Н-грамски заснована класификација текста на српском језику применом методе структуралних подржавајућих вектора

УДК 811.163.41'322.2

САЖЕТАК: У раду су представљени резултати класификације хијерархијски организованог корпуса докумената на српском језику коришћењем методе подржавајућих вектора (МПВ, енгл. *Support Vector Machine, SVM*). Примењене су две технике класификације изведене из методе МПВ са структурним излазом: вишекласна равна (енгл. *flat*) и хијерархијска класификација. Модел заједничке репрезентације документа и класе или хијерархије класа којима документ припада, специфичан за овај облик МПВ методе, базиран је на н-грамима бајтова различите дужине. Коришћене су четири tf-idf статистике које одређују значајност н-грама за одређени документ. Описане технике и статистике тестиране су на хијерархијски структуираном подскупу Ебарт корпуса новинских текстова. Добијени резултати за оба типа класификатора на нивоу целог корпуса су приближни, док на нивоу појединачних класа хијерархијски тип класификатора показује боље резултате за већину класа са малим бројем текстова.

КЉУЧНЕ РЕЧИ: хијерархијска класификација текста, метода подржавајућих вектора, Ебарт корпус.

РАД ПРИМЉЕН: 22. мај 2015.
РАД ПРИХВАЋЕН: 30. октобар 2015.

Јована Ковачевић
jovana@matf.bg.ac.rs

Јелена Граовац
jgraovac@matf.bg.ac.rs

Универзитет у Београду
Математички факултет
Катедра за рачунарство
и информатику

1. Увод

Класификација текста је један од задатака истраживања текста (енгл. *text mining*) – области рачунарске лингвистике која обухвата скуп техника за издвајање корисне, скривене, претходно непознате информације из текстуалних

докумената. У случају класификације текста скривена информација је припадност текста, према садржају, једној или већем броју класа из предефинисаног скупа класа (Manning and Schütze, 1999). Класификација може да се спроведе ручно, али је тај поступак временски захтеван и скуп. С обзиром на широку распрострањеност брзих рачунара, аутоматска класификација постала је основ за ефикасну обраду великих колекција докумената и откривање знања садржаног у њима.

У аутоматској или полуаутоматској класификацији текста углавном се користе два различита приступа: приступ заснован на лексичко-семантичким језичким ресурсима и приступ заснован на машинском учењу. Системи прве врсте користе лексичко-семантичке мреже, као што су WordNet (Miller, 1995) и FrameNet (Johnson et al.), уз ресурсе и алате какви су електронски речници и лексикон граматике (Gross, 1997), семантичке онтологије, онтологије именованих ентитета и властитих имена. Ови језички ресурси омогућују развој модела класификације за морфолошки и деривационо изузетно богат језик какав је српски. Класификатор се најчешће ручно гради на основу богатих морфолошких, синтаксичких и семантичких информација садржаних у лексичким ресурсима (Scott and Matwin, 1998) и не захтева се постојање скупа класификованих текстова који би се користили за учење класификатора. За разлику од ових, системи који користе други приступ подразумевају постојање класификованих корпуса текстова подељених на скуп за учење и скуп за тестирање. На основу података за учење граде се класификатори (класификациони модели, функције припадности класи) применом разних статистичких метода, нпр. бајесовске класификације (Eyheramendy et al., 2003), условних случајних поља (McCallum and Pereira, 2001), скривених Марковљевих модела (Yi and Beheshti, 2013), или метода заснованих на подржавајућим векторима, неуронским мрежама (Sebastiani, 2002), најближим суседима (Yang and Pedersen, 1997), дрветима одлучивања (Quinlan, 1996), итд. Посебно је значајан вишејезични EuroVoc класификатор JEX (Steinberger et al., 2013) који се састоји од обучених класификатора за 22 различита језика Европске уније.

Одабир методе, па и самог приступа проблему класификације текста зависи од два кључна фактора: доступности језичких ресурса и доступности података за обучавање. Уколико језички ресурси, као што су лексикони, речници и граматике, семантичке мреже постоје, има смисла користити приступ заснован на њима. Овај приступ мора да узме у обзир карактеристике и специфичности сваког језика на који се примењује, па тако у случају српског језика, чињеницу да користи два алфабета (ћирилични и латинични), да је правопис фонолошки заснован, да је морфолошки систем богат, ред речи у реченици слободан, систем слагања веома сложен (Vitas et al., 2003). Све ове карактеристике чине да су кораци претходне обраде, као што су избор и издвајање својстава на основу

којих се врши класификација (енгл. *feature selection, feature extraction*) доста сложени. Уколико корпуси потребни за обучавање алгоритама – у случају класификације текстова то су базе класификованих текстова – постоје или их је лако и јефтино направити, погодно је применити приступ заснован на машинском учењу. Техникама машинског учења класификатор се генерише аутоматски, „учењем“ карактеристика класа на основу скупа података за учење придружених свакој класи. То су подаци који су ручно класификовани у класе од стране експерта из домена. Након процеса подучавања (тренирања, учења), класификатор најчешће аутоматски генерише скуп правила која треба да задовољава податак да би био класификован у одређену класу.

У зависности од броја класа, класификација може бити бинарна, када су дефинисане само две могуће класе или вишекласна, када је дефинисано више могућих класа. У зависности од тога да ли се класе могу преклапати или не, класификација може бити једнозначна (енгл. *single-label*), када једном податку може бити додељена тачно једна класа или вишезначна (енгл. *multi-label*), када једном податку може бити додељена једна, ни једна или више класа, односно класе се могу преклапати. Према структури која дефинише односе међу класама, класификација може бити хијерархијска или нехијерархијска. Уколико се у току процеса класификације класе посматрају самостално без икакве структуре која дефинише односе између њих, тада се ради о нехијерархијској класификацији. Када број различитих класа, или број података унутар једне класе постане јако велики, јављају се проблеми тачног и ефикасног претраживања и управљања подацима на нивоу једне класе. У том случају, класе се најчешће организују у стаблолике структуре и уводи се хијерахијска структура међу њима (Sun and Lim, 2001) (нпр. Yahoo хијерархија).

У (Graovac, 2013) и (Pavlović-Lažetić and Graovac, 2010) приказана је метода класификације докумената на српском језику заснована на српском WordNet-у (Krstev et al., 2004) развијеном за српски језик у оквиру Групе за језичке технологије на Математичком факултету Универзитета у Београду[1] и примењена је на корпус новинских текстова Ебарт[2]. Коришћени су и други језички ресурси развијени за српски језик у оквиру Групе – електронски речник (Vitas and Krstev, 2005), лексикон граматике (Vitas et al., 2003), онтологија властитих имена (Krstev et al., 2005).

У радовима (Graovac et al., 2015; Graovac, 2014a; Graovac and Pavlović-Lažetić, 2014; Graovac, 2014b) приказане су методе класификације засноване на машинском учењу које користе н-грамску методу за представљање текста и методу κ најближих суседа (енгл. *k nearest neighbors, kNN*) за изградњу класификатора. Методе су језички независне, примењене су на корпусе текстова

[1] www.matf.bg.ac.rs/$\sim$cvetana/LT-pregled.html
[2] http://www.arhiv.rs/novinska-arhiva/

најраспрострањенијих писама и језика, различитих лексичких, морфолошких, синтаксичких и правописних карактеристика (енглески, кинески, арапски, шпански), веома су једноставне и показале су веома добре резултате. Посебно, у (Graovac, 2012) приказана је примена н-грамске методе машинског учења на класификацију текстова на српском језику и корпусу новинских текстова Ебарт. Класификација је вишекласна, вишезначна и нехијерархијска. У овом раду по први пут је примењена метода структуралних подржавајућих вектора на класификацију текстова на српском језику (Ебарт корпус). Над „равним“ (енгл. *flat*) корпусом дефинисана је хијерархијска структура класа а из самог корпуса издвојен је подскуп докумената који по садржају одговарају тим класама. Затим су над овим хијерархијски организованим корпусом примењене две технике класификације, изведене из методе МПВ са структуралним излазом: вишекласна (равна, flat) класификација (избор једне од већег броја класа) и хијерархијска класификација (избор хијерархије класа којима документ припада). Модел заједничке репрезентације документа и класе или хијерархије класа којима документ припада, специфичан за овај облик МПВ методе, базиран је на н-грамима бајтова различите дужине.

Најзад, на основу података за тестирање евалуиран је модел и одређена је тачност резултата коришћењем једне од стандардних мера у области претраживања информација – Ф1 мере која комбинује одзив (енгл. *recall*) и прецизност (енгл. *precision*) (Tan et al., 2006).

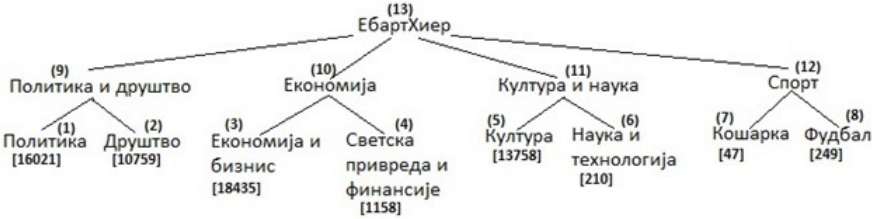
У даљем тексту рада биће прво приказан Ебарт корпус – највећа дигитална медијска документација у Србији, као и хијерархијски организован подкорпус издвојен из Ебарт корпуса за потребе тестирања метода хијерархијске класификације (део 2.).

Затим ће бити приказана у основним цртама коришћена методологија (део 3.): метода МПВ са структурним излазом и њена прилагођавања за примену на вишекласну и хијерархијску класификацију (тачка 3.1), концепт н-грама бајтова (тачка 3.2) и специфични н-грамски начин заједничког представљања документа и класе (односно хијерархије класа) којој документ припада као и кораци учења и тестирања у примени ове методе (тачка 3.3). Мере евалуације уведене су у тачки 3.4.

Главни резултат рада – резултат класификације текстова – биће приказан у делу 4.. Биће приказани и резултати евалуације и однос према сродним (упоредивим) резултатима. Најзад, у делу 5. интерпретираћемо и дискутовати добијене резултате, утицај примењених метода и могућности усавршавања.

2. Подаци (Dataset)

Ебарт (www.arhiv.rs/novinska-arhiva/) представља највећи архив новинских текстова савременог српског језика у дигиталном облику. Он постоји од 2003. године и до данас је у њему ускладиштено више од 2.000.000 текстова из штампаних медија. Актуелна архива је класификована на тематске целине по угледу на уобичајене новинске рубрике: политика, спољна политика, друштво, економија, хроника, култура, забава, спорт, медији, фељтон, писма читалаца, и друго. У циљу тестирања методе подржавајућих вектора на проблему хијерархијске класификације текстова, из „равног“ Ебарт корпуса је издвојен хијерархијски организован подкорпус који смо назвали ЕбартХиер. ЕбартХиер обухвата све чланке из дневног листа Политика, објављене од 2003. до 2006. године, који припадају рубрикама Политика, Друштво, Економија и бизнис, Светска привреда и финансије, Култура, Наука и технологија, као и чланке из листа Спортски журнал (објављене од 2003. до 2006) који припадају рубрикама Кошарка и Фудбал. Сви документи који припадају по тематици сличним рубрикама/класама, груписани су у заједничку класу на вишем нивоу хијерархије. Тако су настале класе Политика и друштво, Економија, Култура и наука и Спорт. Овако добијени корпус се карактерише дрволиком структуром приказаном на Слици 1.



Слика 1. Приказ дрволике структуре ЕбартХиер корпуса. Сваком имену класе придружен је број којим се та класа нумерише. У листовима, у угластим заградама, приказан је број докумената у свакој класи

.

Класе су непреклапајуће (један документ може припадати само једној класи) и сваки документ се може класификовати само у класу која припада листу хијерархије. Корпус се карактерише изразито неравномерном расподелом докумената по класама (видети Слику 1). Средња вредност дужине свих докумената у корпусу је 327,75 речи у документу (најкраћи документ има 7

а најдужи 2576 речи). На Слици 2 је приказан хистограм средњих вредности дужина докумената (као и дужине најкраћег и најдужег документа) по свим класама. Сви документи су представљени латиничним писмом коришћењем UTF-8 кодне шеме.

3. Методологија

3.1 Метод структуралних подржавајућих вектора

Метода подржавајућих вектора (МПВ) се показала веома ефикасном у класификацији текстова (Joachims, 1998). У овом раду биће примењена метода структуралних подржавајућих вектора (МСПВ, (Tsochantaridis et al., 2004)) која представља генерализацију МПВ методе на структурални излаз, који може бити у облику низа, стабла, усмереног ацикличког графа итд.

Слика 2. Хистограм средњих дужина докумената у класама. У заградама су додатно приказане дужине најкраћег и најдужег документа у свакој класи

Пре него што представимо МСПВ методу, биће дат преглед основне МПВ методе за бинарну класификацију из угла класификације текстова. Стандардни приступ у обучавању предиктора за бинарну класификацију је учење дискриминантне функције (енгл. *Discriminant function*) $F(x)$ и класификовање улазног вектора x на основу знака функције $F(x)$. С обзиром да линеарне методе обично имају ефикасне алгоритме за обучавање, уобичајено је

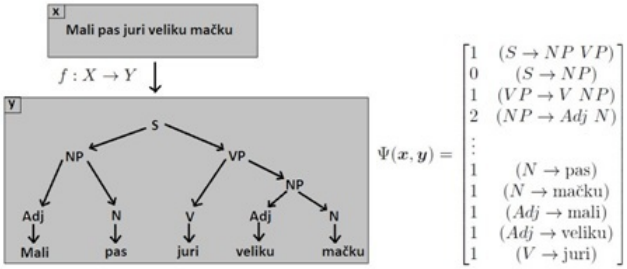
претпоставити да је функција $F(x)$ линеарна, односно да се може представити у облику $F(x) = \langle \omega, x \rangle$, где је ω вектор параметара који се уче а „$\langle \dots \rangle$“ ознака за скаларни производ. Улазни вектор x може помоћу неке функције Ψ бити пресликан у други простор и у том случају дискриминантну функцију записујемо $F(x) = \langle \omega, \Psi(x) \rangle$.

Бинарни класификатор може да предвиди да ли дати текст припада одређеној класи или не, што значи да његов излаз може бити -1, уколико текст не припада датој класи, или 1, у супротном. Уколико желимо да знамо којој класи текст припада из датог скупа класа, окрећемо се методама које предвиђају структурални излаз, конкретно МСПВ. У основној МПВ методи закључивање о излазу y за дати улаз x се врши на основу знака дискриминаторне функције, односно $sgn(F(x)) = y$, где је $y \in \{-1, 1\}$. Било би идеално када бисмо и за структурални случај могли да пронађемо функцију $F(x)$ која тачно пресликава скуп улазних података (у овом случају текстова) у скуп излазних података (у овом случају класа). С обзиром да је веома тешко конструисати баш такву функцију, идеја је конструисати функцију $F : X \times Y \to \mathbb{R}$ која би мерила колико добро излаз у одговара улазу x.

Желимо да функција F буде таква да што је веће $F(x, y)$, то излаз y боље одговара улазу x, односно у конкретном случају, дата класа боље одговара датом тексту. У овој генерализацији, дискриминантна функција постаје функција два аргумента, улаза и излаза, $F(x, y)$, при чему излаз није из скупа $\{-1, 1\}$ већ може представљати низ, стабло, граф итд. Ако са Y означимо скуп свих могућих излаза, без обзира у ком је облику излаз (низ, стабло, граф, ...), МСПВ предвиђа излаз (класу) који најбоље одговара улазу (тексту) тј. предвиђа излазни вектор у који максимизира вредност функције F за дати улазни вектор x. Прецизније, МСПВ предвиђа излаз на основу следеће једначине: $y^* = \underset{y \in Y}{\operatorname{argmax}}(F(x^*, y))$. Аналогно са МПВ, и у МСПВ претпостављамо да је функција F линеарна по ω, као и да се векторски пар улаз-излаз (x, y) може погодније представити пресликавањем помоћу неке функције Ψ, у неки други простор, па стога функцију F можемо записати као $F(x, y) = \langle \omega, \Psi(x, y) \rangle$.

Функција Ψ представља заједнички, векторски запис за један пар (x, y) и њен облик зависи од скупа података над којима се метода примењује. На пример, једна од примена МСПВ јесте одређивање („предвиђање“) дрвета извођења дате реченице у датој (формалној) граматици и у том случају, улазни податак x би представљао вектор речи које се појављују у реченици, излазни податак би представљао дрво извођења у датој граматици, а функција $\Psi(x, y)$ би била заједничка репрезентација реченице и њеног дрвета извођења. Та заједничка репрезентација би могла да буде вектор чија димензија одговара укупном броју правила у граматици, укључујући и правила извођења свих речи из скупа за

обучавање. Сваки елемент овог вектора би одговарао једном од свих могућих правила те граматике, а вредност на свакој позицији вектора одговарала би укупном броју појављивања (у дрвету извођења) правила које одговара тој позицији. Пример репрезентације вектора $\Psi(x, y)$ у овом случају приказан је на Слици 3.



Слика 3. Пример заједничке репрезентације улаза и излаза конструисан по узору на сличан пример извођења у енглеској граматици из (Tsochantaridis et al., 2004)

У примеру приказаном на Слици 3, вектор x чине речи дате реченице („Мали пас јури велику мачку“), а вектор y дрво извођења у датој формалној граматици. Вектор $\Psi(x, y)$ означава да је у извођењу реченице правило $S \to NPVP$ примењено једанпут, правило $S \to NP$ ниједном, правило $VP \to VNP$ једанпут, правило $NP \to AdjN$ два пута и тако даље, правило $V \to juri$ једанпут.

Постоје различите формулације МСПВ методе (Tsochantaridis et al., 2005), а у овом раду коришћена је такозвана формулација са једном „лабавом“ (енгл. *slack*) променљивом са рескалирањем маргине (Joachims et al., 2009). Како је при подучавању класификатора добро да маргина која раздваја инстанце различитих класа буде што шира а како је, са друге стране, могуће да неке инстанце при томе буду погрешно класификоване, ова формулација МСПВ подучава параметре класификатора у зависности од једне позитивне константе C која контролише нагодбу између минимизације грешке над скупом за подучавање и максимизације маргине (ова константа биће посебно анализирана у даљем раду). Описани алгоритам има полиномијалну сложеност по броју примера за подучавање, што је доказано у (Joachims et al., 2009).

3.2 Н-грами

Ако је дата ниска симбола $s = s_1 s_2 ... s_N$ над азбуком Σ, н-грам ниске s (за N, н природне бројеве) дефинише се као било која подниска суседних симбола ниске s дужине н. Над азбуком Σ може се дефинисати укупно $|\Sigma|^{\text{н}}$ различитих н-грама, при чему је $|\Sigma|$ величина (кардиналност) азбуке Σ. Овако представљени н-грами могу бити дефинисани на нивоу речи, карактера или бајта. На пример, 2-грами (уобичајен назив биграми) над ниском „dela, ne reči“ на нивоу речи садржаће само два биграма, „dela ne“ и „ne reči“ а биграми карактера (азбука Σ је латинична), биће de; el; la; a,; ,_; _n; ne; e_; _r; re; eč; či. Ако су карактери кодирани UTF-8 кодном схемом, слово „č“ кодирано је са два бајта чији је декадни садржај 196 141, редом, карактер „ ” (space, белина) кодиран је једним бајтом чији је садржај декадни број 32, итд, па је цела ниска у рачунару записана низом бајтова чија је декадна вредност 100 101 108 97 44 32 110 101 32 114 101 196 141 105. Дакле, у случају језика над латиничном азбуком, н-грами на нивоу бајта и нивоу карактера су веома слични с обзиром на чињеницу да је један карактер обично представљен једним бајтом. Разлика је обично и у скупу карактера који се разматрају (н-грами на нивоу карактера обично не узимају у обзир разлику између малих и великих слова, интерпункцију, цифре и размаке), а разлика је посебно значајна када се користи ћирилично писмо или друга писма као што су арапско, кинеско, и сл. Н-грами бајтова и н-грами карактера се равноправно користе за репрезентацију текстова у решавању различитих задатака из области истраживања података (Kešelj et al., 2003; Abou-Assaleh et al., 2004; Reddy and Pujari, 2006; Santos et al., 2011; Lui et al., 2014), са сличним резултатима. Мада н-грами бајтова понекад немају придружен смисао, посебно за човека (на пример, када садрже само један од два бајта којима је представљен један карактер), за њихову екстракцију није неопходно поседовати информацију о кодној шеми која је коришћена за запис текста па зато они представљају поједностављену репрезентацију за рад рачунара. У овом раду ми ћемо користити н-граме бајтова.

Када се користе у процесу обраде природних језика, неке од добрих особина које н-грами показују су релативна неосетљивост на правописне грешке, азбука знакова је унапред позната, независност од језика и садржаја, извршавање у једном пролазу, не захтева се никакво лингвистичко предзнање, и сл. Основни проблем код коришћења н-грама је експоненцијални број њихових могућности у односу на кардиналност азбуке. Ако је Σ азбука енглеског језика и ако се у разматрање укључи и знак за празнину, онда је $|\Sigma| = 27$. Ако се прави разлика између малих и великих слова и ако се у разматрање укључе цифре, онда је $|\Sigma| = 63$. Јасно је да ће многи алгоритми са н-грамима бити веома скупи са становишта израчунљивости већ за н $= 5$ или н $= 6$ (на пример, број различитих 5-грама над азбуком Σ је $63^5 \sim 10^9$). Коришћење модела и

техника заснованих на н-грамима у процесу обраде природних језика показало се као ефикасан приступ. Овај приступ нашао је примену у оквиру задатка претраживања информација (De Heer, 1974), компресије текста (Wiśniewski, 1987), откривања и исправљања правописних грешака (Zamora et al., 1981), откривање ауторства текста (Kešelj et al., 2003) и друго.

3.3 Репрезентација података

За потребе овог истраживања развијена су два класификатора који за основу имају оригинални МСПВ алгоритам и међусобно се разликују по начину на који су излазни подаци, односно класе текстова, представљени. Оба класификатора користе исту репрезентацију улазних података, н-грамима на нивоу бајта. Свака позиција у вектору н-грама једнака је вредности tf-idf статистике за дати н-грам. Статистика tf-idf (енгл. *term frequency – inverse document frequency*) је обично дефинисана тако да рефлектује колико је неки н-грам важан за документ у оквиру неког корпуса. Ова мера пропорционално расте са порастом броја појава н-грама у документу, али опада са порастом броја његових појава у целом корпусу. За документ ће бити значајнији они н-грами који имају већу вредност tf-idf статистике, односно они н-грами који се чешће јављају у том документу али се не јављају често у целом корпусу. Постоје разне варијанте за рачунање вредности tf и idf, а у овом раду су коришћене следеће мере (Manning et al., 2008)

1. *classic tf-idf*: $tf \cdot \log\left(\frac{n}{n_k+1}\right)$
2. *log tf-idf*: $1 + \log\left(tf\right) \cdot \log\left(\frac{n}{n_k+1}\right)$
3. *boolean1 tf-idf*: $\log\left(\frac{n}{n_k+1}\right)$
4. *boolean2 tf-idf*: $\log\left(1 + \frac{n}{n_k}\right)$

при чему tf представља нормализовану фреквенцију н-грама у припадајућем документу, n представља укупан број докумената у целом корпусу а n_k представља број оних докумената у корпусу у којима се бар једном појављује посматрани н-грам.

Репрезентација излазних података односно класа код развијених класификатора се разликује. Сваком тексту ЕбартХиер корпуса придружена је једна класа која се налази у листу ЕбартХиер хијерархије. У првом класификатору, свака класа је представљена као јединствени природни број, не узимајући притом у обзир повезаност класа кроз хијерархију. Ако класе нумеришемо редом као на Слици 1, скуп Y сводимо на скуп $\{1, 2, 3, 4, 5, 6, 7, 8\}$. На овај начин основну МСПВ методу сводимо на вишекласни равни класификатор. Вектор $\Psi(x, y)$, заједничка репрезентација улазног и излазног податка, у првом класификатору је димензије $p \cdot q$, где је p укупан број

различитих н-грама, односно димензија вектора x, а q купан број различитих класа корпуса. На тај начин, свака класа добија свој блок у ком се налазе нуле, уколико дати текст не припада датој класи, или вредности улазног вектора x, у супротном. На пример, ако текст x припада класи k, тада ће заједничка репрезентација имати следећи облик:

$$\Psi(x,y) = \left[\underbrace{0,\ldots,0}_{\text{класа 1}},\ldots,\underbrace{x_1,\ldots,x_p}_{\text{класа k}},\ldots,\underbrace{0,\ldots,0}_{\text{класа q}}\right]$$

У другом класификатору, свака класа је представљена као вектор чворова у ЕбартХиер хијерархији. Сваком чвору одговара једна позиција у вектору, при чему се на некој позицији налази „1“ ако се дати чвор појављује у путањи од корена хијерархије до листа дате класе, а „0“ у супротном. У нашем примеру (Слика 1), класа економија и бизнис била би представљена као $(0,0,1,0,0,0,0,0,0,1,0,0,1)$. На овај начин МСПВ методу сводимо на хијерархијски класификатор.

Вектор $\Psi(x,y)$ је у другом класификатору димензије $p \cdot r$, где је p укупан број различитих н-грама, односно одговара димензији вектора x, а r укупан број различитих чворова хијерархије. На тај начин, сваки чвор добија свој блок у коме се налазе нуле, уколико дати текст не припада класи која садржи дати чвор, или вредности улазног вектора x, у супротном. На пример, ако текст x припада класи економија и бизнис, представљеној са $(0,0,1,0,0,0,0,0,0,1,0,0,1)$, тада ће заједничка репрезентација имати следећи облик $\Psi(x,y) = [0,0,x,0,0,0,0,0,0,x,0,0,x]$ где је $0 = [\underbrace{0,\ldots,0}_{\text{p пута}}]$, а вектор x одговара улазном вектору. Основна разлика у репрезентацији вектора $\Psi(x,y)$ између равног и хијерархијског класификатора је у томе што равни класификатор третира све класе појединачно, док хијерархијски узима у обзир да су класе део дрволике хијерархије. Ова разлика се осликава на заједничку репрезентацију улазног и излазног податка, вектор $\Psi(x,y)$, који је различитих димензија за два случаја: код равног класификатора резервисане су позиције за сваку класу (која се налази у листовима хијерархије) док су код хијерархијског класификатора резервисане позиције за сваки чвор хијерархије класа, укључујући и листове, и унутрашње чворове и корен.

3.4 Опис имплементације

Оба класификатора су настала прилагођавањем јавно доступног и бесплатног оквира за МСПВ методу SVM$^{\text{struct}}$[3]. SVM$^{\text{struct}}$ је доступан у разним

[3] http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html

програмским језицима а овде је била коришћена имплементација у програмском језику C. Прилагођавање основне имплементације подразумевало је, између осталог, допуњавање постојећих структура за улазни вектор x и излазни вектор y, имплементацију функције за генерисање вектора $\Psi(x, y)$ (заједничке репрезентације улазног вектора x и излазног вектора y), имплементацију функције која дефинише меру тачности и имплементацију функције за евалуацију квалитета класификатора.

Подаци су припремани на следећи начин:

1. за сваки текст је алатом Text::Ngrams (Kešelj et al., 2003) генерисан низ н-грама бајтова које он садржи као и колико се пута сваки н-грам бајтова појављује. Генерисани су подаци за н-граме бајтова дужине $\{2, 3, 4, 5, 6, 7\}$.
2. имплементиран је скрипт који је за сваки текст на основу н-грама генерисао улазни вектор x у формату који захтева класификатор SVM$^{\text{struct}}$.

3.5 Опис експеримента

Целокупан корпус текстова подељен је на скуп за подучавање и скуп за тестирање у односу 2:1, што је један од најчешћих начина поделе у класификацији текстова (Bellotti and Crook, 2009). Од укупно 60.637 текстова, скуп за подучавање је сачињавало 40.426 а скуп за тестирање 20.211 текстова. Расподела текстова по класама такође прати ову размеру и приказана је у Табели 1.

Класа	Скуп за подучавање	Скуп за тестирање	Цео корпус
1	10681	5340	16021
2	7173	3586	10759
3	12290	6145	18435
4	772	386	1158
5	9172	4586	13758
6	140	70	210
7	32	15	47
8	166	83	249
Укупно	40426	20211	60637

Табела 1. Број текстова у скупу за подучавање и скупу за тестирање по класама

Сваки текст ЕбартХиер корпуса представљен је н-грамима бајтова на основу којих се изграђује векторска репрезентација погодна за улаз у класификатор,

такозвани улазни вектор x. Сваком н-граму бајтова из корпуса одговара једна позиција у вектору x, а вредност на тој позицији једнака је вредности једне од 4 tf-idf статистике за одговарајући н-грам бајтова. У зависности од дужине н-грама бајтова ($n \in \{2 \dots 7\}$) и од одабране tf-idf мере (classic, log, boolean1, boolean2), генерисано је 24 репрезентације корпуса ($n = 2$, мера=classic, ..., $n = 7$, мера=classic, $n = 2$, мера=log, ..., $n = 7$, мера=log, ...,$n = 2$, мера=boolean1, ..., $n = 7$, мера=boolean1, $n = 2$, мера =boolean2, ..., $n = 7$, мера=boolean2). У свакој репрезентацији, скупови за подучавање и тестирање се састоје од истих текстова како би се могли међусобно поредити.

У намери да испитамо да ли један тип класификатора даје боље резултате за одређену дужину н-грама бајтова или за одређену tf-idf статистику, извршили смо равну и хијерархијску класификацију за сваку репрезентацију корпуса на следећи начин:

1. Извршена је 10-унакрсна валидација (енгл. *cross validation*) на скупу за подучавање којом је одређена оптимална вредност параметра C.
2. За добијену вредност параметра, подучаван је класификациони модел на целом скупу за подучавање.
3. Тестирање и евалуација модела спроведени су на скупу за тестирање.

Након добијених резултата тестирања, извршено је поређење перформанси добијених модела.

3.6 Евалуација

Да би се анализирале перформансе МСПВ метода хијерархијске класификације, користимо неке од уобичајених мера квалитета класификације – прецизност (енгл. *precision*), која се дефинише као проценат примера који су исправно класификовани међу свим примерима који су додељени одређеној класи, одзив (покривање, енгл. *recall*) који дефинише колико примера за тестирање из дате класе класификатор може да препозна и Ф-мера (енгл. *F-measure, F1*), која се дефинише као хармонијска средина прецизности и одзива (Baeza-Yates et al., 1999):

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

при чему је TP (True Positives) број тачно позитивно класификованих докумената, TN (True Negative) број тачно негативно класификованих докумената, FP (False Positive) број погрешно позитивно класификованих

докумената и *FN* (False Negative) број погрешно негативно класификованих докумената.

Ове мере су дефинисане за случај бинарне класификације (када постоје само две класе). У случају када се класификација врши на више од две класе, потребно је извршити усредњавање ових мера по свим класама. То може да буде урађено на два начина: може се тражити макропросек, где се свакој класи придаје исти значај, и микропросек, где се фаворизују класе које садрже већи број докумената. Код макропросека се најпре израчунава вредност мере за сваку класу појединачно, а затим се врши усредњавање тих вредности по броју класа. Код микропросека се израчунавају вредности за *TP, TN, FP* и *FN* за сваку класу појединачно. Затим се израчунавају вредности *TP, TN, FP* и *FN* као суме свих *TP, TN, FP* и *FN* за све класе редом. На крају се израчунава вредност мере за добијене сумиране вредности *TP, TN, FP* и *FN*. У овом раду користићемо микропросек Ф-мере (микро-Ф1). Основни недостатак ових мера је подједнако кажњавање грешака на разним нивоима хијерархије.

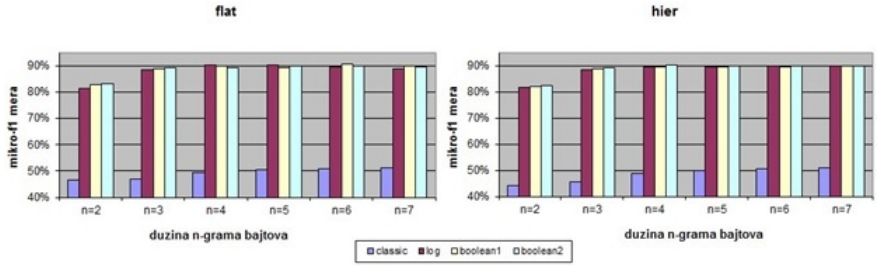
4. Резултати

На ЕбартХиер корпус примењене су две варијанте МСПВ методе: у једној се спроводи равна класификација, тј. не узима се у обзир да су класе хијерархијски повезане, а у другој се спроводи хијерархијска класификација. За текстове корпуса генерисано је 6 различитих н-грамских репрезентација, за н-граме бајтова дужине н из скупа $\{2, \ldots, 7\}$, за 4 различите tf-idf мере приказане у поглављу 3.3. Над сваким овако добијеним скупом за подучавање, за оба типа класификатора, извршена је 10-унакрсна валидација којом је одређен параметар класификатора C из скупа вредности 10^{-2} до 10^{2} са кораком 10, који даје најбоље резултате.

На Слици 4 приказани су резултати евалуације оба типа класификатора, представљени микро-Ф1 мером, за корпус представљен н-грамима бајтова дужине од 2 до 7 и за све 4 tf-idf мере, док су комплетни нумерички подаци приказани у Табели 2. Оба типа класификатора (равни и хијерархијски) показују тенденцију да класификатори са мером classic имају лошије перформансе од осталих. Вредности микро-Ф1 мере су за остале мере приближне, при чему разлике микро-Ф1 мере различитих класификатора за исту меру скоро нигде не прелазе 1% (осим за меру classic, за n=2 и n=3 где редом износе 2,32% и 1,66%).

Перформансе најбољих класификатора за оба типа (равни и хијерархијски) анализиране су и по класама. Међу равним класификаторима, највећу вредност микро-Ф1 мере имао је класификатор за меру boolean1 за текстове представљене 6-грамима бајтова (микро-Ф1мера износи 90,43%) а за хијерархијски за меру

boolean2 за текстове представљене 4-грамима бајтова (микро-Ф1 мера износи 90,17%). С обзиром да је број текстова по класама различит (наведен на Слици 1), класе се могу поделити у две групе: „велике“ (1, 2, 3 и 5) и „мале“ (4, 6, 7, 8). Резултати су представљени на Слици 5. Резултати за „велике“ класе су приближни за оба типа класификатора, док су разлике знатно израженије за мале класе. Најбољи хијерархијски класификатор много боље предвиђа „мале“ класе 6 и 7. Највећу тачност најбољи класификатори оба типа показују за „малу“ класу означену са 8 (фудбал).



Слика 4. Резултати евалуације два типа класификатора (flat-равног и hier-хијерархијског) за различите улазне податке (дужине н-грама бајтова од 2 до 7) и различите мере

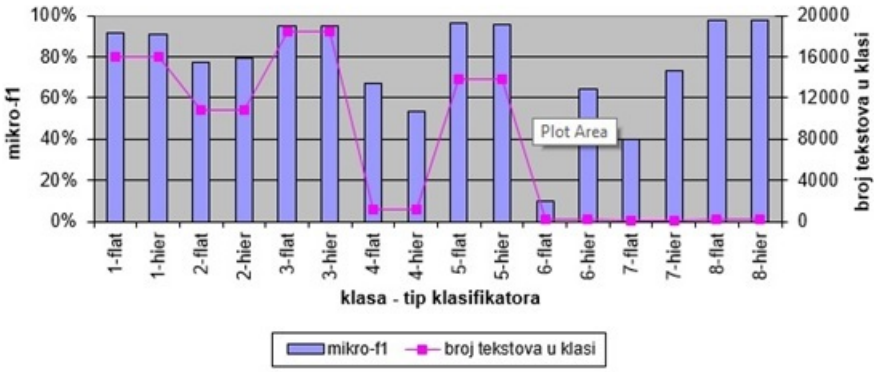
flat	classic	log	boolean1	boolean2
n=2	46.53%	81.46%	82.71%	83.25%
n=3	47.08%	88.27%	88.70%	89.28%
n=4	49.33%	90.05%	90.02%	89.18%
n=5	50.35%	90.05%	89.23%	89.77%
n=6	50.89%	89.60%	90.43%	89.74%
n=7	51.12%	88.88%	89.88%	89.60%

hier	classic	log	boolean1	boolean2
n=2	44.21%	81.75%	82.04%	82.45%
n=3	45.42%	88.34%	88.86%	89.24%
n=4	48.61%	89.39%	89.58%	90.17%
n=5	49.84%	89.46%	89.57%	89.70%
n=6	50.65%	89.84%	89.47%	89.78%
n=7	51.01%	89.88%	89.68%	89.69%

Табела 2. Перформансе равног (flat, табела са леве стране) и хијерархијског (hier, табела са десне стране) класификатора за различите репрезентације улазног корпуса приказани мером микро-Ф1

5. Закључак

Класификација текстова на српском језику из хијерархијски организованог корпуса, применом методе подржавајућих вектора показује приближне резултате за равну и хијерархијску варијанту, за сваку tf-idf меру.



Слика 5. Резултати евалуације најбољих класификатора оба типа (flat-равног и hier-хијерархијског) по класама. Приказан је и број текстова у корпусу за сваку класу

Укупно најбољи резултат добијен је равном варијантом класификатора за boolean1 меру и износи 90,43%. Ово је нешто бољи резултат од раније публикованог резултата за н-грамску класификацију равно организованог подскупа Ебарт корпуса за који је израчуната вредност микро-Ф1 мере 88,5% (Graovac, 2012). Хијерархијска класификација даје нешто боље резултате од равне за неке „мале“ класе.

Хијерархијска класификација даје резултате слабије од очекиваних. Разлог за то се може тражити у плиткој хијерархији са малим бројем докумената, али такође и у коришћеној мери евалуације. При евалуацији резултата класификације, посебно хијерархијске, није довољно пребројати промашаје већ је потребно проценити и њихову тежину, односно растојање предвиђене од тачне класе.

Имајући то у виду, очекујемо да ће се бољом, хијерархијској класификацији прилагођеном мером евалуације као и богатијом хијерархијом докумената постићи резултати који превазилазе резултате равне класификације (Silla et al., 2011).

Литература

Abou-Assaleh, Tony, Nick Cercone, Vlado Kešelj, and Ray Sweidan. "N-gram-based detection of new malicious code". In *Computer Software and Applications Conference*, COMPSAC 2004. Proceedings of the 28th Annual International, IEEE. Vol. 2 (2004), 41-42. Accessed September 1, 2015. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1342667.

Baeza-Yates, Ricardo, Ribeiro-Neto Berthier, et al. *Modern information retrieval*, vol. 463. ACM press New York, 1999.

Bellotti, Tony and Jonathan Crook. "Support vector machines for creditscoring and discovery of significant features". *Expert Systems with Applications*, vol. 36, no. 2 (2009): 3302-3308. Accessed September 1, 2015. http://www.sciencedirect.com/science/article/pii/S0957417408000857.

De Heer, T. "Experiments with syntactic traces in information retrieval". *Information Storage and Retrieval*, vol. 10, no. 3 (1974): 133–144. Accessed September 1, 2015. http://www.sciencedirect.com/science/article/pii/0020027174900151.

Eyheramendy, Susana, David D. Lewis, and David Madigan. "On the Naive Bayes model for text categorization". In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics Conference*, 332-339. Florida: Society for Artificial Intelligence and Statistics, 2003.

Graovac, Jelena. "Serbian text categorization using byte level n-grams". In *Proceedings of CLoBL 2012: Workshop on Computational Linguistics and Natural Language, 5th Balkan Conference in Informatics*, 93–97. 2012 Accessed September 1, 2015. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.416.6155&rep=rep1&type=pdf.

Graovac, Jelena. "Wordnet-based text categorization technique". *INFOtheca – Journal of Information and Library Science*, vol. 14, no. 2 (2013): 2–17. Accessed September 1, 2015. http://infoteka.unilib.rs/2013/br.2/eng/Infotheca-2-2013-Jelena-Graovac.pdf.

Graovac, Jelena. "A variant of n-gram based language-independent text categorization". *Intelligent Data Analysis*, vol. 18, no. 4 (2014): 677–695.

Graovac, Jelena. "Text categorization using n-gram based language independent technique". Natural Language Processing for Serbian - Resources and Applications. In *Proceedings of the Conference "35th Anniversary of Computational Linguistics in Serbia"*, 124–135, 2014. Accessed September 1, 2015. http://poincare.matf.bg.ac.rs/~jgraovac/publications/jg35CLS.pdf.

Graovac, Jelena and Gordana Pavlović-Lažetić. "Language-Independent Sentiment Polarity Detection in Movie Reviews: A Case Study of English and Spanish". In *6th International Conference ICT Innovations 2014*, 13–22, 2014. Accessed Sep-

tember 1, 2015. http://poincare.matf.bg.ac.rs/~jgraovac/publications/jgSPD.pdf.

Graovac, Jelena and Jelena Kovačević, and Gordana Pavlović-Lažetić. “Language Independent n-Gram-Based Text Categorization with Weighting Factors: A Case Study”. *JIDM - Journal of Information and Data Management*, vol. 6, no. 1 (2015): 4–17.

Gross, Maurice. “The construction of local grammars”. In *Finite State Language Processing eds. Emmanuel Roche and Yves Schabs*, 329–354. Massachusetts: The MIT Press, 1997. Accessed September 1, 2015. https://halshs.archives-ouvertes.fr/halshs-00278316/document.

Joachims, Thorsten. “Text categorization with support vector machines: Learning with many relevant features”. Springer Berlin Heidelberg, 1998.

Joachims, Thorsten, Thomas Finely, and Chun-Nam John Yu. “Cutting Plane Training of Structural SVMs”. *Machine Learning Journal*, vol. 77, no. 1 (2009): 27–59. Accessed September 1, 2015. http://link.springer.com/article/10.1007/s10994-009-5108-8.

Johnson, Christopher R., Miriam R. L. Petruck, Collin F. Backer, Michael Ellsworth, Josef Ruppenhofer, and Charles J. Fillmore. “FrameNet: Theory and Practice”, 2002. Accessed September 1, 2015. http://www.icsi.berkeley.edu/framenet.

Kešelj, Vlado, Fuchung Peng, Nick Cercone, and Calvin Thomas. “N-gram-based author profiles for authorship attribution”. In *Proceedings of the conference pacific association for computational linguistics, PACLING*, vol. 3, 255–264. 2003. Accessed September 1, 2015. http://web.cs.dal.ca/~vlado/papers/pacling03.pdf.

Krstev, Cvetana, and Gordana Pavlović-Lažetić, and Ivan Obradović. “Using textual and lexical resources in developing Serbian wordnet”. *Romanian Journal of Information Science and Technology*, vol. 7, no. 1–2 (2004): 147–161. Accessed September 1, 2015. http://xn--c1azn.xn--90a3ac/LicnePrezentacije/ivan_obradovic/Radovi/RJIS_2004.pdf.

Krstev, Cvetana, Duško Vitas, Denis Maurel, and Mickaël Tran. “Multilingual ontology of proper names”. In *Proceedings of 2nd Language & Technology Conference*, April 21-23, 2005, Poznań, Poland, ed. Zygmunt Vetulani, 116–119. Poznań, Wydawnictwo Poznańskie, 2005. Accessed September 1, 2015. https://hal.archives-ouvertes.fr/hal-01108242/document.

Lui, Marco, Jey Han Lau, and Timothy Baldwin. “Automatic detection and language identification of multilingual documents”. *Transactions of the Association for Computational Linguistics 2* (2014): 27–40. Accessed September 1, 2015. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/86/30.

Manning, Christopher, and Hinnrich Schütze. *Foundations of Statistical Natural Language Processing*. Cambridge: MIT Press, 1999.

Manning, Christopher, Prabhakar Raghavan, and Hinnrich Schütze. "Scoring, term weighting, and the vector space model". In *Introduction to Information Retrieval.* Cambridge: Cambridge University Press, 2008.

Lafferty, John, Andrew McCallum and Fernando C. N. Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". *Machine Learning*, 282–289. 2001

Miller, George A. "WordNet: a lexical database for English". *Communications of the ACM*,vol. 38, no. 11 (1995): 39–41. Accessed September 1, 2015. http://nlp.cs.swarthmore.edu/~richardw/papers/miller1995-wordnet.pdf.

Pavlović-Lažetić, Gordana and Jelena Graovac. "Ontology-driven conceptual document classification". In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, 383–386. 2010. Accessed September 1, 2015. http://poincare.matf.bg.ac.rs/~jgraovac/publications/odcdc.pdf.

Quinlan, J. R. "Improved use of continuous attributes in c4. 5". *Journal of Artificial Intelligence Research*, vol. 4 (1996): 77–90. Accessed September 1, 2015. http://www.jair.org/media/279/live-279-1538-jair.pdf.

Reddy, D. Krishna Sandeep, and Arun K. Pujari. "N-gram analysis for computer virus detection". *Journal in Computer Virology*, vol. 2, num. 3 (2006): 231–239. Accessed September 1, 2015. http://link.springer.com/article/10.1007/s11416-006-0027-8.

Santos, Igor, Javier Nieves, and Pablo G. Bringas. "Semi-supervised learning for unknown malware detection". In *International Symposium on Distributed Computing and Artificial Intelligence* vol. 91, 415-422. Berlin Heidelberg, 2011.

Scott, Sam, and Stan Matwin. "Text classification using wordnet hypernyms". In *Usage of WordNet in Natural Language Processing Systems: Proceedings of the Workshop*, 38-44, 1998. Accessed September 1, 2015. http://www.aclweb.org/website/old_anthology/W/W98/W98-0706.pdf.

Sebastiani, Fabrizio. "Machine learning in automated text categorization". *In ACM computing surveys (CSUR)*, vol. 34, no. 1 (2002): 1–47.

Silla J., Carlos N. and Alex A. Freitas. "A survey of hierarchical classification across different application domains". *Data Mining and Knowledge Discovery*, vol.22, no. 1–2 (2011): 31–72. Accessed September 1, 2015. http://link.springer.com/article/10.1007/s10618-010-0175-9.

Steinberger, Ralf, Mohamed Ebrahim, and Marco Turchi. "JRC EuroVoc Indexer JEX-A freely available multi-label categorisation tool". *arXiv preprint arXiv*, vol. 1309, no. 5223 (2013). Accessed September 1, 2015. http://arxiv.org/ftp/arxiv/papers/1309/1309.5223.pdf.

Sun, Aixin, and Ee-Peng Lim. "Hierarchical text classification and evaluation". In *Proceedings 2001 IEEE International Conference on Data Mining*, 521–528. Computer Society, 2001.

Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*, vol. 1. Pearson Education India, 2006.

Tsochantaridis, Ioannis, Thomas Hofman, Thorsten Joachims, and Yasemin Altun. “Support Vector Machine Learning for Interdependant and Structured Output Spaces”. In *Proceedings of the twenty-first international conference on Machine learning*. 104. New York: ACM, 2004.

Tsochantaridis, Ioannis, Thomas Hofman, Thorsten Joachims, and Yasemin Altun. “Large Margin Methods for Structured and Interdependant Output Variables”. *Journal of Machine Learning Research*,vol 6, (2005): 1453–1484. Accessed September 1, 2015. http://machinelearning.wustl.edu/mlpapers/paper_files/TsochantaridisJHA05.pdf.

Vitas, Duško and Cvetana Krstev. “Derivational morphology in an e-dictionary of Serbian”. In *Proceedings of 2nd Language & Technology Conference*. April 21-23, 2005, Poznań, Poland, ed. Zygmunt Vetulani, 139-143. Poznań: Wydawnictwo Poznańskie, Accessed September 1, 2015. http://poincare.matf.bg.ac.rs/~cvetana/biblio/ltc_134_vitas_2.pdf.

Vitas, Duško , Cvetana Krstev, Ivan Obradović, Ljubomir Popović, and Gordana Pavlović-Lažetić. “An overview of resources and basic tools for processing of Serbian written texts”. In *Proceedings of the Workshop on Balkan Language Resources, 1st Balkan Conference in Informatics, Thessaloniki, Greece*, 97–104, (2003). Accessed September 1, 2015. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.8096&rep=rep1&type=pdf.

Wiśniewski, Janusz L. “Effective text compression with simultaneous digram and trigram encoding”. *Journal of Information Science*, Vol. 13, No.3, (1987): 159–164.

Yang, Yiming, and Jan O. Pedersen. “A comparative study on feature selection in text categorization”. In *Proceedings of the 14th International Conference on Machine Learning*, 412–420, 1997. Accessed September 1, 2015. http://www.surdeanu.info/mihai/teaching/ista555-spring15/readings/yang97comparative.pdf.

Yi, Kwan, and Jamshid Beheshti. “A text categorization model based on Hidden Markov models”. In *Proceedings of the 31st Annual Conference of the Canadian Association for Information Science*, 275–287, 2013. Accessed September 1, 2015. http://www.cais-acsi.ca/ojs/index.php/cais/article/view/420/585.

Zamora, E. M., Joseph J. Pollock, and Antonio Zamora. “The use of trigram analysis for spelling error detection”. *Information Processing & Management*,vol. 17, no. 6 (1981): 305–316. Accessed September 1, 2015. http://www.sciencedirect.com/science/article/pii/0306457381900443.