни корисник може да измени текстуалну датотеку *data/emoticons.txt* која садржи подразумеване вредности одређивача. У сваком тренутку, одређивачи се могу вратити на подразумевана подешавања из поменуте датотеке, али се и нова подешавања могу преснимити преко постојећих. Обе опције се активирају притиском на одговарајуће дугмиће који се налазе на дну екрана (слика 5). У оба случаја, због могућег неповратног губитка података, корисник ће морати да потврди процес у додатном дијалогу који ће се појавити на екрану.

Након што је корисник задовољан подешавањима може прећи на други део екстракције који се обавља притиском на дугме *екстрактуј вредности из текста* на картици за производњу (слика 7). Пре притиска на ово дугме корисник може, али то није обавезно, да одабере поље *Користи предодређене регуларне изразе за побољшану претрагу*, што би требало да доведе до екстракције већег броја одређивача из текста.

Извршавање започиње провером обележености опционог поља. Уколико је обележено, текст пролази кроз низ *Regex.Replace* функција које проналазе унапред позната одступања од уобичајеног исписивања неколицине одређивача. Најпре се исправљају аутоматска превођења карактера у XML-у. Како овај део трансформације не захтева добро формиран XML документ, ентитетска референца *lt;* (ознака за почетак ентитетске референце *&* је приликом препроцесирања замењена размаком) замењује се карактером *<*, како би емотикон *<3* могао бити пронађен у тексту. Ентитетска референца *039* замењује се карактером *'*, како би емотикони попут *:'(* могли бити пронађени. Ниска *:-* замењује се самосталним карактером *:* , како би у ископавање били укључени и емотикони са *носем* типа *:-D* или *:-(*. Сва ћирилична слова која се користе у унапред одабраним одређивачима се транслитеришу у латиницу како би се ти емотикони пронашли и код корисника који користе ћирилицу (*:Д*, *хД*). Све варијанте емотикона *xD*, *:S* и *o.o* записане великим и малим словима, претварају се у основни формат. Почеци хиперлинкова *http://* и *https://* замењују су неутралном карактерском ниском *yy*, како не би били погрешно протумачени као емотикон *:/* . Коначно, регуларни изрази *[a|h|A|H][a|h|A|H][h|H][a|A][h|H][a|h|A|H]* и *[h|H][a|A][h|H][a|A]* се користе како би било пронађено што више различитих примера изражавања смеха, а пронађени изрази се замењују са *hahaha* односно *haha*.

Уколико опција чишћења и уједначавања није укључена, овај корак се прескаче и биће пронађени и обрађени само одређивачи у свом основном облику који је наведен на картици напредног ископавања.

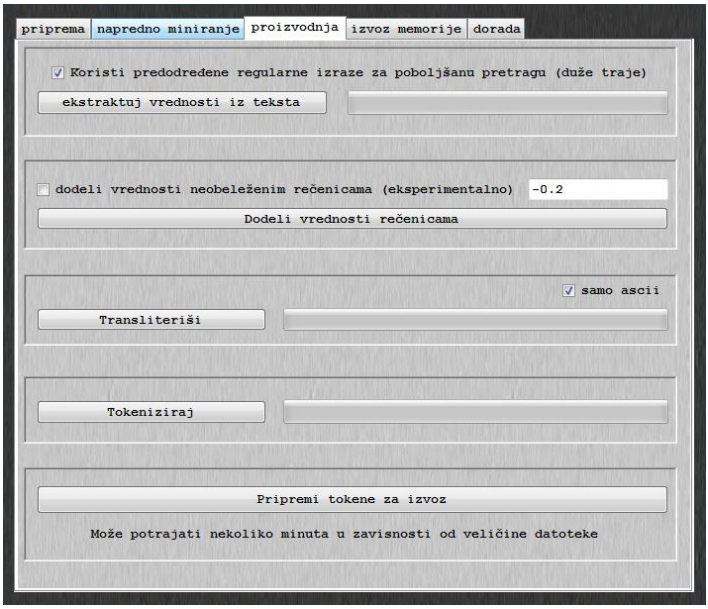
Сама екстракција се састоји од учитавања одабраних израза и њихових вредности са картице напредног ископавања у два низа и потом њихове обраде. Сваки од елемената првог низа, низа израза, пролази кроз *for each* петљу у којој се свако његово понављање у тексту замењује етикетом *<emot value='x'/>*, која представља један празан XML чвор у коме је x вредност текућег одређивача која је преузета из другог низа, низа вредности. Уколико је вредност израза */* инструкције неће бити извршене, а уколико вредност неког израза није између -1 и 1 програм ће пријавити грешку и прекинути извршавање.

Након што је сваки од одређивача замењен чвором чији је атрибут његова вредност (слика 6), XML документ поново постаје добро формиран и спреман за даљу обраду. Корисник ће добити поруку да је екстракција завршена и да може да пређе на следећи корак.

```
▼<message>
   <sender>Korisnik 1</sender>
   <text>tekst :)</text>
 </message>

▼<message>
   <sender>Korisnik 1</sender>
   <text>teskt <emot value="0.26"></text>
 </message>
```

Слика 6. Изглед поруке пре и након екстракције одређивача.



Слика 7. Картица за производњу базе података корак по корак.

3.3 Додела вредности сегментима текста

У експерименту сегмент текста или реченица поистовећена је са једном самосталном поруком коју је један корисник упутио другом.

Основна идеја је да се вредности одређивача пронађених у поруци одразе на вредност саме поруке, или околних порука у специјалним случајевима. У зависности од текстуалног садржаја поруке (осим самог одређивача[5]) и садржаја претходне и наредне поруке постоји три врсте доделе вредности сегментима:

1. уколико порука поред одређивача садржи текст и порука која је следи садржи текст – на поруку се односе само одређивачи из те поруке. Примери:
   *А: Срећан рођендан :**
   Б: Хвала пуно
   На поруку особе А односиће се одређивач *:**, док порука особе Б на њу нема утицај јер не садржи одређивач. *А: Срећан рођендан :**
   Б: Хвала :)
   На поруку особе А односиће се одређивач *:**, док порука особе Б на њу нема утицај јер се одређивач у поруци особе Б односи само на текст поруке особе Б.

2. уколико порука садржи тескт али не и одређивач, а наредна порука садржи одређивач, али не и текст – одређивач ће се односити на тескт поруке која претходи. Пример:
   А: Аутобус ми је побегао
   Б: :(
   Одређивач *:(* из поруке особе Б односиће се на поруку особе А јер порука особе Б не садржи текст, те се њен одређивач односи на поруку особе А, која јој претходи у разговору.

3. уколико порука садржи и одређивач и текст, а наредна порука садржи одређивач, али не и текст – одређивачи из обе поруке се односе на поруку која садржи текст. Пример:
   А: Аутобус ми је побегао -.-
   Б: :(
   На поруку особе А односиће се и одређивач -.- и одређивач *:(*, јер порука особе Б не садржи текст, те се њен одређивач односи на поруку особе А, која претходи у разговору.

[5] У овом случају све поруке које садрже мање од 4 карактера третирају се као празне.

Постоји и четврта могућност – додела вредности поруци која не садржи одређивач, при чему ни следећа порука не садржи одређивач. Ова могућност се заснива на претпоставци да и одсуство одређивача само по себи носи неко значење (према интуицији негативно). Недостатак ове могућности је да не можемо бити сигурни да непостојање одређивача носи (негативно) значење, док је одређивање саме вредности тежак задатак. Додатни проблем је и то да у поруци можда и постоји одређивач који кориснику није познат па га није користио при ископавању тескта. Стога је ова опција необавезна и њена вредност није дефинисана. Уколико корисник жели да зада одређену вредност *необележеним* порукама, то може да уради одабиром поља *додели вредност необележеним реченицама* и исписивањем вредности између -1 и 1 у текстуално поље пре покретања саме доделе (слика 7).

```
▼<root>
  ▼<conversation>
    ▼<messages>
      ▼<message>
          <sender>Korisnik 1</sender>
          <text>tekst poruke 1 <emot value="0.26"><emot value="-0.26"></text>
        </message>
      ▼<message>
          <sender>Korisnik 2</sender>
          <text><emot value="0.15"></text>
        </message>
```

```
▼<root>
   <emotext value="0.15">tekst poruke 1</emotext>
```

Слика 8. Изглед одломка документа пре и после доделе вредности сегментима када и наредна порука садржи одређивач, али не и текст (случај 3).

Сама додела вредности извршава се путем XSLT трансформације. Најпре се за сваку *обележену* поруку рачуна аритметичка средина вредности свих одређивача коју се на њу односе, затим се та вредност уписује у њен чвор у нови атрибут *value*, а чворови који су обележавали одређиваче се бришу. Потом, уколико је опционо поље било обележено, осталим порукама се додаје задата вредност. Резултат трансформације

је XML документ који садржи корени елемент и у њему чворове свих порука које имају додељену вредност. По завршетку трансформације корисник ће добити поруку да је додела завршена и зелено светло за наставак даље.

3.4 Транслитерација и сравњивање

Ова два корака обављају се заједно притиском на дугме *транслитериши* картице за производњу. Сврха овог дела програма је да се добије чист текст за прављење базе. У овом случају текстом се сматрају ниске које садрже искључиво бројеве и латинична слова. Необавезно поље за избор *само ascii*, служи да се скуп додатно ограничи само на латинична слова из ASCII скупа карактера (бројеви се овом опцијом не изузимају).

Разлог због којег је превођење у ASCII необавезно је што се њиме уводи готово исто толико проблема колико се и решава. Под претпоставком да сви корисници друштвених мрежа не користе у пуној мери капацитет Unicode каракатерског скупа, већ и *деградирану латиницу*,[6] долази до проблема јер неке речи попут *истраживање* и *истразивање* неће бити препознате као иста, већ као две различите речи. Превођењем свих карактера у ASCII скуп овај проблем се избегава али се уводи нови проблем да речи које би се заправо разликовале у Unicode скупу буду препознате као иста реч – пример могу да буду речи *шпанац* и *спанаћ*, које ће програм препознати као исте речи. Ову опцију је, дакле, најбоље користити зависно од ситуације.

Пре даљег наставка обраде, тренутни XML документ пролази кроз XSL трансформацију у којој се помоћу функције *translate* сва велика слова претварају у мала, како програм не би у коначној бази правио разлику између две исте речи, које се разликују једино у величини слова.

Други део сређивања текста, чишћење свих карактера који нису слова или цифре, отклања могућност да реч садржи неки интерпукцијски или други неалфанумерички карактер. Поново се функцијом *translate* XSL трансформације сви нежељени карактери (сви карактери осим малих латиничних слова и цифара) претварају у размак. И овим поступком могу се добити некоректне речи ако је корисник случајно унео непожељни карактер у току куцања речи (*Мар?ија*) или

[6] Латинична слова без дијакритичких знакова, на пример: *c*, *z*, *s* уместо *č*, *ž*, *š*.

ако је намерно користио неки специјални карактер (*M@puja*). У првом случају ниска ће постати *Мар ија*, а у другом *M puja*.

3.5 Токенизација

Токенизација текста обавља се притиском на дугме *токенизација* картице за производ (слика 7). Подела на токене се врши искључиво према размаку, па се токен дефинише као било која ниска карактера између почетка поруке и првог размака у поруци, било која ниска карактера између два суседна размака или било која ниска карактера између последњег разамака у поруци и краја поруке. Она се у овом случају врши на нивоу XML документа, тако што сваки токен добија сопствени XML чвор. Први део процеса токенизације обавља се функцијом *Regex.Replace* у три корака:

- Најпре се проналази сваки почетак поруке у XML документу. То се обавља претрагом ниске *″>*, која се јавља једино на крају отворене етикете елемента који садржи атрибут, а то је елемент који садржи текстуалну поруку. Како корени елемент не садржи атрибут, он неће бити захваћен овом претрагом. Затим се после сваке пронађене ниске додаје ниска *<token>*, која означава почетак токена.
- Потом се проналази крај сваке поруке, претрагом ниске *</emotext>* што је етикета затварања елемента поруке. Испред сваке овакве пронађене ниске додаје се *</token>*, како би се добила ниска *</token></emotext>*. Овим кораком свака порука у XML документу постаје један токен: *<emotext value=″x″><token> ... </token></emotext>* .
- Последњи корак је подела унутрашњости сваке поруке на токене, на сваком месту где се налази размак. Како не би дошло до нежељене замене унутар отворене етикете *<emotext value=″x″>*, на месту између назива елемента и назива атрибута, најпре се обавља замена сваког *emotext value* са *emotext_ value*, а тек онда замена сваког размака са *</token><token>*. Након тога се карактер _ поново враћа у размак и добијају се поруке у облику: *<emotext value=″x″><token> ... </token><token> ... </token></emotext>* .

Други део токенизације обавља се помоћу четири додатне XSL трансформације које филтрирају одабране токене и дају им вредност:

– Прва трансформација брише све поруке које садрже више од четрдесет токена, како би се избегла додела вредности свим речима у, на пример, неком тексту налепљеном уз ћаскање, а који није активна порука у конверзацији.
– Друга трансформација брише све токене који садрже мање од два карактера (под претпоставком да не носе значење јер представљају нефункционалне речи или су настали разбијањем великих ниски попут линкова), и све токене који садрже више од двадесет карактера (под претпоставком да је велика већина њих насумична).

Између ове две трансформације потребно је спровести још један корак, увођење додатног токена без вредности на последње место у документу. То се ради уметањем ниске *<token>ERR0001</token>* пре затворене етикете кореног елемента, како би се добило *<token>ERR0001</token>*. Значење ове ниске биће објашњено у наредном одељку.

– Трећа трансформација служи за осамостаљивање токена. Она најпре сваком токену додељује атрибут са вредношћу поруке у којој се налази, а затим брише елементе реченица, остављајући у кореном чвору само чворове токена са придруженим вредностима. Она такође сортитра све токене по абецедном реду у односу на њихов текстуални садржај и додељује сваком токену нови атрибут *no*, чија вредност постаје редни број токена у документу, одређен XSL функцијом *position*.

Слика 9. Изглед одломка документа пре и након токенизације.

– Четврта трансформација не доприноси садржају документа већ само економичности његове обраде. Име сваког елемента *token* се скраћује на *t*, сваког атрибута *value* на *v*, што у великој мери смањује величину документа и олакшава и убрзава даљи рад са датотеком.

3.6 Израда индекса токена са вредностима

Овај корак се обавља притиском на дугме *припреми токене за извоз* у задњем пољу картице за производњу (слика 7). Извршење ове команде могуће је тек након што су извршени сви претходни кораци припреме.

Како би индекс послужио сврси, сви појмови у њему добијају два атрибута: средња вредност свих одређивача који се на тај појам односе у учитаном корпусу и број понављања тог појма у њему. Креирање индекса обавља се у шест корака XSL трансформацијом XML документа који у себи садржи токене:

- У бази уређеној у абецедном поретку се проналази први чвор, x, и први следећи чвор који се од њега разликује, y.
- Узимају се њихови редни бројеви токена из атрибута *no*. Редни број чвора x је n, а редни број чвора y је m.
- Текст чвора x се уписује у нови XML документ инвертованог индекса у елемент t (token).
- Новом елементу t додељује се атрибут c (count), који означава број понављања тог токена у бази, а његова вредност је једнака разлици редних бројева два пронађена чвора.

$$c = m - n \tag{2}$$

- Новом елементу t додељује се атрибут v (value), који означава вредност тог токена на скали позитивно-негативно, и она је једнака аритметичкој средини вредности из атрибута v свих токена који садрже исти текст.

$$v_x = \frac{\sum_{i=n}^{n+c-1} v_i}{c_x} \tag{3}$$

- Чвор y постаје први чвор у документу и процедура креће из почетка.

Овим поступком атрибути се додељују свим токенима, сем последњем, помоћном, *ERR0001* токену, који је створен приликом токенизације документа и служи за обележавање завршетка документа и за израчунавање атрибута c код токена који се налази испред њега.

Након што је успешно извршено претварање скупа токена са вредностима у коначан инвертовани индекс појмова са вредностима (табела 3.6), корисник ће добити поруку о успешно извршеној трансформацији, а индекс ће бити сачуван у привременој датотеци док се не изврши његов коначан извоз.

t (појам)	v (вредност)	no (редни број)
zdravko	-0.1	859231
zdravko	0.3	859232
zdravlje	0.26	859233

t (појам)	v (средња вредност)	c (број понављања)
zdravko	0.2	2
zdravlje	0.26	1

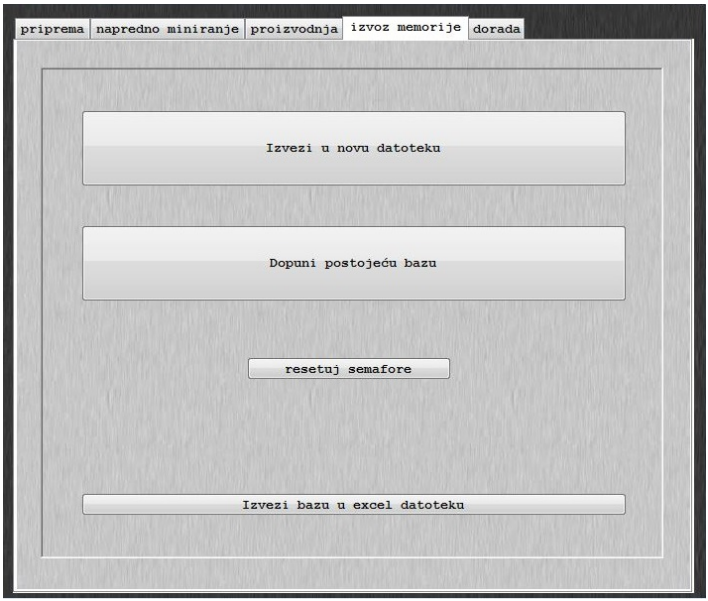
Табела 2. Пример изгледа дела базе података пре и након коначне трансформације.

4. Извоз и допуна базе података

Притисак на дугме *Извези у нову датотеку* картице за извоз меморије (слика 10) отвара *Windows Explorer* каталог за чување нове датотеке. Формат у коме ће датотека бити сачувана је XML, а корисник бира њено име и место. Садржај се копира из привремене датотеке креиране у претходном кораку, и представља коначну базу података.

У случају да је корисник већ обрађивао неке датотеке, на располагању му је и опција *Допуни постојећу базу* (слика 10). Она омогућава да се подацима извученим из новог корпуса допуни претходно извезена база података. У овом случају, притиском на одговарајуће дугме отвара се *Windows Explorer* каталог за отварање нове датотеке. Корисник треба да обележи датотеку коју жели да допуни, а затим се креира допуњена база у четири корака:

– Изабрана датотека се у потпуности учитава у меморију и додаје на садржај претходно креираног индекса сачуваног у привременој датотеци.
– Проналази се и брише ниска *</root><root>*. Тиме се добија документ са једним, уместо два корена елемента.
– XSL трансформацијом се све токени поново сортирају по абецедном реду.
– Како у овом случају могу постојати највише два токена са истим текстом, њихова унификација је једноставнија. Уколико токен нема парњака, преписује се, а уколико има добија само нове вредности атрибута. Атрибут вредности се израчунава као количник збира производа атрибута v и c оба елемента и укупног броја њихових



Слика 10. Изглед картице за извоз/допуну базе података.

понављања (2.4). Атрибут понављања једнак је укупном броју њихових понављања (2.5).

$$v = \frac{v_1 * c_1 + v_2 * c_2}{c_1 + c_2} \tag{4}$$

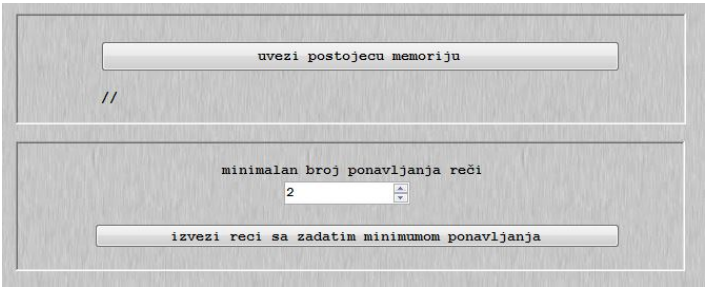
$$c = c_1 + c_2 \tag{5}$$

Нови елемент са атрибутима се чува док се стари бришу из индекса, и тиме се добија по једна копија сваког токена са правилно одређеним вредностима атрибута.

Нови документ замењује онај који је обележен у каталогу за отварање, те је ово опција коју треба употребљавати пажљиво. По завршетку, корисник ће добити поруку о успешно извршеном допуњавању.

Уколико корисник жели да ручно прегледа новостворену базу, а не осећа се пријатно у XML окружењу, може да је извезе у CSV (engl.

comma separated values – вредности раздвојене зарезом) формат који може да се отвори помоћу *Microsoft Excel*-а, *Open Office Calc*-а или сличног софтвера за рад са табелама. Он то чини притиском на последње четврто дугме на картици за извоз меморије.

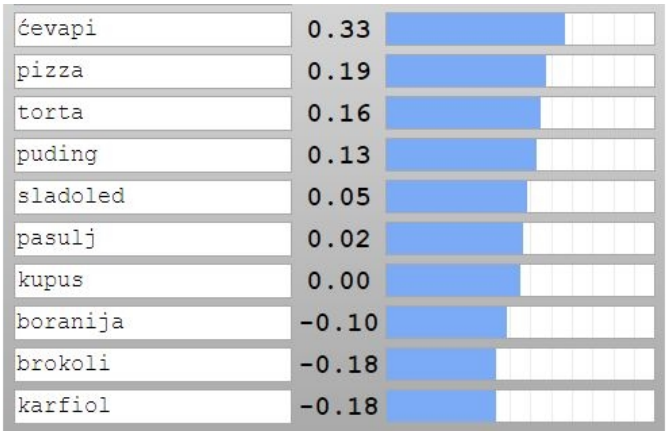


Слика 11. Изглед одељка за дораду базе података

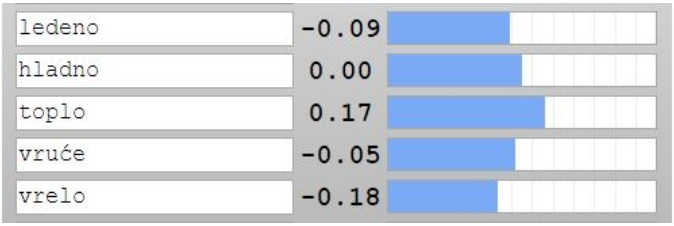
База података се може додатно проредити на основу тога колико често се појмови понављају. Ова опција се налази на петој картици софтвера, експерименталној картици за дораду. Корисник најпре треба да учита базу података коју жели да доради притиском на дугме *увези постојећу меморију*, а затим бира колико барем пута појам треба да се понавља да би био узет у обзир (слика 11).

Уколико корисник, на пример, одабере број 2, притиском на дугме *извези речи са задатим минимумом понављања* формираће се нова база из које ће бити избрисани сви појмови који се понављају мање од 2 пута (тј. појављују се само једном), и нова база ће бити сачувана на жељеном месту. Ову опцију ће корисник користити уколико сматра да је потребно да се појам појави одређени број пута како би његова вредност могла да се сматра репрезентативном.

Након што је корисник завршио креирање жељених база и успешно их извезао, може их тестирати ручно уз помоћ претраге уграђене у било који софтвер за рад са текстом или XML документима. Такође, база се у било којем од својих облика може тестирати и у софтверу за проверу вредности прављеним специјално за овај експеримент (слике 12 и 13).



Слика 12. Пример поређења вредности појмова из базе података коришћењем веб-апликације – брза храна, дезерти, поврће.



Слика 13. Пример поређења вредности појмова из базе података коришћењем веб-апликације – температуре.

5. Закључак

5.1 Осврт на резултат експеримента

Упркос релативно малом коришћеном узорку од 1.843.826 порука, од грубо процењених 500 милијарди порука које припадају корисницима друштвене мреже *Facebook* у Србији, експеримент је испунио своје циљеве. На основу екстрахованих свега двадесет и осам одређивачких ниски уз помоћ седам регуларних израза који проналазе њихове варијације, и без претходног знања о језику и његовој граматици,

софтвер је формирао базе података са појмовима и њиховим вредностима на скали позитивно-негативно.

Систем је тестирало неколико независних оцењивача и на основу њихових реакција може се закључити да су вредности ненасумичне и репрезентативне, што показује да је овај начин класификовања појмова могућ у било којем природном језику који користи одређиваче, али је и даље неопходно спровести системску евалуацију.

Коришћени алгоритми су и даље далеко од савршених и у њима има доста простора за побољшавање и напредак. Добра документованост процеса истраживања података требало би да допринесе будућим истраживањима исте или сличних идеја. Следећи корак свакако може да буде екстраховање проширеног скупа одређивача, рад над проширеним скупом текстова или, идеално, проширени скуп параметара који би се вредновали. Корпус проширен, на пример, на друге природне језике умногоме би допринео обиму појмова којима се вредност додељује, док би проширени скуп параметара могао да дода нову дубину разумевања текста од стране како машине тако и људи.

5.2 Могућност примене

Овај начин екстракције одређивача и уопште сличног истраживања података може наћи разноврсне примене које се могу поделити у две групе.

Друштвена и демографска истраживања:

– Маркетиншка истаживања: истаживање деловања тренутног или алтернативног приступа рекламирању производа и услуга. Ово је најчешћа употреба сличних истраживања пре свега из новчаних разлога, јер компанијама може да уштеди новац или да им обезбеди нови прилив добара.
– Истраживање јавног мњења: шта људи воле, шта не воле, какво уопште мишљење имају о стварима или идејама на које се појмови или текст односе. Може се употребити на различите начине у друштвеним и демографским истраживањама за брзо и ефикасно прикупљање велике количине информација.

Развијање интелигентних система за рад са информацијама:

– Проналажење информација: проналажење специфичних информација у тексту, као и проналажење информација које

се не могу прецизно дефинисати. Класификација текстова према расположењу које је у њима изражено ради лакшег проналажења потребних информација.
– Анализа и разумевање природног језика: разумевање писаног текста и текстуалних упита, анализа расположења које текст носи, рад са дигиталним језичким ресурсима попут аутоматизоване паралелизације или аутоматизација било које операције која захтева дубоко разумевање писаног текста.
– Вештачка интелигенција: аутоматизовање конверзације на природном језику, одржавање разговора и рад са странкама.

5.3 Могући недостаци и побољшања

– Проблем могућих насумичних одговора приликом анкетирања:
Направити систем који одбацује одговоре корисника који су одговарали насумично, на пример, увођењем додатних питања која у тексту захтевају специфичне одговоре. Тако немарни корисници који не читају питања имају веће шансе да буду идентификовани.
– Системска евалуација резултата:
Спровести детаљну системску евалуацију како би се утврдила веродостојност добијених резултата.

Литература

Davidov, Dmitry, Oren Tsur и Ari Rappoport. "Enhanced Sentiment Learning Using Twitter Hashtags and Smileys". У *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, Association for Computational Linguistics, 2010, 241–249.

Go, Alec, Richa Bhayani и Lei Huang. "Twitter Sentiment Classification Using Distant Supervision". *CS224N Project Report, Stanford* Vol. 1 (2009).

Mishne, Gilad. "Experiments with mood classification in blog posts". У *Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access*, Vol. 19, Citeseer, 2005, 321–327.

Neviarouskaya, Alena, Helmut Prendinger и Mitsuru Ishizuka. "Compositionality Principle in Recognition of Fine-Grained Emotions from Text". У *Proceedings of the Third International ICWSM Conferenc*. The AAAI Press, 2009, 278–281.

Ptaszynski, Michael, Pawel Dybala, Rafal Rzepka и Kenji Araki. “Towards Fully Automatic Emoticon Analysis System”. У *Proceedings of the Sixteenth Annual Meeting of the Association for Natural Language Processing*, 2010, 223–228.

Ptaszynski, Michael, Rafal Rzepka, Kenji Araki и Yoshio Momouchi. “Research on Emoticons: Review of the Field and Proposal of Research Framework”. У *The Seventeenth Annual Meeting of The Association for Natural Language Processing*. The Association for Natural Language Processing, 2011, 1159–1162.