ПРИСТУП ИЗГРАДЊИ СТЕМЕРА И ЛЕМАТИЗАТОРА ЗА ЈЕЗИКЕ С БОГАТОМ ФЛЕКСИЈОМ И ОСКУДНИМ РЕСУРСИМА ЗАСНОВАН НА ОБУХВАТАЊУ СУФИКСА

Владо Кешељ,
Универзитет Dalhousie

Данко Шипка,
Државни универзитет Аризона

Превод с енглеског: Вања Радуловић

Апстракт: Представљамо општи суфиксни метод за конструисање стемера и лематизера за језике са богатом флексијом и оскудним ресурсима. Описали смо један ефикасан приступ помоћу кога се процес може директно имплементирати, а евалуација је извршена на конструкцији стемера за српски језик. Евалуација на веродостојним подацима дала је тачност од 79%.

1 Увод

Два важна задатка ниског нивоа у обради природних језика су стеминг и лематизација. Стеминг је добро познат у обради природних језика, проналажењу информација, и истраживањима везаним за копање по тексту, као неопходна припрема за друге задатке, какви су нпр. проналажење текста и докумената, груписање докумената, класификација, екстракција информација и у другим применама која зависе од садржаја. Стеминг би се могао описати као трансформација речи код које може дођи до уклањања неких суфикса при чему се не губи основни семантички садржај. Врло фреквентне речи се обично уклањају листама одбачених речи, или „стоп речи", у систему за проналажење информација, па се на њих стеминг ни не примењује. Стеминг можемо да замислимо као процес нормализације у којем се неколико морфолошких варијанти мапира у исти облик. О стемингу и његовој примени у проналажењу информација детаљно се говори у [7]. Систему за проналажење информација стемер доноси две важне користи: (1) Постиже се бољи одзив у проналажењу информација јер се речи из упита сравњују са различитим варијантним облицима у документу и (2) Стеминг смањује обим целокупног речника термина што доводи до значајно веће ефикасности у односу на брзину и потребну меморију услед смањења величине индекса термина и самих димензија вектора термина. Наиме, у моделу векторског простора код проналажења информација документи су представљени као тежински вектори, где свака тежина одговара термину из речника. Уклањање речи листом за одбацивање као и неких врло ретких речи из речника први је корак у смањењу димензија, а поврх тога, процењује се да даље смањење стемингом износи и једну трећину [6]. Битна предност у учинку проналажења понекад се оспорава ([4] §3.4), барем у случају енглеског језика, али за језике са богатом флексијом, свођење на корен речи је кључно [5]. Стеминг се такође може користити у припремној обради за екстраховање информација и за многе друге задатке.

Да ли је уклањање суфикса довољно? Поред уклањања суфикса примамљиво звучи и уклањање префикса, али префикси обично мењају значење речи у потпуности па се обично сматра да је боље да се не дирају [7]. Још од објављивања Портеровог стемера [6] стеминг је углавном трансформација која се заснива на одсецају суфикса, и оно је већ успешно примењено на више индо европских језика, на пример: стемери за 16 језика су имплементирани у *Snowball*-у [7]. Међутим, ова методологија оријентисана на суфиксе се не сме уопштавати на све језике,

јер, на пример, арапски се, у морфолошким трансформацијама ослања на префиксе, суфиксе и инфиксе, као што је коришћење префикса за означавање глаголског лица. Језици који припадају групи банту језика користе префиксе за грађење множине. За неке језике, као што је кинески, оваква питања уопште нису од важности. Неправилне флексије се не могу лако описати трансформацијама заснованим на уклањању суфикса па њих треба третирати преко листи изузетака, од којих је једна и листа одбачених речи. Појам *стем* и појам *корен* су повезани али се разликују: *стем* је производ уклањања који „окупља" све семантички блиске речи, док је корен „унутрашња" реч из које се почетна реч изводи, односно он има етимолошко значење [7]. Добијање корена речи обично захтева и уклањање префикса, али се он код стеминга не дира. Други рачунарски проблем везан за стеминг је морфолошка анализа, чији је циљ растављање речи на најмање делове који чувају јединично занчење које је повезано са значењем првобитне речи [3].

Лематизација. Слично стемингу, лематизација је морфолошка трансформација која мења реч у нормализовани облик. Ипак, док је циљ стеминга прикупљање повезаних морфолошких варијација у јединствени облик, *лематизер* враћа одговарајућу лему, која је нормализовани облик каква би се нашао и у речнику.

У одељку 2 овог рада ћемо говорити о другим радовима из ове области, затим ћемо у 3. одељку формално представити свој приступ и методу. У 4. одељку ћемо описати ресурсе које смо користили као полазну основу. У 5. одељку описаћемо експерименте и дискутоваћемо добијене резултате, а у 6. одељку закључујемо са прегледом резултата и главних доприноса, као и предлогом задатака за будући рад.

Ресурси и програм који се користе у раду јавно су доступни и могу се на нађи на адреси http://www.cs.dal.ca/~vlado/nlp/2007-sr.

2 Преглед других радова из ове области

Портеров стемер за енглески [6] је вероватно најпознатији и у најширој употреби. Из отприлике истог периода, премда је настао нешто раније, је други добро познати стемер, Ловинсов стемер. Првобитни Портеров стемер је био имплементиран у BCPL-у, програмском језику који је претходио C и који се данас ретко користи. Стемер је наново имплементиран у доста различитих језика, али читалац мора имати на уму да многи од њих не имплементирају стемер тачно онако како је спецификован.[1] И Портеров и Ловинсов стемер пример су алгоритамских стемера. Постоје два општа приступа стемингу: један је заснован на речнику а други је алгоритамски. У следећем поглављу о њима ћемо детаљније говорити.

Након појаве Портеровог стемера, имплементиран је још одређени број других стемера. На пример, *Snowball* [7] тренутно садржи стемере за 16 језика. Руски је једини балтословенски језик који је тренутно имплементиран у *Snowball*-у. Постоје и друге имплементације које су јавно доступне. Значајна је станица CPAN[2], на којој је угошћено неколико стемера, укључујући и омотачки модул за *Snowball*. За већину језика нема јавно доступних стемера, посебно за језике са оскудним електронским лингвистичким ресурсима. Парафразирајући [7], можемо рећи да постоји велики број публикација које говоре о стемингу, али мало њих даје описе који би се могли директно имплементирати у популарним и ефикасним програмским језицима, какви су C, Перл, Јава и сл; такође, мали број публикација пружа квантитативну анализу и процену учинка стемера.

Теоретска основа наше методологије повезана је са методологијом коначних аутомата која је описана у [1] и [8].

Што се тиче српског језика, наша потрага за ширим скупом стемера за било који од словенских језика бивше Југославије дала је само неколико резултата. Заправо су пронађе-

на само два стемера. Стемер за словеначки је описан у [5] и процењен је на једном задатку проналажења информација, али нисмо могли да пронађемо ниједну доступну имплементацију. „Три нова стемера за словеначки" су поменута на веб презентацији INCO-Copernicus пројекта[3]. На *Snowball* листи се одвијала дискусија око укључивања словеначког стемера у оквир система[4]. Јавно је доступан Перл кôд за хрватски стемер[5]. Он укључује врло скромну документацију (само неколико коментара о ревизији кôда), и корисничку идентификацију аутора 'dpavlin'. Изгледа да је у питању кратак и добро написан стемер, али није јасно колика је његова обухватност. Може бити да је у питању стемер за играње начињен за само 143 речи колико их је дато у подацима за тестирање. Други слични пројекти из морфолошке анализе који су изгледа имплементирали *лематизере*, а не стемере су [10] за српски и [9] за хрватски.

Доприноси. Три главна доприноса овог рада су: (1) развој и имплементација јавно доступног стемера за српски заједно са придруженим ресурсима, (2) опис и анализа квантитативне анализе стемера и различитих корака у процесу њиховог развоја, и (3) предлагање и тестирање општег приступа изградњи стемера и лематизера за језике са богатом флексијом а оскудним ресурсима. Налазимо да је метод који смо користили омогућио занимљив увид у алгоритме и структуре података који су потребни за ефикасну имплементацију таквих стемера.

3 Претходна истраживања

Алгоритамски и речнички стемери. Постоје два приступа изградњи стемера:

1. заснован на речнику и
2. алгоритамски.

У *речничком приступу* ослањамо се на широко лингвистичко знање прикупљено у машински читљивом речнику, док у *алгоритамском приступу* користимо релативно мали број правила. Алгоритамски приступ је у општем случају делотворнији и компактнији у смислу величине програма, то јест комплексности Колмогорова. Према Окамовој оштрици, то би требало да доведе до веће општости и робустности када се наиђе на речи које нису претходно виђене. С друге стране, речнички приступ је једноставнији у обради изузетака и може бити лакши за модификовање и одржавање. Граница између ових приступа није јасна: за речнички приступ су обично потребна бар нека правила. На пример, у језицима са богатом флексијом, какав је српски, и властита имена су променљива али се не може очекивати да сва властита имена буду укључена у речник. Слично, алгоритамски стемер обично има листу изузетака, што су у ствари мали речници. Приступ који овде разматрамо је алгоритамски. Алгоритамски приступ има додатне предност, осим оних добро познатих, када је на располагању полазни лексички ресурс ограничене покривености за значајним бројем грешака, односно шума. Претерано подешавање модела са ресурсом, до чега би дошло код примене речничког приступа, довело би до смањења учинка стемера не само за непознате речи, већ и за лексикон на коме је обављено обучавање.

Стеминг (или уклањање суфикса) и лематизација. Под општијим термином *лематизација* разликујемо три различита нивоа, од којих сваки даје све напреднију анализу речи:

1. **уклањање суфикса или стеминг**, је већ описан процес,
2. **директна лематизацију** или превод облика речи у лему, и
3. **анотирана лематизација** или превод облика речи у лему анотирану својствима која су придружена облику речи.

Код **директне лематизације**, за сваку дату реч из текста лематизер враћа лему, то јест основни облик речи, какав се може наћи у речнику. Предност директне лематизације над сте-

мингом укључује боље разликовање варијација једне речи, а то би довело до бољих апликација у домену проналажења информација. Овај приступ се такође може користити код on-line речника где корисници често уносе варијанте једне речи које као такве нису у речнику, али њихов основни облик јесте. Мањкавост директне лематизације је што може постојати значајан број инхерентних двозначности пошто неки облици речи могу одговарати различитим лемама. Без познавања контекста *лематизер* може само да врати све леме и препусти кориснику или апликацији да разреши двозначност. У процесу уклањања суфикса овакве двозначности разрешавају се стапањем различитих лема и њихових облика у исту класу, која има јединствени репрезентативни стем.

Анотирана лематизација пресликава, слично директној лематизацији, облик речи у једну или више лема, додајући скуп морфолошких својстава која су придружена том облику речи, као што су род, падеж и број. Скуп својстава треба да буде такав да обрнути процес морфолошког генерисања производи тачан облик речи на основу задате леме и скупа морфолошких својстава. Анотирана лематизација се може сматрати проширеном директном лематизацијом, пошто пружа више информација. Код анотиране лематизације може доћи до још већег степена вишезначности него код директне лематизације, ако више скупова својстава генерише исти облик речи из исте леме.

Као пример, *уклањање суфикса* или *стеминг* преводи све речи из скупа {boxer, boxers, boxing, boxed,...} у реч ‘box’; *директна лематизација* преводи ‘boxers’ → ‘boxer’ i ‘boxing’ → ‘box’; а *анотирана лематизација* производи следећи превод ‘boxers’ → ‘boxer. noun.plural’.

4. Методологија

У досада објављеној литератури један искључиво алгоритамски приступ стемингу је уклањање суфикса, или прецизније његова замена. Типичан представник је Портеров алгоритам. Алгоритам групише правила у пет корака који се редом примењују, при чему се примењује највише једно правило из групе. Свако правило састоји се из услова и замене облика $s_1 \rightarrow s_2$, а тумачи се да ако је услов задовољен за неку реч и реч има суфикс c_1 он се замењује суфиксом s_1. Услови који се користе у Портеровом стемеру су или такви да се могу представити у облику захтева за одређеним суфиксом, или укључују најмању дужину стема у броју слогова, или је услов такав да стем садржи самогласник. Ако је неколико правила применљиво у једној групи, примењује се најдужи сравњени суфикс. Укупни број правила је 63, али ако желимо да их представимо у формату „обичних суфикса”, на пример ако уместо да сравњујемо „удвостручени сугласник” заправо понављамо правило за сваки сугласник, број правила је око 120. Чини се да се оваква врста правила може применити и на стемере за друге језике из групе индоевропских језика. Ово је мотивисало развој програмског језика специјалне намене *Snowball* [7].

Примећено је да услови за дужину стема нису од посебне важности за руски и словеначки.[6] На основу овог опажања, претпостављамо да би правила замене обичних суфикса требало да буду довољна за развој нашег стемера. Користимо само неколико једноставних услова за дужину стема, а даље разматрање ових услова биће део даљег рада. У поређењу са комплекснијим правилима, правила обичних суфикса су довољна с обзиром на то да се комплекснија суфиксна правила могу изразити преко већег скупа правила обичних суфикса. Пошто је Портеров стемер приближно еквивалентан скупу од 120 правила обичних суфикса за енглески, који је језик са слабом флексијом, очекујемо да би број оваквих правила за језик са богатом флексијом, какав је српски, могао достићи и више хиљада.

Лексички морфолошки ресурси. Наш основни лексички ресурс је листа пресликавања речи w у њихове леме l. Ово „пресликавање" није функција пошто се реч може пресликати у више лема. То је општа релација над речима: $w \xrightarrow{l} l$.

SDDL (Једноставни директни лематизер заснован на речнику, од енглеског *Simple Dictionary-based Direct Lemmatizer*) може се сачинити коришћењем овог ресурса. За било коју задату реч w лема $l(w)$ је одређена релацијом из ресурса $w \xrightarrow{l} l(w)$. Два главна проблема су: (1) двосмисленост, пошто једној речи може бити придружено више лема, и (2) покривеност, документи по правилу садрже речи којих претходно није било у речнику (*hapax legomena*).

Извођење стемера. Процес *извођења стемера* делимо у следеће кораке:

4.1 креирање класа стемова,
4.2 генерисање стемова и суфикса,
4.3 сортирање суфикса по фреквентности, и
4.4 генерисање суфиксних правила.

4.1. Креирање класа стемова. Ако две речи w_1 и w_2 имају исти стем, кажемо да се стапају [7], и пишемо $w_1 \sim w_2$. Ова релација је релација еквиваленције и она дели скуп речи у класе еквиваленције. Ове класе називамо класама стемова. Класе стемова креирамо из нашег ресурса тако што релацију стапања дефинишемо као рефлексивно, симетрично и транзитивно затворење релације $\xrightarrow{l}$. Наиме, за било које три речи w_1, w_2 и w_3:

$w_1 \sim w_1$,
$w_1 \xrightarrow{l} w_2 \Rightarrow w_1 \sim w_2 \wedge w_2 \sim w_1$, и
$w_1 \sim w_2 \wedge w_2 \sim w_3 \Rightarrow w_1 \sim w_3$.

Транзитивно затворење се често имплементира помоћу матрица, али у овом случају то би вероватно било недозвољено скупо због величине матрица. Ефикасан начин је да се користи UNION-FIND структура података [2]. Резултат ове фазе су класе стемова, то јест групе речи које би требало стемером да се стопе. Све речи изведене из исте леме, сходно релацији $\xrightarrow{l}$, ће се стопити, али како више лема може бити придружено једној речи, и ове леме ће бити стопљене у исту класу. Квалитет класа стемова се мора експериментално потврдити.

4.2 Генерисање стемова и суфикса. У овом кораку морамо да идентификујемо шта су одговарајући стемови за сваку реч, а шта су одговарајући суфикси. Због оскудних ресурса који су нам на располагању примењује се једна ненадгледана метода учења. За сваку класу стемова налазимо најдужи заједнички префикс свих речи у класи и дефинишемо га као стем сваке речи из класе. Након тога за сваку реч из класе део речи који остаје проглашавамо за валидни суфикс. Фреквенције појављивања суфикса памтимо очекујући да ће високофреквентни суфикси бити одговарајући кандидати за правила за уклањање суфикса.

4.3. Сортирање суфикса по фреквенцији. Генерисани валидни суфикси се сортирају по фреквенцији ради одабира значајних суфикса. Док ће јако фреквентни суфикси вероватно бити корисни, суфикси чија је фреквенција мала, нпр. један, би требало одбацити не само да би се смањио броја правила, већ и да би се генерисала општија правила која не би покушавала да сложе речи које се случајно преклапају.

4.4. Генерисање суфиксних правила. Разматрамо неколико начина за генерисање суфиксних правила и свако од њих експериментално процењујемо. Разматрају се једноставна правила за уклањање суфикса, на пример, правила су облика $s \rightarrow \varepsilon$, где је ε празна ниска.

4.4.а Стемер са обухватањем заснован на учесталости. У првом приступу који се назива стемер са обухватањем заснован на учесталости, прво бирамо суфиксе који се појављују са фреквенцијом која прелази утврђени праг. Ови учестали суфикси се називају *валидни суфикси*, и они су кандидати за алгоритам за уклањање суфикса. Скуп свих валидних суфикса се обележава са S_v. Ако је валидни суфикс s_1 истовремено суфикс другог валидног суфикса s_2, онда се било која реч која се завр-

шава суфиксом s_2 такође завршава суфиксом s_1, па онда кажемо да је суфикс s_1 *обухваћен* суфиксом s_2 што пишемо $s_1 \sqsupseteq s_2$ или кажемо да је s_2 специфичнији од s_1. Ако два валидна суфикса могу да се уклоне из речи, онда је један *обухваћен* другим и онај специфичнији се уклања. У супротном, специфичнији афикс се никад не би применио. Осим тога, ово је принцип који се користи у свим стемерима који наликују Портеровом.

4.4.б Похлепни стемер са обухватањем. Правила за уклањање суфикса се бирају у складу са фреквенцијом суфикса у опадајућем редоследу, слично као у стемеру описаном у 4.4а. Додатни услов се примењује мерењем тачности уклањања новоформоране групе након сваког правила. Ако тачност није побољшана на одређеном прагу, правило се не одабира.

4.4в Оптимални суфиксни стемер. Присуство или одсуство суфикса се може користити на много сложенији начин него што нуде једноставна правила за уклањање суфикса. Нпр. нека правила Портеровог стемера се изражавају овако: „уколико реч има суфикс s_1 а не s_2, онда се уклања суфикс s_3". Циљ оптималног суфиксног стемера је да се истражи да ли се може постићи бољи учинак креирањем оваквих сложенијих правила, користећи притом само оне суфиксе генерисане у кораку 2. Такав оптимизацијски проблем није лако рачунски обрадити, али ми показујемо је то ипак могуће и имплементирамо ефикасан алгоритам који га решава. Кажемо да се две речи w_1 и w_2 не могу разликовати скупом валидиних суфикса S_v, што пишемо као $w_1 \equiv s_v\ w_2$ ако за сваки суфикс $s \in S_v$, s истовремено јесте или није суфикс речи w_1 и w_2. Ако се две речи не могу разликовати, онда се оне или мењају или остају непромењене истим правилом за уклањање суфикса у процесу уклањања. Поред тога, релација $\equiv s_v$ је релација еквиваленције и она дели скуп речи на $|S_v| + 1$ класу еквиваленције (или $|S_v|$ ако $\varepsilon \in S_v$). Ове класе еквиваленције су важне у контексту комплексних суфиксних правила јер се две речи из исте класе не могу одвојити сравњивањем са валидним суфиксима; а ако две речи припадају различитим класама онда се може креирати буловски израз над условима за сравњивање валидних суфикса да би се речи раздвојиле. Према томе, да би нашли оптималну оствариву тачност са скупом суфикса S_v, треба да локално оптимизирамо сваку класу еквиваленције тако што ћемо пронаћи најоптималнији суфикс који треба уклонити из сваке речи у класи. Ово се може ефикасно извести.

5 Евалуација

5.1 Лексички морфолошки ресурси

леме	47.489
облици речи	675.140
парови облик речи → лема	696.263

Табела 1: Статистика лексичког ресурса

Наша обрада почела је са основним лексичким ресурсом за српски језик који је ручно креиран и обогаћен применом деривационих правила. Ресурс је прошао кроз дуги процес прочишћавања, али у њему још увек има грешака. Да би олакшали обраду дијакритичка латинична слова српског језика су транскрибована у такозвано кодирање „dual1" (нпр. č=cx, ć=cy). Ресурс се састоји из парова реч → лема, а основни статистички подаци су приказани у табели 1. Код рачунања различите ознаке за врсту речи нису узимане у обзир. На пример, у енглеском се work/NN (именица) и work/VB (глагол) могу рачунати као две различите леме, али их ми сматрамо једном лемом. Оваква врста двозначност није превише честа у српском.

Можемо такође да приметимо да је број парова (облик речи, лема) већи од броја облика речи, али не много (≈ 3%). Ово значи да би Једноставни директни лематизер заснован на речнику (SDDL), који је описан у претходном

поглављу, могао бити прилично тачан, пошто се око 97% облика речи једноставно пресликава у једну лему. Може се приметити да просечно 14 различитих облика речи долази на једну лему.

5.2 Једноставни директни лематизер заснован на речнику (SDDL)

Битне карактеристике SDDL зависе од нивоа вишезначности речника, то јест, ресурса. Ниво вишезначности речи w дефинишемо као вишезначност$(w) = |\{l : w \xrightarrow{l} l\}|$, односно, број лема које су придружене речи. За једнозначне речи, то јест за речи чији је ниво вишезначности 1, лематизер би требало да дâ коректан одговор, бар према ресурсу. Дистрибуција нивоа вишезначности је дата у табели 2.

Ниво вишезначности	Број облика речи	Проценат
6	1	0,00015 %
5	18	0,0027 %
4	156	0,023 %
3	1566	0,23 %
2	17446	2,58 %
1	655953	97,16 %

Табела 2: Дистрибуција нивоа вишезначности облика речи у ресурсу

Ово упућује да, ако претпотавимо униформну расподелу речи, можемо да очекујемо да SDDL има тачност од најмање 97%. Највишезначнија реч у ресурсу, са 6 одговарајућих лема је „жуте”, а њене леме су: „жут”, „жута”, „жутети”, „жутити”, „жутјети”.

Евалуација заснована на корпусу. У претходној процени претпоставили смо униформну расподелу речи у тексту, што није реално. Речи су обично дистрибуирају према Зипфовом закону [4] који је степени закон расподеле, и који се веома разликује од униформне расподеле. Да би евалуација била реалнија користимо корпус текстова. За репрезентативни корпус свакодневног савременог језика изабрали смо колекцију чланака из недељника „Време” за период од пет година 2001-2005. Величина корпуса је 44МБ и садржи 6,6 милиона речи.

Прва употреба корпуса је да се процени покривеност нашим ресурсом односно проценат речи корпуса које су садржане у ресурсу. Након прве примене, нашли смо да је само 56% речи из корпуса пронађено у ресурсу, уколико се тражи поклапање малих и великих слова. Поред имена, речи имају почетно велико слово и ако су на почетку реченице и у насловима, тако да је покривеност 61% ако се не прави разлика између малих и великих слова. Преглед непрепознатих речи показује да је међу њима око 35% властитих имена. Друга битна група непрепознатих речи су везници и предлози који су врло фреквентни, а десило се да нису укључени у ресурс. Властита имена су група коју је тешко предвидети па не можемо да претпоставимо да би били покривени бољим ресурсом. Како год, имена имају сличне морфолошке обрасце као и заједничке именице што је додатни доказ да алгоритамски приступ има предности, као и да би се могао боље уопштити. Од ових 61% речи из корпуса, њих 50% (тачније 49,79%) су недвосмислене у ресурсу (вишезначност(w)=1). Ово је отприлике $50/61 \approx 82\%$ препознатих речи, што је мање оптимистична процена од оне која се добија за униформну расподелу. Ово указује да би SDDL имао прецизност од најмање 50%, а вероватно не много већу од 61%, под претпоставком да се користи нека једноставна стратегија за непознате речи.

1	457	(1,1%)	8	3946	(9,5%)
4	1436	(3,4%)	9	1494	(3,6%)
5	1703	(4,1%)	12	3962	(9,6%)
6	1320	(3,2%)	13	2433	(5,8%)
7	11942	(28,7%)	29	547	(1,3%)
31	2633	(6,3%)			
32	1481	(3,6%)			
33	2872	(6,9%)			
34	446	(1,1%)			
37	632	(1,5%)			

Табела 3: Расподела величина класа стемова, којих има више од 1% од свих класа стемова

Пре него што смо наставили са евалуацијом нашег метода за генерисање стемера, побољшали смо лексички ресурс на следећи начин. Десет најфреквентнијих речи које нису покривене ресурсом су: „и”, „у”, „на”, „за”, „су”, „а”, „не”, „од”, „са” и „о” а то су све врло фреквентне граматичке речи које су изостављене из ресурса само због тога што та врста речи није била укључена (везници: „и” и „а”, предлози: „у”, „на”, „за”, „од”, „са” и „о”, помоћни глагол „су” и прилог „не”). Пошто смо додали још 200 парова реч-лема, покривеност је порасла до 85% са 73% једнозначних речи из ресурса. Ово је употребљива тачност, али су ограничења да захтева скоро 700.000 парова реч-лема, нема могућност за уопштавање и вероватно садржи неке грешке које очигледно постоје у ресурсу.

5.3 Евалуација стемера

Корак 4.1: Креирање класа стемова. После транзитивног затворења 677.868 јединствених речи из ресурса је распоређено у 41.681 класу, што је у просеку 16,3 речи по класи. Број речи по класи варира између 1 и 307 речи по класи. Класе са више од 80 речи су врло ретке. На пример, две највеће класе стемова имају 307 и 283 речи. Након испитивања, увидели смо да су настале погрешним спајањем две или више правих класа стемова највероватније захваљујући неким погрешним паровима реч-лема. Најчешћа величина класе стемова је 7, што чини 29% свих класа стемова. Расподела величина оних класа којих има више од 1% од свих класа стемова приказана је у табели 3.

Корак 4.2: Генерисање стемова и суфикса. Након производње стемова и суфикса у овом кораку, сваки добијени празан стем био је индикатор неправилне класе стемова. У већем броју случајева, то је проузроковао префикс „нај-” који се користи за облике суперлатива прилога и придева. Као што смо већ рекли, деривације произведене префиксима не треба узимати у обзир код стеминга. На пример, ако претражујемо колекцију докумената у потрази за највишим планинским врхом на свету, засигурно нас интересује тачно „највиши”, а не „високи врх” нити компарација „виши врх”, иако би се они у енглеском стопили (high, higher, highest). Уклањање префикса „нај-” довело би до додатних грешака пошто се он појављује као префикс и у речима које нису у суперлативу, какве су „најамник” и „најахати”. Овај проблем би се могао решити ако би се уклањао префикса „нај-” само када се сравњује истовремено са одговарајућим суфиксом за суперлативе „-ија”, „-ији” и слично. Овај проблем решавамо тако што раздвајамо суперлативне од несуперлативних класа стемова.

Други извор празних стемова су неправилне флексије, какве су множина „људи” именице „човек” или „човјек” или облик помоћног глагола „ћеш” од „бити”. За оба случаја решење може бити листа изузетака, али смо ми одлучили да их издвојимо у одвојене класе стемова. Претпостављамо да корисник система за проналажење информација ионако не би очекивао да се термини за претрагу проширују на овај начин (нпр. „људи” за „човек”), или би се у сваком случају помоћни глаголи уклонили стоп листом.

Стемови дужине 1 потенцијално указују на неправилне класе, али они нису систематски уклоњени. Један од примера је реч „беже” што је облик садашњег времена глагола „бежати”, али и облик вокатива именице „бег” што доводи до погрешног стапања две, иначе правилне, класе стемова. У првом пролазу, 650 речи је произвело празан стем. За све њих смо ручно преправили оригинални ресурс, што је довело до деобе одговарајућих класа стемова и до стварања непразних стемова. Кратки стемови (нпр. дужине 1) такође се често стварају неправилним стапањем класа стемова, али смо претпоставили да можда није неопходно да и њих исправљамо у овом експерименту, пошто ка-

сније методе користе најучесталије суфиксе, који би требало да су веома поуздани. Метода најдужег заједничког префикса која је коришћена за генерисање стемова довела је до додатног преклапања између класа стемова, заправо их стапајући: створено је 39.289 стемова, од којих је 1.823 (4,6%) вишезначно у смислу да су придружени већем броју класа стемова. Само 253 је имало ниво вишезначности од три или више, при чему ниво вишезначности стемова брзо опада када се они сортирају у опадајућем редоследу. Стемови чија је вишезначност највећа дати су у следећој листи.

43 ist	18 post	16 samo	14 ekst
26 rast	18 sat	15 ust	13 zast
12 ost	7 pos	7 nast	
12 konst	7 podst	7 nas	

Код ових стемова чија је вишезначност велика није задржано значење речи, па је унапређење овог корака део нашег даљег рада.

Корак 4.3: Сортирање суфикса по фреквентности. У овом кораку је генерисано 18.274 суфикса, а врх сортиране листе генерисаних суфикса са фреквенцијама, дат је у наредној табели.

24833 -e	6495 -oj
22874 -u	6475 -omu
22389 -i	6121 -oga
22184 -a	6118 -og
19475 -om	5929 -ti
17756 -o	5775 -t
16190 ‘’ (empty)	4412 -h
	4399 -m
8996 -im	4303 -суесх
8281 -ama	4289 -суu
8101 -ih	4273 -le
7573 -te	4272 -la
7472 -ima	4268 -li
6821 -mo	4252 -суе

Сви ови суфикси имају линвистичку интерпретацију.

Корак 4.4: Генерисање суфиксних правила. Директна имплементација евалуације различитих приступа генерисању суфиксних правила води ка веома спорој евалуацији. Једна успешна имплементација са компактним префиксним дрветом (такође познато и као дрво *Патриција*) са обрнутим нискама значајно је смањила време извршавања, са 5-6 сати за почетни експеримент на 5-10 минута.

(4.4а) Стемер са обухватањем заснован на фреквентности. За овакав стемер, почели смо са празним скупом валидних суфикса и додавали му постепено једно по једно правило по редоследу одређеном фреквенцијским правилом. Након сваког стема мерена је тачност стеминга према нашим генерисаним стемовима. Тачност је на почетку била 2,4% за $S_v = \varnothing$ и постепено се повећавала до 56,3% за 98 суфиксних правила, а затим је постепено опадала све до 14,2% када је било укључено свих 17.839 суфикса.

(4.4б) Похлепни стемер са обухватањем. У овом приступу додавали смо правила истим редом као у 4.4а, али смо пре и после додавања сваког суфикса мерили тачности A_1 и A_2 у броју правилних стемова. Правило се прихвата ако је $A_2 - A_1 > \Theta$, где је Θ задати параметар, тј. суфикс се прихвата само ако је побољшање тачности веће од задатог прага. На пример, ако је $\Theta = 0$, онда се суфикс прихвата само ако не смањује укупну тачност, ако је $\Theta = 1$, онда се број правилних стемова мора повећати бар за 1, итд. Што је параметар Θ већи, очекује се боље уоштавање избором мањег броја правила која су квалитетнија, али могу смањити перформансе. Резултати су приказани у следећој табели.

Θ	Исправни суфикси	Прецизност
0	9849	74,15
1	8633	74,16
2	3367	73,38
3	1901	72,95
4	1557	72,83

Θ	Исправни суфикси	Прецизност
5	1262	72,66
6	1124	72,56
7	1002	72,46
8	933	72,39
9	878	72,32
10	831	72,26
15	673	71,99
20	592	71,78
25	497	71,48
30	453	71,30
35	423	71,16
40	410	71,09
45	380	70,90
50	360	70,76
60	347	70.65
70	319	70.39
80	310	70.29
90	298	70.14
100	294	70.23
150	273	69.87
200	230	68.80
250	218	68.43
300	202	67.77
350	188	67.08
400	180	66.65
450	179	66.59
500	175	66.31
600	131	62.74
700	121	61.82
800	114	61.03
900	87	57.76
1000	85	57.48

Два интересантна закључка која се одавде могу извести су да је тачност далеко већа него са претходним приступом, и да тачност опада врло споро у почетку док број правила опада брзо, што је друго врло охрабрујуће запажање. У случају када је $\Theta = 7$ добијамо 1002 суфиксна правила са тачношћу која је за само око 1,7% мања од најбоље. Ово потврђује нашу процену да ће стемеру за српски језик бити потребно око 1000 правила.

(4.4в) Оптимални суфиксни стемер. Тачност оптималног суфиксног стемера је 81,83%. Ово је горња граница која се може постићи са добијеним скупом валидних суфикса и одговарајућим правилима за уклањање суфикса, када се процењују на произведеном скупу стемова. Видимо да похлепни приступ и није толико лошији, посебно ако се има у виду да наш циљ не би требало да буде постизање оптималне тачности јер бисмо на тај начин обухватили и почетне грешке лексичких ресурса као и неке неправилне стемове произведене у већ описаном процесу.

5.4 Непристрасна евалуација

Да би евалуирали стемер на непристрасан начин користимо корпус вести, примењујемо стемере на узорку скупа речи из корпуса и ручно оцењујемо добијене стемове. Одабрали смо за процену два стемера: оптимални суфиксни стемер (4.4в) и похлепни стемер (4.4б) са параметром $\Theta = 7$ и 1000 генерисаних правила. Интерактивни програм чита редом речи из корпуса и на њих примењује оба стемера. Како смо више заинтересовани за речи које не постоје у ресурсу, оне речи које постоје у ресурсу а за које стемери производе исти стем игноришемо. Сви остали стемови пролазе ручну евалуацију која има четири избора: тачан само за похлепни стемер, тачан само за оптимални стемер, тачан у случају оба стемера, погрешан у случају оба стемера и игнориши. Опција „игнориши" се користи да би се искључиле неке функционалне речи за које је очигледно да су речи за одбацивање и неке енглеске речи које се појављују у корпусу. За стем се процењује да је правилан ако се оригинално значење може јасно из њега предвидети (нема претераног уклањања), и изгледа да он покрива све морфолошке варијације леме (нема премалог уклањања). Након евалуације 1000 „не-игнорисаних" речи из корпуса (са могућим понављањем)

резултат је био: 127 правилних речи само за похлепни стемер, 90 само за оптимални стемер, 663 правилних речи за оба стемера и 120 неправилних за оба. Ови резултати потврђују наше две претпоставке: (1) Произведени стемери изгледају употребљиви за проналажење информација (тачност похлепног стемера је 79% а тачност оптималног стемера је 75%); и (2) похлепни приступ не само што даје исто тако добре резултате као и оптимални, него даје и бољу тачност са само 1000 правила.

6. Закључак и даљи рад

Описали смо и проценили општи, углавном аутоматски, приступ генерисању стемера за језике са богатом флексијом и само неколико ресурса. У току рада су откривена нека ограничења процеса, али и могућности за даља побољшања. Коначна евалуација је дала тачност од 79% на веродостојним подацима за похлепни стемер, што је чак и нешто више од тачности добијене на подацима за обуку, показујући веома добре способности за уопштавање.

Неки правци будућег рада су: (1) евалуација на више података, (2) примена и правила за замену суфикса а не само правила за уклањање суфикса, и (3) укључивање параметра за дужину стема. Са правилима за замену суфикса, метода се може директно применити на генерисање лематизера.

[1]Званична веб станица Портеровог стемера је http://tartarus.org/˜martin/PorterStemmer/, и она је веродостојни извор за имплементације оригиналног стемера. За брзу проверу аутентичности Портеровог стемера користи се реч ‘agreement’ – у оригиналном Портеровом стемеру она се не мења, док је неке некоректне имплементације мењају.

[2]CPAN – Comprehensive Perl Archive Network, http://cpan.org/, је open-source репозиториј за Перл.

[3]http://www.mf.uni-lj.si/ds/new-stemmers.html

[4]http://snowball.tartarus.org/archives/snowball-discuss/0722.html

[5]http://svn.rot13.org/index.cgi/stem-hr

[6]Извор: *Snowball* листа слања.

Литература

[1] K. Beesley and L. Karttunent. Finite State Morphology. CSLI, 2003.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction fo Algorithms. The MIT Press, 2nd edition, 2002.

[3] Chris Jordan, John Healy, and Vlado Kešelj. Swordfish: An unsupervised ngram based approach to morphological analysis. In SIGIR’06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 657–658, Seattle,Washington, USA, August 2006. ACM Press.

[4] Daniel Jurafsky and James H. Martin. Speech and Language Processing. Prentice Hall Series in Artificial Intelligence. Prentice Hall, 2000.

[5] Mirko Popovič and PeterWillett. The effectiveness of stemming for naturallanguage access to Slovene textual data. Journal of the American Society for Information Science, 43(5):384–390, 1992.

[6] Martin F. Porter. An algorithm for suffix stripping. Program, 14(3):130–137, July 1980.

[7] Martin F. Porter. Snowball: A language for stemming algorithms. Published on WWW, October 2001. Last access in April 2007.

[8] S. Sheremetyeva, W. Jin, and S. Nirenburg. Rapid deployment morphology. Machine Translation, 13(4):239–268, 1998.

[9] Marko Tadić. Hrvatski lematizacijski poslužitelj. Published onWWW, 2005. Last access in April 2007.

[10] Duško Vitas and Cvetana Krstev. Derivational morphology in an e-dictionary of Serbian. In Proceedings of 2nd Language & Technology Conference, pages 139–143, Poznan, Poland, April 21–23 2005.