Универзитет у Београду
Електротехнички факултет

Милош Н. Павковић

Побољшање перформанси прикупљања кориснички генерисаних садржаја на Вебу применом адаптивних интелигентних метода

Докторска дисертација

Београд, 2020.

UNIVERSITY OF BELGRADE
SCHOOL OF ELECTRICAL ENGINEERING

Miloš N. Pavković

PERFORMANCE IMPROVEMENT OF USER-GENERATED DATA RETRIEVAL FROM THE WEB, BASED ON ADAPTIVE INTELLIGENT METHODS

Doctoral dissertation

Belgrade, 2020.

Ментор:

др Јелица Протић, редовни професор

Универзитет у Београду – Електротехнички факултет

Чланови комисије:

Др Јелица Протић, редовни професор

Универзитет у Београду – Електротехнички факултет

Др Бошко Николић, редовни професор

Универзитет у Београду – Електротехнички факултет

Др Милош Ковачевић, редовни професор

Универзитет у Београду – Грађевински факултет

Датум одбране:

_____ године.

Захвалнице

У нади да сам допринео испуњењу циља који сваки докторат представља, желим да искористим прилику и да се искрено захвалим изузетним особама које су биле уз мене на овом путу и својим значајним сугестијама, знањем и залагањем, помогле у изради дисертације.

Дисертација је резултат дугогодишњег рада са ментором, др Јелицом Протић, редовним професором Универзитета у Београду - Електротехничког факултета, којој се искрено захваљујем на великој посвећености, подршци и непроцењивој помоћи у свим фазама израде. Својим перфекционизмом, драгоценим сугестијама и неуморним корекцијама, допринела је да ова дисертација добије свој коначни облик. Сматрам великом срећом што сам имао прилику да сарађујем са ментором који поседује изузетне стручне референце и добру вољу да своје знање и искуство несебично подели са мном.

Добар део резултата приказаних у овој дисертацији проистекао је из сарадње са компанијом *SocialGist Inc.* у Детроиту, где тренутно и радим. Посебно се захваљујем свим колегама у овој компанији што су ми омогућили да кроз интересантне пројекте стекнем неопходне резултате потребне за реализацију делова ове тезе. Из ове сарадње су проистекле идеје које су касније послужиле као основа за развој модела потребних за овај рад и који су у великој мери допринели резултату ове дисертације.

Додатно се захваљујем колегама из партнерске институције *HottoLink Inc.* из Јапана, који су ми омогућили да стекнем додатна теоријска и експериментална знања о машинском учењу. Такође им се захваљујем на искуственим саветима који су допринели одговарајућем представљању резултата истраживања.

Својим пријатељима и колегама са Института за математику и информатику, Природно-математичког факултета, Универзитета у Крагујевцу, који су употпунили део активности које нису биле саставни део дисертације, желим посебно да се захвалим јер је због њих било вредно радити и завршавати са послом на време.

Свим члановима комисије хвала на труду уложеном у читање и датим смерницама приликом коначног уобличавања докторске дисертације.

Мајци се нарочито захваљујем на томе што је учинила да школовање и одрастања буде тако посебно. Захваљујем се мојој породици на неизмерном разумевању и стрпљењу, која је за све ове године увек нашла речи подршке и топлине, била уз мене, уливала ми самопоуздање и пружала безусловну љубав.

Резиме

Наслов: Побољшање перформанси прикупљања кориснички генерисаних садржаја на Вебу применом адаптивних интелигентних метода

Сажетак: Кориснички генерисан садржај на веб форуму се много чешће додаје него што се брише или мења па се самим тим, циљање истог, приликом инкременталног претраживања, разликује у односу на класично претраживање страна веб сајта. Додавање новог садржаја на форуму може резултовати померањем већ постојећег садржаја на нове или постојеће стране. Инкрементално претраживање форума није тривијалан задатак, јер игнорисање начина на које је садржај презентован, дистрибуиран и сортиран може довести до преноса постова који су већ били индексирани у претходним циклусима претраживања. С друге стране постоји широк спектар форумских технологија које омогућавају различите навигационе путање ка својим најновијим постовима као и различите начине презентовања и сортирања истих.

Један од главних резултата тезе је структурно вођени инкрементални претраживач форума (SInFo) који је специјализован за циљање најновијег садржаја приликом инкременталног претраживања коришћењем напредних оптимизационих техника и машинског учења. Главни циљ представљеног претраживача јесте избегавање већ индексираног садржаја у новим циклусима претраживања форума без обзира на његову технологију. Да би овај циљ могао бити испуњен, следеће карактеристике веб форума су искоришћене: (1) начин сортирања на индексним и дискусионим странама и (2) доступне навигационе путање између страна које тренутна веб форумска технологија нуди. С обзиром на то да приликом утврђивања типа сортирања битну улогу има датум креирања садражаја, детекција и нормализација истих није једноставан задатак. За овај задатак су коришћени модели машинског учења, јер генерисани датуми могу бити у различитим форматима и на различтим језицима. С друге стране, детекција навигационих путања се постиже интерпретацијом формата URL линкова и скенирањем страна на које они указују.

Показано је да се коришћењем предложених метода и техника, приликом циљања страна са најновијим садржајем, минимизује број преузимања дуплираног садржаја и максимизује искоришћеност навигационе структуре и путања тренутне форум технологије. Експерименти су изведени на широком спектру већ постојећих популарних форумских технологија као и на индивидуалним *stand-alone* форумским технологијама. SInFo је показао високу прецизност и минималан број преноса дуплог садражаја у сваком новом циклусу претраживања. Већина дупликата на које је предложени претраживач наилазио је са страна које су морале бити посећене како би се исправно утврдила навигациона путања или пронашао одговарајући URL. Додатно, модели машинског учења, иако су комплексни постижу добре перформансе приликом претраживања и имају високу прецизност у детекцији и нормализацији датума, достижући F1-меру од 99%.

Кључне речи: Технике претраживања, Сакупљање података, Машинско учење, Инкрементално претраживање, Оптимизација, Стратегија веб претраживања, веб форуми

Научна област: Електротехничко и рачунарско инжењерство

Ужа научна област: Софтверско инжењерство

УДК број: 621.3

ABSTRACT

Title: Performance improvement of user-generated data retrieval from the Web, based on adaptive intelligent methods

Abstract: User-generated content on Web forums is added much more often than it is deleted or changed, so its targeting during incremental crawling differs from the Web site pages crawling. Adding new content to a forum can result in moving existing content to new or existing pages. Incremental forum crawling is not a trivial task, because ignoring in which way the content is presented, distributed and sorted can lead to the transfer of posts that have already been indexed in the previous crawl cycles. On the other hand, there is a wide spectrum of forum technologies that allow different navigational paths to its latest posts, as well as different ways of presenting and sorting user generated content.

This thesis presents Structure-driven Incremental Forum crawler (SInFo) that specializes in targeting the latest content in incremental forum crawling using advanced optimization techniques and machine learning. The main goal of the presented system is to avoid already indexed content in new crawling cycles regardless of its technology. In order to achieve this, the following Web Forum features have been used: (1) the sort method on the index and thread pages and (2) the available navigation paths between the pages that the current Web Forum technology offers. Since the date of content creation plays an important role in determining the type of sort, their detection and normalization is not a trivial task. Machine learning models were used for this task, because the generated dates can be in different formats and in different languages. On the other hand, the detection of navigational paths is achieved by interpreting the URL format and scanning the pages they target.

It has been shown that using the proposed methods and techniques while targeting pages with the latest content can achieve a minimum number of duplicate content downloads and maximize the utilization of the navigational structure and paths of the current forum technology. The experiments were performed on a wide range of already existing popular forum technologies as well as on individual stand-alone forum technologies. SInFo has demonstrated high precision and a minimum number of duplicate content transfers in each new crawl cycle. Most of the duplicates that the proposed system encountered are from pages that had to be visited in order to correctly determine the navigational path or to find the appropriate URL. Additionally, machine learning models, although complex, achieved good performance while crawling and have high accuracy in date detection and normalization, reaching an F1-measure of 99%.

Keywords: Crawling technique, Data retrieval, Machine learning, Incremental crawling, Optimization, Traversal strategy, Web forums

Scientific field: Electrical and Computer Engineering

Scientific subfield: Software Engineering

UDC number: 621.3

САДРЖАЈ

1 Увод

Платформе за дискусије на интернету, као што су веб форуми, “Q&A” (*Question and Answer*) сајтови и сајтови са корисничким коментарима, данас представљају значајан део друштвених веб страница. Користећи дискусионе платформе, корисници могу да размењују мишљења о одређеним темама и проблемима, да учествују у опсежним полемикама или да проналазе одговоре на нека стручна питања [1]. Када су се појавили, форуми су претежно представљали средство друштвене комуникације. Настанком друштвених и медијских мрежа дошло је до промене првобитне намене оваквих сајтова, који су еволуирали у специјализоване алате за размену мишљења експерата. Дискусионе платформе данас представљају најбољи медиј за многе стручне размене информација о темама и питањима од општег интереса. Док су друштвене и медијске мреже преузеле један део улоге веб форума, сами форуми су прерасли у специјализоване јавне базе података које садрже исказана стручна мишљења у вези одређених тема и проблема.

Веб форум, као један од најзаступљенијих облика дискусионе платформе, веома је погодан за анализу и примену разних техника претраживања и прикупљања информација. На форумима, кориснички генерисан садржај се скоро никада не мења нити брише, него се стално додаје. Из тог разлога је неопходан ефикасан инкрементални претраживач који проналази и прикупља искључиво нове садржаје у сваком циклусу претраживања, како би пропусни опсег, време прикупљања података и укупне перформансе система биле оптималне. Иако су уложени одређени напори у области истраживања и развоја специјализованих форумских претраживача [2]–[4], ниједан од њих се није директно усредсредио на инкрементално претраживање форума, тј. на то како циљати само садржај генерисан након последњег циклуса претраживања. Поновну посету веб форуму, у потрази за новим садржајем, треба свести на откривање новоотворених тема и њихових постова од момента последњег претраживања форума (покривеност), те на претраживање тема које су пронађене у претходним посетама (свежина), у потрази за новим постовима [4].

Ефикасно претраживање новог садржаја на веб форуму, како би се смањило време прикупљања и растеретио пропусни опсег, није једноставан задатак из разлога што се мора идентификовати и прескочити садржај прикупљен у претходним циклусима претраживања. Циљ овог рада је конципирање и развијање инкременталног веб претраживача, примењеног на дискусионе платформе, а посебно форуме, који кластеризацијом веб страна и применом техника машинског учења аутоматски прати скелетне линкове форума, врши екстракцију кориснички генерисаног садржаја и интелигентно се адаптира технологији коју конкретан форум користи, тако да време, проток и ресурси рачунара буду оптимално искоришћени.

У оквиру овог рада биће представљено свеобухватно и опсежно истраживање на великом броју веб форумских технологија, у циљу категоризације главних врста репрезентација које могу имати веб стране форума. Биће извршена анализа на скупу репрезентативних форума са великим бројем корисника и кориснички генерисаног садржаја, како би се дошло до исцрпног прегледа форумских репрезентација и њихових навигационих путања.

Циљеви оптимизације инкременталног претраживања веб форума су: (1) боља искоришћеност пропусног опсега, (2) смањено време извршавања, (3) мање дуплираног садржаја и (4) боља искоришћеност ресурса система на којем се претраживач извршава. Посећивање страна са већ преузетим садржајем у неком од претходних инкременталних

циклуса проузрокује непотребну потрошњу пропусног опсега и продужава време извршавања. Коришћењем метода и алгоритама који ће бити представљени у овом раду, где се циљано прескаче већ претходно посећени садржај, значајно се оптимизује процес претраживања.

Проблем циљања искључиво новог садржаја приликом инкременталног претраживања своди се на проблем детекције начина на који је тај садржај презентован и повезан. Овај једноставан а у исто време робустан и скалабилан приступ омогућава лаку примену на различите дискусионе технологије и већ постојеће веб претраживаче. Такође, искоришћеност постојећих навигационих путања и опција које технологија која презентује кориснички генерисан садржај поседује, се на овај начин настоји максимизовати. Ово омогућава прецизније циљање новог генерисаног садржаја.

У оквиру дисертације биће дат детаљан преглед и класификација већ постојећих специјализованих претраживача веб форума. Иако ови претраживачи нису дизајнирани да се користе у инкременталним стратегијама, представљају добру основу за имплементацију инкременталних претраживача, поређење и саму евалуацију у експериментима.

Резултати су добијени у сврху испитивања скалабилности и евалуације перформанси добијеног решења симулационом техником генерисања форумског садржаја, у различитим временским периодима и на великом броју различитих форумских технологија.

Остатак рада је организован у оквиру седам поглавља у којима је описана терминологија, дефиниција проблема, преглед стања постојећих технологија и могућих делимичних решења, као и представљање целокупног SinFo система који је главни резулат ове дисертације.

У оквиру другог поглавља – Терминологија, дат је преглед најчешће коришћених термина, инспирисаних претходним радовима о форумима и форумским претраживачима, а тичу се њихових делова.

У трећем поглављу дата је дефиниција проблема инкременталне стратегије претраживања форума, циљања најновијег садржаја, као и опис проблема са генеричким претраживачима приликом претраживања у инкременталним концептима. Такође је направљен осврт на структуру и организацију форума, као и на начине приказа и сортирања новонасталог садржаја који су битни приликом претраживања.

Постојеће форумске технологије представљене су у четвртом поглављу, које пружа осврт на основне карактеристике популарних форумских технологија, њихов опис, као и специфичности приликом претраживања. На крају ове главе су описане тренутно популарне форумске технологије, као и преглед њихових предности и мана.

Пето поглавље садржи преглед претходних радова и осврт на доприносе у области не само инкременталног претраживања него и прегледа процеса прикупљања података, као и осврт на најсавременије специјализоване претраживаче форума. Неки од представљених радова су коришћени у овом раду као основа за предложену архитектуру.

Шесто поглавље даје приказ предложеног система и представља детаљан преглед предложене архитектуре, њених модела машинског учења, метода и алгоритама. За сваки од метода је дат алгоритам представљен псеудокодом, док су за моделе машинског учења приложени детаљни прикази њихових архитектура.

Седмо поглавље сумира извршене експерименте и представља детаљну евалуацију предложеног система кроз симулацију инкременталне стратегије претраживања и генерисања садржаја форума у контролисаним временским интервалима. Модели машинског учења су евалуирани кроз тестне скупове, пре него што су коришћени у евалуацији претраживача.

Осмо поглавље садржи закључак и правце даљег развоја и потенцијалног будућег рада на тематици из ове докторске дисертација.

2 Терминологија

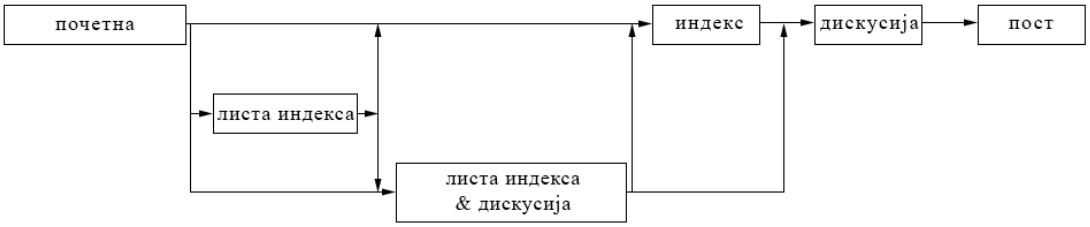
У оквиру ове дисертације употребљена је уобичајена терминологија која је део области која се изучава. Следећи појмови су коришћени у даљем излагању и инспирисани су терминима из радова [2], [3].

- **Веб форум**. Под термином веб форум се подразумева веб сајт или секција веб сајта која служи за размену комуникације између корисника. Сваки веб форум има технологију тј. софтверски пакет који га покреће и који може бити јединствен само за тај конкретан веб форум, или може бити шаблонски, односно већ нека од познатих веб форумских технологија.
- **Скелетни линкови**. URL-ови који повезују искључиво важне стране веб форума, као што су почетна, индекс или дискусиона страна. Уколико се ове стране сматрају чворовима, а линкови који их повезују гранама, добија се усмерени граф чија структура представља скелет форума. Пример скелета форума се може видети на Слици 3.1
- **Пост**. Блок на дискусионој страни који садржи кориснички генерисан садржај. Поред овог садржаја, у зависности од технологије форума, могу се налазити још и додатне информације као што су датум креирања поста, аутор који га је креирао, његови датуми регистрације и последње активности и сл.
- ***Sticky* Пост**. Истакнути пост на дискусионој страни за који је, од стране администратора форума, процењено да има већи значај. Овај пост се обично налази на почетку дискусионе стране пре свих осталих постова, без обзира на начин представљања и сортирања садржаја.
- Типови стране на веб форуму:
  - **Почетна**. Почетна страна је улазна страна веб форума. Ова страна не мора бити иста као и почетна страна веб сајта где се налази веб форум. Може бити на различитом поддомену или чак другој веб адреси тог веб сајта. У хијерархији страна веб форума, улазна страна се рачуна као најстарији потомак свих индексних или дискусионих страна.
  - **Индексна страна (Индекс)**. Индексна страна, или *board*, је секција веб форума која одваја одређене тематике и групише их у једну логичку целину. Један веб форум може имате више индекса, а сваки индекс може имате своје под индексе ако је потребна финија подела тематике којом се бави над индекс. Индексна страна садржи листу дискусија и/или листу под индекса и обично има изглед налик структури табеле. Слика 3.2 представља пример индексне стране.
  - **Дискусија или тема**. Дискусија представља страну која на себи садржи корисничке коментаре (постове) који припадају одређеној индексној страни. Слика 3.3 приказује пример дискусионе стране.
  - **Остале**. Остале стране су стране форума које нису индекс или дискусионе стране. Ове стране могу бити често постављана питања, страна за логовање и сл.
- Типови URL-ова на веб форуму:
  - **Индекс URL**. URL који показује на прву индексну страну и налази се на почетној страни или некој другој индексној страни.

- **URL дискусије или URL теме**. URL који показује на прву страну дискусије и налази се на индексној страни.
- **Пагинациони URL**. URL који служи да повеже стране индекса или дискусије у једну логичку целину и може се називати још и URL за страничење. Пратећи ове линкове остаје се на истом индексу или дискусији, само се прелази на њихову другу страну. Примери ових URL-ова се могу видети на Слици 3.2а, Слици 3.2б и Слици 3.3а.

3 Дефиниција проблема

Проблем са генеричким претраживачима је то што имају тенденцију да третирају сваку страну веб сајта као појединачни објекат који поновно посећују током времена, замењујући стари садржај базе података са новим прикупљеним. Овај приступ није адекватан за дату структуру онлајн дискусије као што је веб форум и његову презентацију садржаја. Слика 3.1 приказује скелетну структуру форума и показује да је, без обзира на технологију форума, његова логичка структура увек дефинисана као хијерархија са имплицитним путевима, представљеним синтаксним дијаграмом који се састоји од важних страна [5]. Сви чворови између почетне и дискусионе стране називају се индексним странама [3].

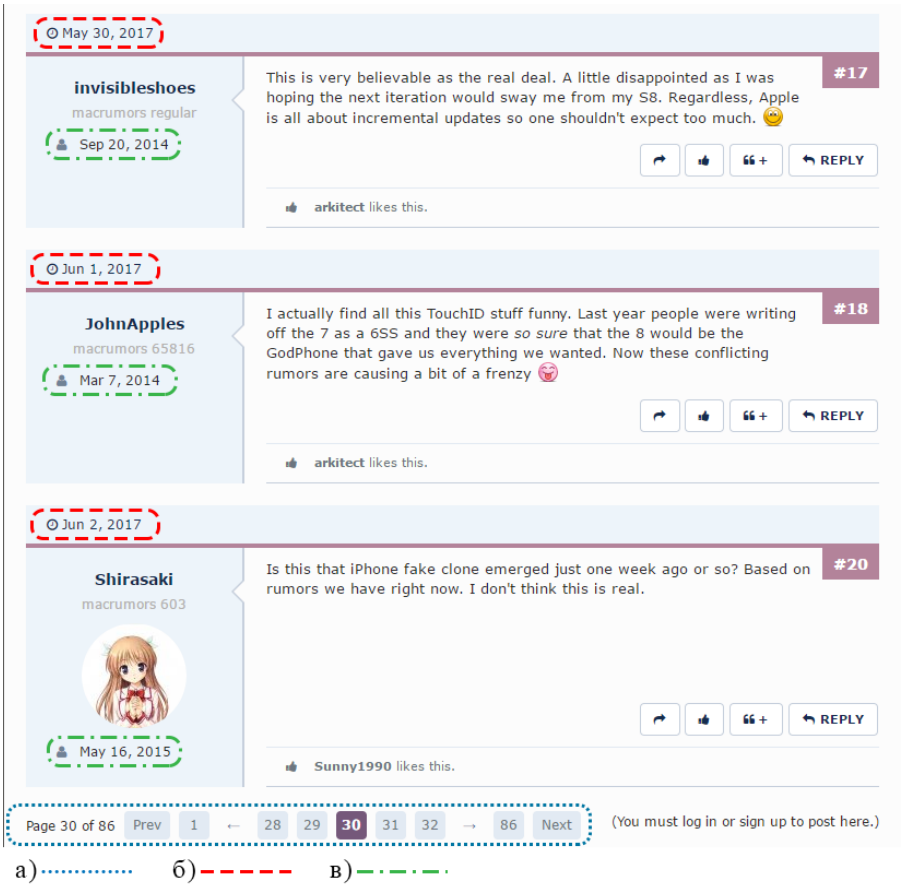


Слика 3.1 – Дијаграм структуре форума и његових имплицитних путања – скелетна структура

Индексне стране су репрезентоване структурама налик табелама, где сваки ред садржи информације о индексу или дискусији. Терминални чвор у графу структуре форума садржи постове са кориснички генерисаним садржајем из дискусије. Још једна карактеристика веб форума је да се његов раније генерисани садржај ретко мења.

Слика 3.2 – Пример индексне стране са ubuntuforums.org а) линкови за пагинацију индексне стране б) линкови за пагинацију дискусије в) URL стране ка последњој активности теме

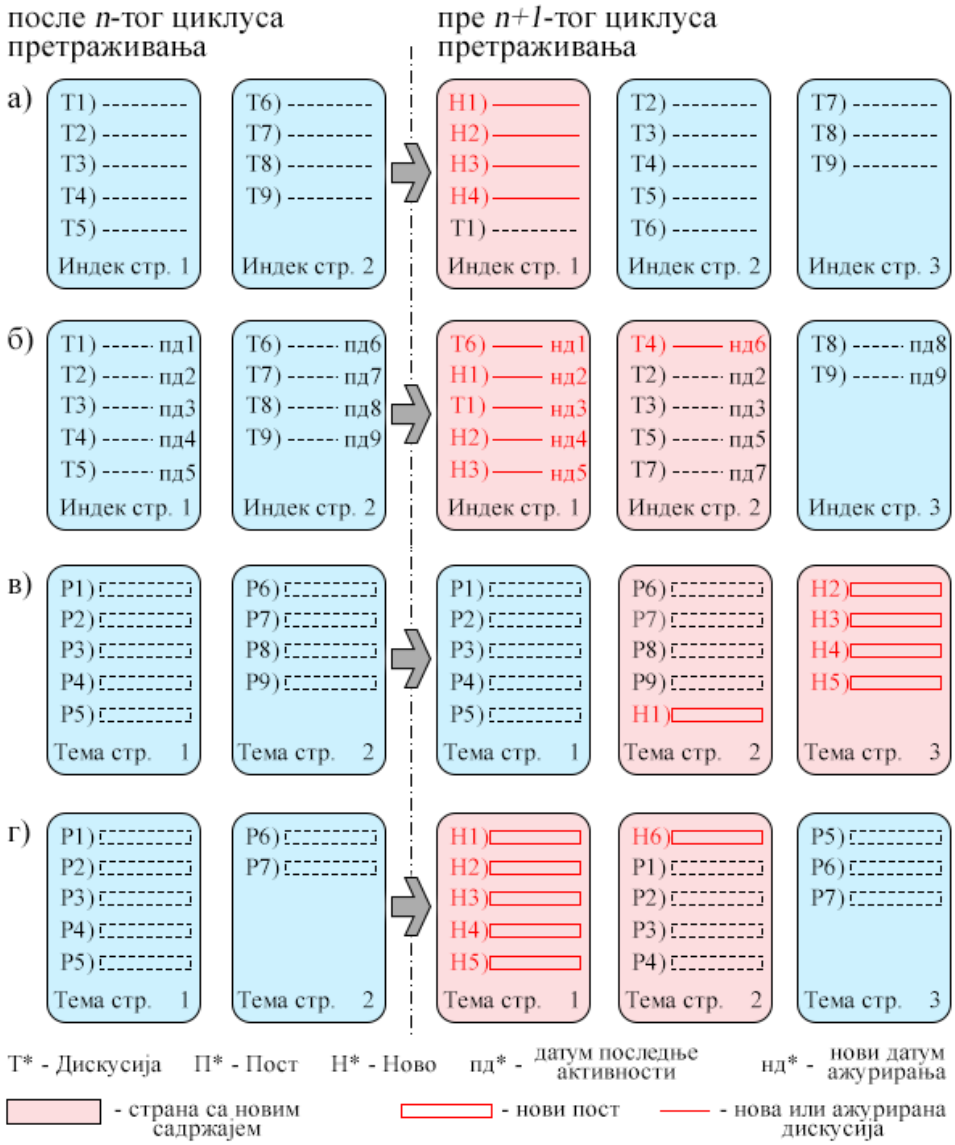
Садржај форума се временом углавном само додаје, дели и организује између више страна повезаних URL-овима за страничење [6], [7] тј. пагинационим линковима. Као што је приказано на Слици 3.2, дискусије које припадају једној индексној страни могу бити дистрибуиране између више страна које су повезане URL-овима за страничење. Такође, постови који припадају једној дискусији могу бити дистрибуирани на више страна те исте дискусије и повезани са URL-овима за страничење - Слика 3.3а и Слика 3.2б. Специјализовани претраживачи форума, као што су FoCUS (*Forum Crawler Under Supervision*) [3] и iRobot [2], су дизајнирани за преузимање само важних страница са оригиналним корисничким садржајем од интереса, праћењем искључиво скелетних линкова веб форума. Не претражујући остале неважне странице (као што су стране за пријављивање, претраживање, корисничке или странице помоћи итд.), ови претраживачи су ефикаснији у смислу пропусног опсега и покривености од генеричких претраживача приликом претраживања веб форума први пут.



Слика 3.3 – Пример дискусије са постовима на forums.macrumors.com. а) линкови за пагинацију тренутне дискусије б) датум креирања поста в) датум регистрације корисника на форуму

Специјализовани претраживачи FoCUS и iRobot не решавају проблем циљања најновијег садржаја веб форума, већ се више фокусирају на прикупљање искључиво важног садржаја. Ови претраживачи су ефикасни у избегавању дуплог и небитног садржаја само приликом првог претраживања, када се иницијално пролази кроз читав форум. У инкременталној стратегији претраживања, приликом поновне посете форуму, не узимање у обзир начина на који је садржај представљен и сортиран, може довести до посета странама чији је садржај већ био индексиран у неким од претходних циклуса. Директан резултат оваквог рада претраживача је већа оптерећеност пропусног опсега, дуже време претраге и могући вишеструки пренос садржаја.

Тренутно је у употреби широк спектар софтверских пакета за форуме [8], [9], укључујући неке тренутно веома популарне као што су *phpBB, Vanilla* и *SMF*. Бројни веб форуми имају индивидуалне, само за њих развијене софтверске пакете који их покрећу, тако да се међу форумима могу пронаћи различита решења изгледа страна, сортирања, презентације садржаја и навигационих путања између страна. Спроводећи опсежну анализу великог броја технологија веб форума, извршена је класификација могуће презентације садржаја на основу изгледа и сортирања индексних и дискусионих страна.



Слика 3.4 – Примери индексне и дискусионе стране између два сукцесивна претраживања форума. а) нове дискусије на индексној страни сортиране по датуму креирања у опадајућем поретку. б) нове и старе ажуриране дискусије на индексној страни сортиране по датуму последње активности у опадајућем редоследу в) нови постови на дискусији поређани у растућем поретку г) нови постови на дискусији поређани у опадајућем поретку

Дискусије са индексне стране могу бити сортиране по датуму креирања или датуму последње активности. Узимање начина сортирања у разматрање је од велике важности приликом детекције нових дискусија за време претраживања.

Слика 3.4а приказује пример где су дискусије са индексне стране поређане по опадајућем датуму креирања. Између два циклуса претраживања, нове дискусије се појављују само на првим индексним странама и потискују старе дискусије на следеће индексне стране. Приликом посете оваквом типу сортираног индекса, претраживачу је довољно да посети само пар првих индексних страна на којима су лоциране нове дискусије. Случај где су дискусије на индексној страни сортиране по опадајућем датуму последње активности је приказан на Слици 3.4б. Између два сукцесивна циклуса претраживања форума, нове дискусије су креиране, а неке од постојећих дискусија су имале активност, што даје микс нових и старих дискусија на првим индексним странама.

Слично као и у претходном случају, само првих неколико индексних страна је потребно посетити јер оне садрже и старе и нове дискусије, али са новим садржајем. С друге стране, постови на дискусији могу бити сортирани у опадајућем или растућем поретку датума објаве. Ако су постови на страни поређани у опадајућем поретку датума објаве, додавање нових постова на прву страну доводи до потискивања постојећих постова на следеће стране дискусије - Слика 3.4г. Ово је слично случају када се додају нове дискусије на индексну страну која је сортирана по датуму креирања дискусија у опадајућем поредку. У овом случају, потребно је претражити само неколико првих страна дискусије да би се дошло до нових постова.

Слика 3.4в приказује дискусију на којој су постови сортирани по растућем редоследу датума објаве. У овом случају нови постови се додају на последњу страну дискусије. Како се постови додају, нове стране дискусије се креирају и додају на крају секвенце страничења. За овај тип дискусије претраживач треба да посети само последње стране да би пронашао нови садржај.

Иако је могуће да форумске технологије динамички мењају начин сортирања и приказ, дисертација се неће освртати на ове случајеве јер немају сви форуми овакву могућност, док поједини чак могу да користе AJAX (*Asynchronous JavaScript and XML*) [10] позиве пре него традиционалне URL линкове да би променили приказ.

4 Постојеће форумске технологије и њихове главне карактеристике

У оквиру овог поглавља биће описане опште карактеристике које су заједничке за различите форумске технологије. Наводе се карактеристике које су битне приликом претраживања форума, као и карактеристике које је пожељно да има свака форумска технологија. На крају поглавља ће бити направљен осврт на најпопуларније софтверске пакете за форуме, као и на њихове предности и мане [11].

4.1 Преглед карактеристика форумских технологија

Табела 4.1 приказује опште информације истакнутих форумских технологија, као и њихове главне карактеристике које су биле познате за време писања овог рада. За информације које нису могле бити проверене, остављена су празна поља.

ТАБЕЛА 4.1
ГЛАВНЕ КАРАКТЕРИСТИКЕ И ОПШТЕ ИНФОРМАЦИЈЕ ИСТАКНУТИХ ФОРУМСКИХ ТЕХНОЛОГИЈА

Технологија	Датум последње верзије	Тренутне верзија	Заступљен програмски језик	Flat	Threaded	Кориснички избор тема	Праћење непрочитаних порука	Преносивост	Email/NNTP интерфејс	Језици	SSN
bbPress	2020-04-29	5.4.1	PHP	Да			Потпуно	Не	Plugin	Да	
Beehive	2016-11-05	1.5.2	PHP	Да	Да	Делимично	Потпуно	Делимично		Да	
Discourse	2020-06-01	2.4.5	Ruby	Да	Не	Да	Сесија	Поптуно	Делимично	Да	OAuth2
Discuz!	2018-01-01	X3.4	PHP	Да		Да	Потпуно			Да	
FluxBB	2018-12-31	1.5.11	PHP	Да	Не	Делимично	Потпуно			Да	
FUDforum	2018-04-09	3.0.9	PHP	Да	Да	Да	Потпуно	Делимично	Да	Да	
Invision	2020-02-04	4.4.10	PHP	Да		Да	Потпуно		Не	Да	Не
MyBB	2019-12-30	1.8.22	PHP	Да	Да	Да	Потпуно	Не		Да	
Phorum	2017-08-23	5.2.23	PHP	Да	Да	Да	Потпуно	Plugin	Не	Не	
phpBB	2020-08-07	3.3.1	PHP	Да	Не	Да	Потпуно	Не		Да	Да
PunBB	2015-10-14	1.4.4	PHP	Да		Да	Сесија		Не	Не	
SMF	2019-12-30	2.0.17	PHP	Да	Не	Да	Потпуно	Делимично		Да	
Thredded	2019-08-17	0.16.13	Ruby	Да	Не	Да	Потпуно	Не	Само Читање	Да	
Vanilla	2019-07-09	3.1	PHP	Да		Да	Потпуно	Plugin		Да	Да
vBulletin	2020-01-08	5.5.6	PHP	Да	Не	Да	Потпуно	Plugin		Не	
XenForo	2020-01-27	2.1.7	PHP	Да	Да	Да	Потпуно	Да		Да	Да

У наставку ће бити наведене прво битне карактеристике за претраживање форума, које могу да утичу на сам приказ информација на странама, па самим тим утичу и на стратегију претраживања.

Постоје два начина приказа садржаја на форуму: *Flat* и *Threaded* – оба могу бити подржана од стране форумске технологије. *Flat* приказ је онај код којег се свака нова порука додаје на крај или почетак дискусије, без постављања релационог односа са неком од

претходних порука (једини однос је то што се поруке налазе у оквиру исте дискусије). Са друге стране, скоро увек постоји функција цитирања поста неког другог корисника како би се омогућило референцирање на друге постове. *Threaded* приказ је онај код којег корисници могу одредити да ли је њихова порука одговор на неку постојећу поруку. Приказ односа на страни између порука се реализује кроз структурно увлачење одговора на дату поруку. Овакав тип форума се најчешће користи за дискусије на којима су појединачне поруке обично кратке, као што су неке друштвене вести (попут *Slashdot* или *reddit*), или коментарисање попут *Disqus*. Док је велика већина форума *Flat, Threaded* форуми са својом структуром приказа могу битно да утичу на стратегију претраживања постова на њиховим дискусијама.

Карактеристика *кориснички избор тема* пружа администратору могућност да изабере изглед и стил форума. Додатно, неки форуми пружају могућност администратору да дефинише више стилова и тема што омогућава кориснику да сам изабере свој изглед. Теме могу бити другачијих боја и графике, или могу подразумевати и другачији приказ, као и формат приказа оптимизован за мале уређаје. Табела поређења приказује да ли дата технологија форума дозвољава администратору да прилагоди изглед без посебног кодирања. Подршка је делимична тамо где је могуће прилагођавање изгледа, али уз додатно кодирање. Различите теме могу битно да утичу на структурни приказ информација на странама, што може да доведе до другачијег циљања садржаја постова на једној истој технологији.

Језици се односе на питање да ли су интернационализација и локализација форумске технологије довољни да истовремено омогуће и стварно пруже граматички исправну подршку матерњем језику циљних корисника. За потребе ове табеле разматрају се језици који се користе приликом инсталације софтверског пакета. Битна компонента рада система предложног у овом раду је детекција датума, који могу бити не само у различитом формату, него и на различитим локалним језицима. Зато се ова карактеристика сматра битном за једну форумску технологију приликом претраживања. С обзиром на то да скоро све технологије подржавају овакав вид локализације, посебан акценат је стављен на исправну детекцију различитих језичких формата датума коришћењем модела машинског учења.

Неке од карактеристика наведених у табели нису меродавне за само претраживање форума, тако да се само укратко описују у даљем тексту. *Праћење непрочитаних порука* се односи на начин на који дата технологија форума прати и приказује поруке које тренутни корисник није прочитао. Праћење порука може бити: (1) потпуно, и односи се на технологије које бележе у базу података информацију о томе које поруке су прочитане или непрочитане за сваког корисника засебно, и (2) преко сесије, које се односи на стартовање сесије неког корисника. *Преносивост* форума пружа корисницима могућност да извезу инсталације форума, а затим да их увезу у оквиру неког другог софтвера или користе неки алат који би их конвертовао. Преносивост може бити кључна особина при одабиру неке форумске технологије. *Email/NNTP интерфејс (Network News Transfer Protocol)* означава да ли дата технологија форума може да се користи са стандардним *Email/NNTP* клијентима. SSN (*Single sign-on*) могућност је често потребна да би се побољшала приступност апликацији и лакши и бржи логин. Ово се постиже коришћењем стандарда попут *OAuth* [12] и *OpenID* [13].

Постоје неке додатне карактеристике форума које нису наведене у оквиру табеле. Једна од њих се односи на системску аутоматску препоруку која је заснована на садржају. Ово може помоћи самим корисницима форума да пронађу постојеће дискусије које су сличне онима које тренутно гледају или су повезане са дискусијом коју претражују. Многи корисници нису заинтересовани за претрагу форума већ директно креирају нове дискусије како би добили одговор на своје питање. На неким форумима, када корисник откуца назив нове дискусије, форум аутоматски предлаже сличне, већ разрађене теме са тог форума. Ово помаже да се број сувишних дискусија смањи док корисници који занемарују претрагу могу пронаћи одговор на своје питање док покушавају да креирају нову дискусију.

Форуми се разликују и по начину одбране од претраживања као и од нежељених порука које неке веб стране или аутоматизовани сервиси покушавају да промовишу у облику садржаја поста. Форумске технологије углавном поседују ефикасан систем одбране од аутоматизованог претраживања и ефикасан скуп алата за уклањање нежељене поште. На пример, систем CAPTCHA [14] се користи код већине форумских технологија и углавном служи за спречавање аутоматских регистрација. Док обично форуми имају јавно доступан садржај, неки од њих су затворени и доступни само за регистроване кориснике. Код оваквих форума само претраживање није могуће ако се не поседује кориснички налог преко којег се може приступити садржају форума. Технике аутоматског логовања и креирања корисничког налога на неком форуму, као и заобилазак система за блокирање претраживача није тема ове дисертације и није даље разматран. Сви форуми који су коришћени за претраживање и тестирање предложеног система у овом раду су имали јавно доступан садржај.

4.2 Преглед најпопуларнијих форумских технологија, њихових предности и мана

Форуми су важна традиционална компонента интернета која корисницима омогућава дискусију и као такви представљају претечу данашњих друштвених мрежа. Данас је лакше него икада поставити форум и омогућити корисницима међусобну интеракцију. Оно што се поставља као питање је која је постојећа форумска технологија најбоља за решавање одређених захтева корисника.

Већина веб форума је прилично једноставна. Омогућава регистрацију корисника и отварање налога, креирање дискусија и постављање порука као одговор на поруке других корисника. У већини случајева веб форуми укључују и систем корисничких улога, тако да на врху могу да постоје администратори и модератори који управљају страницама, док редовни корисници поседују минималне дозволе. Поред ових основних функционалности, постоје неке додатне карактеристике које одређене форумске технологије чине бољим:

- *Прилагођавање профила* – омогућава корисницима да прилагоде своје профиле, што може повећати њихову активност јер имају боље окружење и неку врсту контроле.
- *Напредан уређивач текста* – како се форуми већински заснивају на тексту, корисницима се нуди робусан уређивач за креирање и прилагођавање њихових порука.
- *Приватне поруке* – иако се форуми баве јавном расправом, омогућавање приватних порука је добар начин за подстицање интеракције између корисника.
- *Потписи* – још један вид прилагођавања профила који може повећати активност корисника.
- *Концепт напредовања* – многи форуми постављају корисничке нивое који се могу откључати након одређене активности или времена. Ово представља врсту награде кориснику за често учешће на форуму и мотивише их да буду активнији.

Код избора неке форумске технологије, не морају се захтевати све наведене функционалности. Потреба зависи од врсте искуства које се жели понудити корисницима. На пример, ако форум треба да омогући само постављање неких упита, сувишно је да корисници имају могућност потписа или да форум поседује концепт напредовања. Међутим, већина форума може имати користи од што више модерних функционалности. Заправо, битно је пружити искуство које је најближе платформи друштвених медија или мрежа, тако да корисници осећају потребу за даљим учешћем на форуму. Из тог разлога је разумно при

одабиру технологије форума, тражити најбољи форумски софтвер који нуди што више од претходно наведених функционалности, чак иако се не намерава њихова тренутна употреба.

У наставку овог поглавља биће наведено неколико најзаступљенијих форумских технологија. За сваку од ових технологија биће истакнуте предности и мане.

4.2.1 phpBB

PhpBB [15] је популарна и флексибилна форумска платформа која омогућава креирање више форума и под форума на којима регистровани корисници могу да постављају поруке - Слика 4.1. На креираном форуму може се постављати произвољан број интерних порука. Ова форумска технологија може да подржи велики број активних корисника истовремено.

Поред основних функционалности, *phpBB* технологија омогућава коришћење екстензија за додавање нових опција. Постоји велики број бесплатних екстензија које се могу користити и које омогућавају имплементирање додатних функционалности у оквиру форума. *PhpBB* укључује и избор тема које се користе за промену изгледа форума. Поред тога, корисници форума имају доста опција приликом прилагођавања својих профила и коментара. Додатно, *phpBB* је софтвер отвореног кода (*open-source*) [16] и има велику подршку активне форумске заједнице.

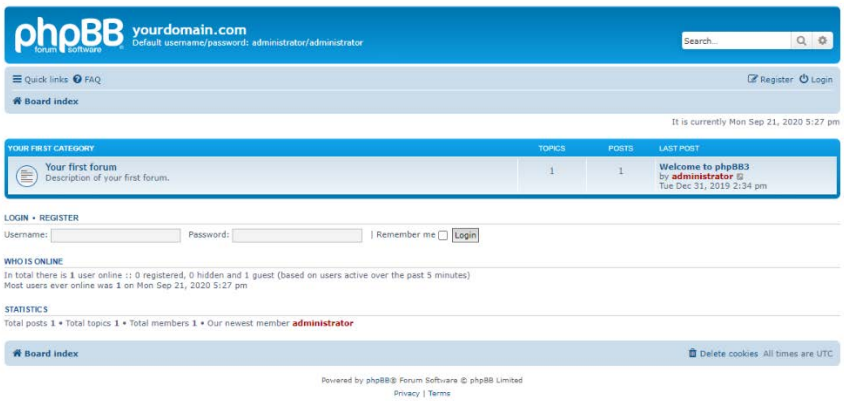
Предности:

- Пружа више опција хијерархије корисничких налога.
- Омогућава корисницима да прилагоде своје постове и профиле.
- Нуди приступ великом броју уређивачких опција.
- Омогућава подешавање екстензија и тема форума.
- Веома је брз.

Мане:

- Већина *phpBB* тема су из ранијих година и не испуњавају модерне визуелне стандарде.

Генерално, *phpBB* представља једну од најприлагодљивијих форумских технологија. Помоћу додатних екстензија добија се приступ свим функционалностима које могу бити потребне. *PhpBB* поседује функционалности које се могу пронаћи међу свим заступљенијим форумским технологијама, као што су креирање форума са порукама, регистрацију корисника и уређивање. Како се ове карактеристике не би понављале, у даљем тексту ће бити фокус на оно што је специфично за сваку од преосталих платформи.



Слика 4.1 – Пример phpBB форума

4.2.2 *MyBB*

MyBB [17] дели пуно функционалности са *phpBB* технологијом - Слика 4.2. Додатно поседује робусне системе за екстензије и теме, има веома активну заједницу и изузетно је једноставан за употребу. Основна разлика у односу на *phpBB* је што *MyBB* не нуди толико велики број екстензија. Са друге стране, *MyBB* је боље стилизован, теме изгледају визуелно модерније и корисницима је једноставнији за коришћење.

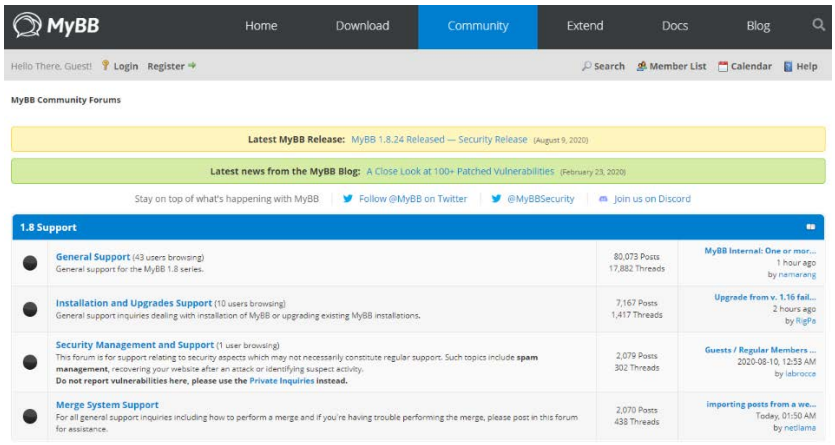
Предности:

- Омогућава једноставну употребу са свим основним функционалностима.
- Пружа напредне могућности подешавања екстензија и тема.
- Укључује системе за праћење репутације корисника.
- Имплементира употребу календара ради планирања догађаја.

Мане:

- Као форумска технологија има мање опција и мање је напредна од *phpBB*.

MyBB може бити избор уколико се жели постићи стилизован изглед. Иако *phpBB* нуди боље функционалности, *MyBB* је и даље довољно робустан за управљање већином популарних функционалности.



Слика 4.2 – Пример myBB форума

4.2.3 *WordPress*

WordPress [18] није оригинално замишљен као форумска технологија и представља врло популаран систем за управљање садржајем CMS (*Content Management System*) [19] - Слика 4.3. Ова платформа омогућава креирање вишенаменских садржаја, од једноставних блогова, до интернет продавница, захваљујући огромној колекцији екстензија. Један од разлога за постављање *WordPress* форума су екстензије типа *Asgaros Forum*, *wpForo Forum* и *bbPress*. Било који од ових додатака пружа све основне функционалности форума, као и мноштво корисничких додатака.

Иако се чини контрапродуктивним коришћење вишенаменске платформе у сврху изградње сопствене интернет заједнице као што је форум, *WordPress* је најбоља платформа за постављање редовне веб странице и форума упоредо.

Предности:

- Поседује приступ великој колекцији тема и екстензија.

- Изузетно једноставан за употребу.
- Омогућава упоредо постављање веб стране и форума.

Мане:

- Платформа није оригинално намењена у сврху креирања веб форума, па захтева више операција и искуства да би се креирао форум.

Уколико се поседује искуство у раду са *WordPress* платформом, има смисла користити је и за постављање форума.

Слика 4.3 – WordPress систем за управљање садржајем

4.2.4 Joomla!

Joomla! [20] је исто као и *WordPress*, CMS систем који омогућава прављење веб стране било које намене, укључујући и форуме - Слика 4.4. Има широки спектар екстензија и тема, тако да је веома прилагодљива платформа. У односу на *WordPress* пружа већу контролу приликом коришћења, али је са друге стране усмерен на кориснике са одређеним искуством у веб развоју, па самим тим није једноставан за употребу. Да би се *Joomla!* користила за потребе форума, неопходно је изабрати одговарајуће екстензије. Неки од најбољих су *Kunena*, *ChronoForum* и *EasyDiscuss*, при чему последња опција није бесплатна.

Предности:

- Нуди високи степен контроле приликом креирања веб страна.
- Омогућава велики избор екстензија намењених креирању форума.
- Пружа добре методе заштите и оптимизовано претраживање – SEO (*Search Engine Optimization*) [21].

Мане:

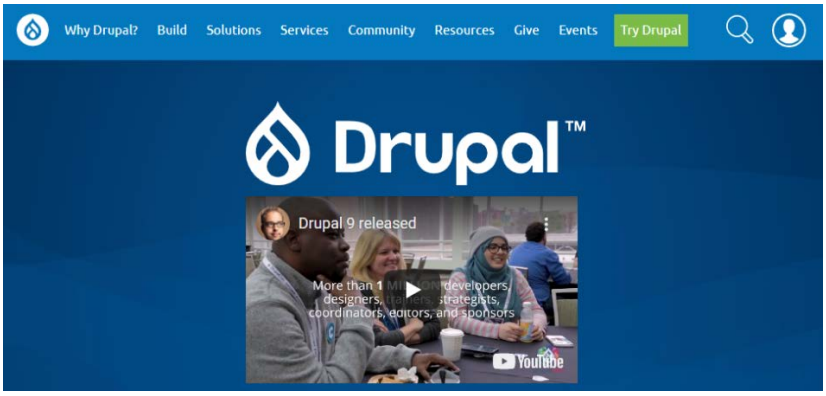
- Иницијално коришћење *Joomla!* платформе без претходног искуства је теже него код других платформи.
- Избор екстензија за форуме је ограниченији у поређењу са *WordPress* платформом.

Док *Joomla!* захтева веће искуство у веб развоју како би се из ње извукао максимум, *WordPress* могу користити и почетници.

Слика 4.4 – Joomla! систем за управљање садржајем

4.2.5 Drupal

Drupal [22] такође спада у групу популарних CMS система, који је корак изнад *Joomla!*-е и *WordPress*-а у погледу основних карактеристика - Слика 4.5. Што се тиче функционалности, *Drupal* је један од најсвестранијих CMS система. Са друге стране, коришћење свих ових функционалности захтева прећашње напредно искуство у раду са овом платформом. *Drupal* нуди и велику колекцију модула и тема у зависности од нивоа искуства, као и од потребе. За разлику од других CMS система, за постављање форума коришћењем *Drupal* платформе није потребно користити неке додатне екстензије јер *Drupal* укључује ову функционалност у своје основне карактеристике. Ипак, постоје модули који омогућавају проширивање основних опција форума, као што је *Advanced Forum*.



Слика 4.5 – Drupal систем за управљање садржајем

Предности:

- Пружа широки спектар могућности прилагођавања.
- Нуди подразумевану могућност креирања форума.
- Омогућава приступ модулима и темама који се могу користити за побољшање форума.

Мане:

- Тешко је започети рад са *Drupal* платформом без претходног напреднијег искуства у веб развоју.

Drupal представља озбиљан избор уколико се тражи висок ниво подразумеваних функционалности и скалабилност. За искусне програмере не представља проблем за коришћење приликом покретања форума.

4.2.6 *Vanilla*

До сада наведене платформе припадају искључиво групи *open-source* софтвера. *Vanilla* [23] (Слика 4.6) поред *open-source* верзије пружа и премијум варијанту која није бесплатна и намењена је купцима на нивоу компанија. Са друге стране, бесплатна *open-source* варијанта ове платформе нуди сасвим довољан спектар коришћења и контроле над креирањем класичних форума.

Платформа се издваја од осталих због високог нивоа могућности уређивања и контроле над функционалностима, садржајем и изгледом. У бесплатној верзији омогућен је приступ темама и екстензијама којих нема много па је коришћење ипак ограничено на подразумеване функционалности. Додатно, са основном верзијом се добија и приступ *Best Of* секцијама, напредном уређивачу текста и аутоматском чувању постова.

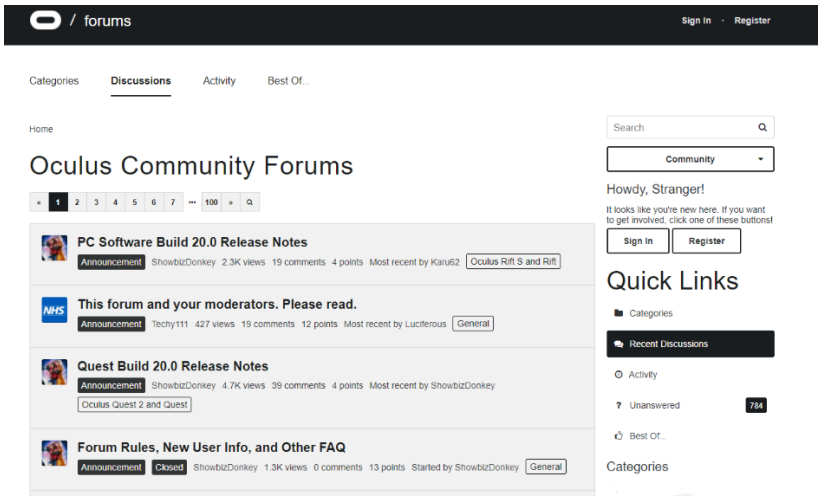
Предности:

- Пружа могућност постављања класичних форума коришћењем основних тема и екстензија.
- Омогућава својим корисницима употребу напредног уређивача текста, са широким спектром опција.
- Аутоматски чува садржај поста док је још у фази писања.
- Поседује опцију креирања приватних корисничких група.

Мане:

- Колекција бесплатних екстензија и тема није велика.

Vanilla поседује бесплатну верзију софтвера која је једна од бољих опција за креирање форума. Одликује се високим могућностима уређивања иако има малу колекцију бесплатних екстензија.



Слика 4.6 – Пример Vanilla форума

4.2.7 *SMF – Simple Machines Forum*

Simple Machines Forum [24] је најсличнији *phpBB* платформи. Оно што издваја ову форумску технологију је флексибилност његових модула и тематских система. *Simple Machines Forum* користи систем који инсталира и ажурира модуле у ходу. Поред тога, нуди највећу колекцију екстензија и тема које се могу пронаћи за неку *open-source* форумску

технологију. Подржава и претплате за чланове форума, као и све остале функционалности које се очекују од једног форумског пакета.

Предности:

- Омогућава приступ највећој колекцији екстензија и тема када је реч о форумској технологији.
- Има једноставну могућност преласка на друге подржане језике.
- Поједностављене опције претплате на неку дискусију премијум корисника.

Мане:

- Већина тема су старијег датума и не задовољавају новије дизајнерске стандарде као ни прилагођавање за мобилне уређаје.
- Не нуди пуно опција за подешавање профила.

Simple Machines Forum је добра опција уколико је један од захтева приступ великом броју екстензија и тема. Ово омогућава имплементацију великог броја функционалности у постављени форум.

Слика 4.7 – Пример SMF форума

5 Преглед сродних и коришћених решења

Методи циљања најновијег садржаја веб форума у инкременталној стратегији претраживања нису довољно покривени у отвореној литератури, али постоје неке претходне студије које су мотивисале ово истраживање, што је резултовало методама и решењима представљеним у оквиру овог рада.

Неки ранији радови [25], [26] су се бавили уопштеним проблемима веб претраживача. Аутори су развили моделе засноване на експоненцијалној расподели како би проценили време између промена на веб странама и увели концепте као што су животни век и старост. Садржај на веб странама је представљен као "амортизујућа роба" *(depreciating commodity)* па га је потребно с времена на време освежити. Иако ово може да важи за веб сајтове где једна страна представља јединствени извор информације који се може ажурирати на различите начине, форуми с друге стране не мењају постојећи садржај и ретко бришу свој раније генерисани садржај. Такође, ови модели не третирају све стране једне дискусије или индекса као целину, и не повезују их на начин како је то представљено у структури форума. С друге стране, за друге веб сајтове свака страна се третира независно.

Опсежан увод и преглед процеса прикупљања података са веб форума представљен је у [27]. У овом раду је дискутовано о општим питањима и проблемима који се односе на прикупљање и обраду података са веб форума, а додатно је представљен пакет за анализу и претраживање података са форума ради откривања друштвених улога учесника. Такође, истраживање представљено у раду [28], заједно са отвореним проблемима, даје опсежан преглед техника откривања догађаја из токова на друштвеним мрежама као што су форуми.

Још једно истраживање усредсређено на инкрементално претраживање форума урадили су *Yang* и други [4]. Ово је врло свеобухватна стратегија инкременталног претраживања форума која користи информације прикупљене на нивоу сајта, представљене у [2], како би креирала репрезентацију организационе структуре веб форума засновану на графу повезаности. Добијена структура се касније користи да би се реконструисале све индексне и дискусионе стране, те да би се могле третирати као један објекат а не као индивидуалне стране. За сваку страну индекса или дискусије врши се екстракција датума креирања садржаја, након чега се тај датум користи у оптимизацији распореда поновног претраживања. У овом раду, такође је фокус на екстракцији датума креирања садржаја, али за потребе детекције начина сортирања на страни. У [4], нађени датум се користи заједно са статистиком на нивоу сајта како би се предвидела фреквенција поновних посета веб сајту и како би се предвидео животни век сваког објекта. Овај рад даје робустан приступ инкременталној стратегији претраживања форума, али се углавном фокусира на то како уравнотежити баланс између покривености и свежине садржаја, као и да предвиди следеће време поновног циклуса претраживања. Стратегија циљања новонасталог садржаја између два претраживања није анализирана, већ се имплицитно претпоставља да се најновији садржај може лако пронаћи поновним посетама странама веб форума, без узимања у разматрање технологије форума и презентације садржаја на истом, која може да варира. Додатно, у оквиру ове дисертације је показано да је за прецизну детекцију датума ипак потребно изградити систем који се базира на техникама машинског учења, а не само на регуларним изразима како су аутори навели.

У претходне радове се такође могу уврстити најсавременији специјализовани претраживачи форума као што су iRobot [2] и FoCUS [3]. И FoCUS и iRobot су дизајнирани за преузимање само важних страна форума, док се неважне стране прескачу, што их чини

ефикасним у смислу пропусног опсега и времена претраживања. iRobot је у стању да мапира URL локације на страни анализом претходно узоркованих страница. Ове странице се прво групишу одабиром информативних кластера на основу претходно утврђених мера. Коришћењем алгоритма обухватног стабла, сличног Примовом [29], проналазе се путање форума помоћу информативних кластера. Док iRobot учи да пронађе локације скелетних линкова форума на страни, FoCUS учи регуларне изразе како би препознао и детектовао скелетне линкове. На основу изгледа и специфичних карактеристика страна веб форума, FoCUS креира класификаторе које користи како би класификовао странице и поделио URL-ове који их гађају у скупове страна индекса, дискусија и пагинационих линкова. Ови скупови се касније користе за рекурзивно учење регуларних израза.

У раније спроведеном истраживању [30], представљен је интелигентни форумски претраживач који је био базиран на побољшаним регуларним изразима који су могли да препознају скелетне линкове форума. Овај претраживач користи побољшана својства регуларних израза да би препознао URL-ове форума што прецизније. У оквиру претходних истраживања [31], стране форума са дискусијама су третиране као један објекат да би се лакше расподелиле за паралелно претраживање, чиме је постигнута задовољавајућа балансираност оптерећења ресурса приликом претраживања.

Сви поменути претраживачи су дизајнирани да посећују само важне стране форума, занемарујући оне неважне. Ова стратегија је погодна код прве посете форуму и иницијалне претраге, где треба сакупити све битне стране и избећи неважне. Примена ових претраживача у инкременталној стратегији би довела до поновног преузимања свог битног садржаја који је већ био сакупљен у неким од претходних циклуса претраживања. Даље, оваква решења нису у могућности да директно циљају само најновији садржај генерисан од последњег циклуса претраживања форума. У истраживању спроведеном у оквиру ове дисертације користи се робустан приступ машинског учења за детекцију и нормализацију датума креираног садржаја на форуму, тако да се ефикасно и са великом прецизношћу може одредити начин сортирања страна. Утврђени начин сортирања омогућава коришћење одговарајућих URL-ова као и навигационих путања технологије форума, зарад циљања страна са најновијим садржајем.

У сврху изградње система, који је главни резултат ове дисертације, су коришћени неки елементи FoCUS претраживача, пошто су експериментални резултати представљени у [3] показали да се FoCUS понаша боље од iRobot-а приликом претраживања форума. Додатно је побољшано препознавање специфичних URL-ова као што је URL који се налази на индексној страни и који референцира страну последње активности на дискусији, а такође је побољшано и проналажење одређеног URL-а у групи линкова који се користе за пагинацију.

Један од новијих претраживача представљених у отвореној литератури је Vigi4Med [32]. Vigi4Med представља високо прецизни структурирани сакупљач података форума прилагодљив за претраживање великих размера. Међутим, захтева програмерско знање и креирање засебне конфигурације за сваки форум. Vigi4Med омогућава да се прикупљени структурирани подаци у RDF (*Resource Description Framework*) формату [33] могу по потреби анонимизовати. Да би овај претраживач могао да циља регионе форума који садрже кориснички генерисане податке и да би могао да се креће по форуму, корисник мора написати и приложити XPath упите [34]. Да би се смањио напор у људској интервенцији за писање XPath упита, у [35] је предложено решење које аутоматизује овај процес. На основу унапред дефинисаних семантичких правила у виду регуларних израза и URL-а почетне стране форума, предложени модел генерише колекцију XPath упита која се касније користи у претраживању и сакупљању података са форума. Иако ово омогућава лакше коришћење претраживача, он и даље у великој мери зависи од робусности семантичких правила регуларних израза. Пре почетка претраживања форума, потребна је детаљна мануелна инспекција како би се семантичка правила ажурирала и прилагодила за тренутни форум ако за тим има потребе.

С друге стране, постоје специјализовани претраживачи попут [36] (CrimeBot) као и решење представљено у [37], који се користе за претраживање илегалних форума (*underground forums*) и прикупљање података у вези са илегалним активностима. Оба претраживача су високо софистицирана и дизајнирана да, аутоматски или уз помоћ корисника, савладају технике против прикупљања које су имплементиране на илегалним форумима, а које претраживачи опште намене не узимају у обзир, као што су [38]: обфускација, скривеност, намерно загушивање, идентификација претраживача, функционалност пријаве (логин) или Тјурингов тест попут CAPTCHA [14] валидације. Иако CrimeBot имплементира инкременталну стратегију претраживања форума, његово техничко решење није јавно објављено у време писања овог рада нити је довољно јасно документовано да би се омогућила његова имплементација или поређење. Поред тога, решење инкременталног претраживача форума предложено у [37] је превише прилагођено форумској технологији *vBulletin*, тако да није широко применљиво јер различите форумске технологије имају различите врсте презентације садржаја и различите имплементационе детаље.

Решења представљена у [39]–[41] за откривање страна које су скоро-дупликати могу бити корисна када се покушава избегавање страна које су већ биле посећене. Откривање дупликата се базира на садржају и постиже се карактеризацијом веб страна са обрасцима отисака попут *SimHash* [41] или *Shingles* [42], а затим упоређивањем са малом Хаминговом дистанцом [43], како би се одредио дупликат. Проблем са детекцијом дупликата је тај што се може применити тек након што се стране преузму са веб сајта, што није ефикасно у смислу оптерећења пропусног опсега и утрошка времена. Додатно, откривање дуплог садржаја у инкременталном претраживању форума захтевало би преузимање свих страна форума и упоређивање њиховог садржаја са оним који се већ налази у бази.

Постоје индустријски стандарди као што је *sitemap* [44], [45], који представља протокол који веб сајтови могу имплементирати како би пружили додатну информацију претраживачу о тренутном статусу страна. *Sitemap* је XML датотека која садржи списак URL локација страна конкретног веб сајта, њихово последње време ажурирања, учесталост промене и приоритет. Ови подаци могу бити коришћени од стране претраживача приликом скенирања форума или било ког другог веб сајта у потрази за новим садржајем. На тај начин, већ прикупљене стране могу бити ажуриране ако је датум последње измене новији од последњег датума индексирања. URL-ови страна који нису пронађени у претходним циклусима претраживања могу се сматрати новим. Иако *sitemap* протокол може помоћи при инкременталном претраживању, експерименти представљени у [3] показали су да администратори веб сајта ретко правилно одржавају ову датотеку, што може довести до претраживања страна које нису релевантне за садржај форума или дупликата страна.

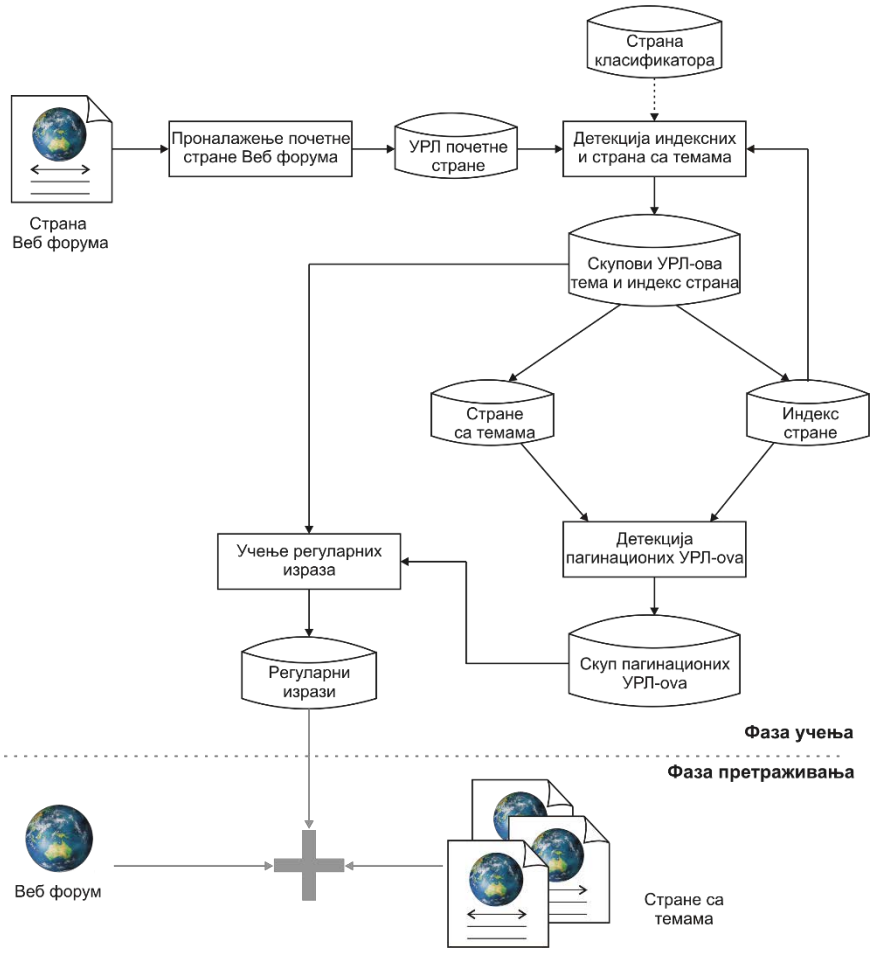
5.1 FoCUS – специјализовани претраживач веб форума

FoCUS је претраживач специјализован за веб форуме, чије се функционисање може разложити у две фазе: прва фаза се састоји од учења регуларних израза, док друга фаза преставља фазу претраживања користећи научене регуларне изразе. Архитектура FoCUS претраживача се може видети на Слици 5.1. Почевши од било које задате стране веб форума, FoCUS прво проналази његову почетну страну. Потом, користећи SVM (*Support Vector Machine*) [45] као класификатор и предефинисане карактеристике страна, почевши од почетне, врши детекцију индексних и дискусионих страна, па URL-ове који на њих указују групише у одвојене скупове који ће касније бити коришћени као тренинг сет. Овако формирани URL скупови се користе за тренинг како би се научили регуларни изрази на начин који су предложили Koppula и други [46]. Тако научени регуларни изрази се касније користе како би

се уз помоћ њих препознали еквивалентни URL-ови приликом претраживања. Генеришу се четири врсте регуларних израза: за проналажење индекс страна, индексних пагинационих URL-ова, дискусионих страна, и пагинационих URL-ова на дискусијама. Ове четири врсте регуларних израза се користе од стране претраживача у другој фази како би се детектовали и прошли скелетни линкови форума и на тај начин избегле његове неважне стране. С обзиром да се у овом раду највише користе елементи FoCUS архитектуре, у даљем тексту ће бити укратко описане његове методе и начин функционисања, док се више детаља може пронаћи у [3].

5.1.1 Прављење индекс и дискусионих URL тренинг сетова

Процедуре за креирање индекс и дискусионих URL тренинг сетова су сличне јер ова два типа URL-а имају сличне карактеристике и разликују се само у одредишној страни коју циљају. Ово значи да се и индекс и дискусиони URL могу пронаћи на индексној страни, да могу имати сличне дужине URL текста и да је једини начин да се разликују заправо инспекција стране на коју они циљају. У даљем тексту је описан метод за препознавање типа стране на коју показују ова два типа URL-а.



Слика 5.1 – Архитектура FoCUS претраживача [3]

Свака од страна ова два типа има неке своје јединствене карактеристике и форме које представљају њен изглед. Пример индексне стране се може видети на Слици 3.2, док се пример дискусије може видети на Слици 3.3. Индексна страна се, у зависности од технологије,

разликује од дискусионе стране по томе што има релативно уске редове у којима се налазе URL-ови дискусија и њима придружене информације као што је наслов, датум креирања, URL ка страни последње активности и сл. С друге стране, дискусионе стране имају доста шире редове у односу на индексне стране и на њима се налази много више текста који представља кориснички генерисан садржај. Иако и један и други тип стране садржи датуме, они су обично поређани у различитим редоследима и на различитим локацијама. Додатно, на дискусионим странама постоје информације о аутору, укључујући URL ка његовом профилу, датум регистрације, датум последње активности, пол, локацију и сл. На основу ових различитости предложене су карактеристике које су послужиле као основа за прављење класификатора страна користећи SVM са подразумеваним линеарним језгром. Све особине које су засноване на изгледу страна, одлазним URL-овима, мета подацима и особинама циљаних записа на страни се могу видети у Табели 5.1., као што је дефинисано у [3].

ТАБЕЛА 5.1
ИСТАКНУТЕ ОСОБИНЕ КОРИШЋЕНЕ ОД СТРАНЕ SVM КЛАСИФИКАТОРА ЗА ПРЕПОЗНАВАЊЕ СТРАНА

Карактеристика	**Опис**
Max/Avg/Var дужине	Максимум / просек / варијанса дужине блока међу свим блоковима
Max/Avg/Var ширине	Максимум / просек / варијанса ширине блока међу свим блоковима
Max/Avg/Var	Максимум / просек / варијанса дужене текста URL-а у карактерима међу свим записима
Max/Avg/Var	Максимум / просек / варијанса дужине обичног текста међу свим записима
Max/Avg/Var	Максимум / просек / варијанса броја листова HTML DOM стабла међу свим блоковима
Max/Avg/Var	Максимум / просек / варијанса URL-ова међу свим блоковима
Има URL	Да ли сваки запис / блок садржи URL
Има кориснички URL	Да ли сваки запис / блок садржи URL који показује на кориснички профил
Има временски запис	Да ли сваки запис / блок садржи датум
Временско сортирање	Редослед свих датума је сортиран
Сличност стабла блока	Сличност текста између свих блокова
Сличност стабла блока	Сличност HTML DOM стабла међу свим блоковима
Однос обичног и URL текста	Однос URL текста и обичног текста измерен у карактерима
Број група	Број група елемената после поравнања HTML DOM стабла

После детекције типа стране, односно одређивања да ли је страна индексна или дискусиона, следећи корак је проналажење URL-ова на индексној страни. По изгледу, индексна страна највише личи на структуру налик табели у чијим колонама се налазе информације. Сваки ред ове структуре садржи URL ка индексној или дискусионој страни заједно са осталим информацијама као што су датум креирања, датум последње измене, аутор који је креирао дискусију, име аутора који је последњи имао активност и сл.

Исти тип информације је смештен у истој колони у оваквој структури, па се ова особина може искористити да би се детектовале локације, односно колоне које садрже URL-ове. Обично URL-ови дискусија или индекса имају дужи текстуални опис од осталих линкова у оваквој структури. Колоне са URL-овима се сврставају у групе, и група која има највећу укупну дужину текста линка се узима за колону која садржи URL-ове који се траже.

Према [47] и [48], URL-ови који се појављују у HTML (*HyperText Markup Language*) структури налик табели, могу бити преузети тако што се DOM (*Document Object Model*) стабла поравнају, а потом се подаци смештају у линк-табелу. FoCUS користи парцијално поравнање HTML DOM стабла представљено у [47]. Пример парцијалног поравнања, где се структура налик табели претвара у линк-табелу, се може видети на Слици 5.2. На приказаној илустрацији се може видети да се, после парцијалног поравнања HTML DOM стабла URL-ови дискусије,

датуми последње активности, имена аутора и остале информације, групишу и пресликавају у колоне. Према већ наведеној претпоставци, Група 2 са Слике 5.2, се узима за кандидат групу где се налазе URL-ови јер садржи најдужи укупан текст свих URL-ова.

У овом тренутку се још увек не може одредити тип URL-а јер се мора проверити тип стране на који ови URL-ови показују. Алгоритам "гласање већине" (*majority voting*) се користи за утврђивање типа стране на који пронађени URL-ови показују, тј. узима се онај тип стране којих има више од пола. Разлог је у чињеници да класификациони резултати одређивања појединачних типова страна могу бити у неком проценту погрешни.

Група 1	Група 2	Група 3	Група 4	Група 5	Група 6	Група 7	Група 8	Група 9	Група 10
D	what can apple see of your apple id	ddddd478	Apr 27, 2020	Replies: Views:	12	545	Apr 27, 2020	now i see it	N
	Main screen layout	_karrol	Apr 27, 2020	Replies: Views:	4	310	Apr 27, 2020	_karrol	
	How do mileage apps work ?	venomx999	Apr 27, 2020	Replies: Views:	2	206	Apr 27, 2020	now i see it	N
S	iPhone SE(2020) vs iPhone XS	solemntrance	Apr 26, 2020	Replies: Views:	26	4k	Apr 27, 2020	Locke Cole	
S	If I need a new iPhone now, what is the best option?	Synna	Apr 17, 2020	Replies: Views:	40	3k	Apr 27, 2020	ScreenSavers	

Слика 5.2 – Пример поравнања HTML DOM стабла

5.1.2 Креирање тренинг сетова са пагинационим линковима

Пагинациони линкови имају својство повезивања индексних или дискусионих страна у једну логичку целину, јер садржај једне дискусије или индекса може бити дистрибуиран на више страна ако га има много. Пагинациони линкови могу бити груписани и нумерисани у једном блоку стране, или пагинационе линкове може чинити само један URL који има текст у свом опису као што је то случај са блоговима. Следећа запажања се користе приликом детекције пагинационих URL-ова:

1. Текст URL-а може бити секвенца бројева као што је 1,2,3… или специјални текст као што је "следеће".
2. Ови URL-ови су груписани и појављују се на истом месту HTML DOM стабла изворне стране на којој се налазе и одредишне стране на које показују.
3. Одредишна страна има сличан изглед као и изворна страна. За поређење сличности ове две стране се извршава компарација њихових HTML DOM стабала.

4. Ако пагинационе линкове чини само један URL, тада је његово својство да на свим странама има исти текст, али различит URL формат и да се налази на истом месту/блоку изворне и одредишне стране.

Алгоритам који имплементира ова правила прво детектује групне пагинационе линкове, па ако не успе у томе покушава са проналажењем пагинационих линкова које чини само један URL. Нађени URL-ови се коришћењем класификатора страна потом групишу у тренинг сетове за пагинационе URL-ове индекса и дискусија.

5.1.3 Учење регуларних израза

FoCUS претраживач користи метод генерисања регуларних израза представљен у [46]. Овај алгоритам полази од генеричког узорка „*,“ и проналази обрасце који одговарају URL-овима из скупова за тренинг. Обрасци се рекурзивно пречишћавају док на крају не остану само обрасци који се користе касније као регуларни изрази. Овај алгоритам није превише строго дефинисан и може узимати у обзир негативне примере уз додатни праг толеранције за одбацивање URL адреса са слабом покривеношћу. На пример, за дате линкове

- http://www.gardenstew.com/about20152.html
- http://www.gardenstew.com/about18382.html
- http://www.gardenstew.com/about19741.html
- http://www.gardenstew.com/about20142.html
- http://www.gardenstew.com/user-34.html
- http://www.gardenstew.com/post-180803.html

од којих су прва четири позитивна, а последња два негативна, генеришу се следећи регуларни изрази

1. http://www.gardenstew.com/about\d+.html.
2. http://www.gardenstew.com/user-\d+.html.
3. http://www.gardenstew.com/post-\d+.html.

Модификација коју су увели аутори FoCUS-а дефинише праг толеранције којим се одређује да ли узимати генерисани регуларни израз као валидан. Разлог је што у тренинг скупу могу да се појаве негативни примери, па је идеја да се регуларни изрази са малом покривеношћу одбаце. Праг толеранције, који је уведен у оквиру рада, наводи да покривеност мора бити већа од 20% укупног броја свих URL-ова који се налазе у тренинг сету.

5.1.4 Проналазак почетног URL-а форума

На почетку сваког претраживања, потребно је одредити почетни URL јер од њега креће да се претражује веб форум. Ова адреса је битна из разлога што преставља корен стабла сачињеног од страна и скелетних линкова форума. Зарад аутоматизације претраживања, а и да би се умањила интервенција корисника, уведено је аутоматско откривање почетног URL-а. Разлог је у чињеници да почетна адреса веб форума не мора да буде иста као и почетна адреса веб сајта где се налази тај веб форум. Почетни URL може да варира од веб сајта до веб сајта и не налази се увек на истом месту.

Уводе се основна хеуристичка правила преко којих се покушава проналазак почетне адресе веб форума. Прво се покушава са проналаском кључних речи као што су ‘*forum*’, ‘*board*’, ‘*community*’, ‘*bbs*’, и ‘*discus*’, у URL адреси која се завршава са ‘/’. Ако се пронађе URL који задовољава ова основна правила, узима се као почетни URL, ако се не пронађе, онда се

почетна адреса веб сајта узима као почетни URL. Додатно, да би систем био скалабилан, уводи се и метод откривања који користи следећа опажања:

1. Скоро свака страна форума садржи URL адресу која води кориснике назад на почетну страну форума.
2. Почетна страна веб сајта који садржи форум мора да садржи URL адресу почетне стране тог форума.
3. Ако је детектована URL адреса заправо индексна URL адреса, она не би требало да буде и почетна URL адреса.
4. Почетна страна веб форума углавном садржи највише индекс URL-ова, јер треба да омогући корисницима да што лакше дођу до свих дискусија.

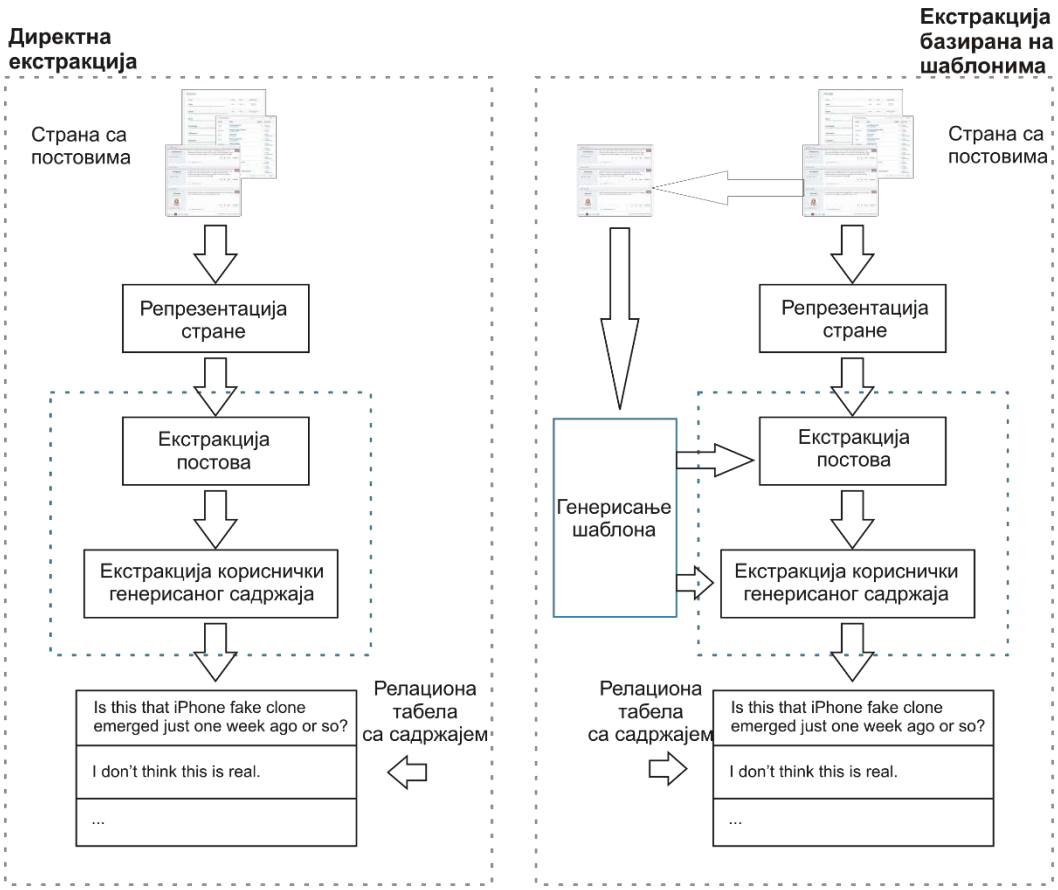
5.2 WeRE - екстракција података са веб форума

WeRE [49] је систем за аутоматску екстракцију кориснички генерисаног садржаја са веб форума коришћењем софистицираних метода. WeRE долази у две варијанте, са директном екстракцијом и екстракцијом базираној на шаблонима - Слика 5.3 Док је директна екстракција погодна за мешовито издвајање садржаја из постова са различитих сајтова, ипак захтева да стране са којих се издваја садржај имају барем неколико постова на страни. С обзиром да ово није случај у пракси, у овом раду је коришћено решење по узору на другу варијанту.

Друга варијанта екстракције захтева једну или више страна на основу којих ће прво бити генерисан шаблон (омотач) који ће се касније користити приликом екстракције. Док је ова варијанта са једне стране зависна од шаблона који се генерише, с друге стране је доста бржа у пракси јер су неки алгоритми који се користе у директној екстракцији за сваку страну понаособ изузетно захтевни.

Као што је приказано на Слици 5.3, WeRE се састоји из следећих фаза. За страну коју систем обрађује, прво се ради екстракција корисних визуелних карактеристика и дефинише се репрезентација стране. Потом се на основу ових карактеристика детектује минимално HTML DOM под-стабло које представља регион унутар којег се садрже сви постови. Након детекције региона постова, врши се екстракција постова, тако што се проналази серија под стабала региона који заједно формирају пост. На крају се проналази минимално под стабло које у себи садржи кориснички генерисан садржај. Резултат је релациона табела, у којој сваки ред одговара једном кориснички генерисаном садржају поста.

Разлика у процесу између директне екстракције и екстракције базиране на шаблонима је у томе што алгоритми, који се користе приликом обраде сваке стране у директној екстракцији, у другој варијанти бивају коришћени само једном и то приликом генерисања шаблона. Генеришу се два типа шаблона, један за регион постова, и један за сам садржај постова. Сваки шаблон заправо преставља сачуване информације које треба да се детектују, а то су путања HTML тага, позиција, дужина и фонт. Када се систему пошаље веб страна на екстракцију, узимају се у обзир само они тагови који се поклапају са путањама тагова из шаблона. Како ово некада није довољно, јер постоји могућност да се појави одређени шум на страни у виду региона или пагинационих линкова, користи се филтрирање на основу позиције и фонта. У даљем тексту су укратко објашњени алгоритми и методи екстракције.



Слика 5.3 – Илустрација WeRE архитектуре [49] *са директном екстракцијом и екстракцијом базираној на шаблонима*

5.2.1 Репрезентација веб стране и детекција корисних визуелних карактеристика

Веб страна се прво парсира у HTML DOM стабло, одакле се даље ради екстракција корисних визуелних информација која се показала као изузетно делотворна [50]–[52]. Визуелне информације које користи WeRE систем су:

- Позиција – координате горњег левог угла елемента;
- Величина – дужина и ширина квадранта елемента који заузима простор на страни;
- Фонт – информације о фонту елемента, као што су величина, стил и боја.

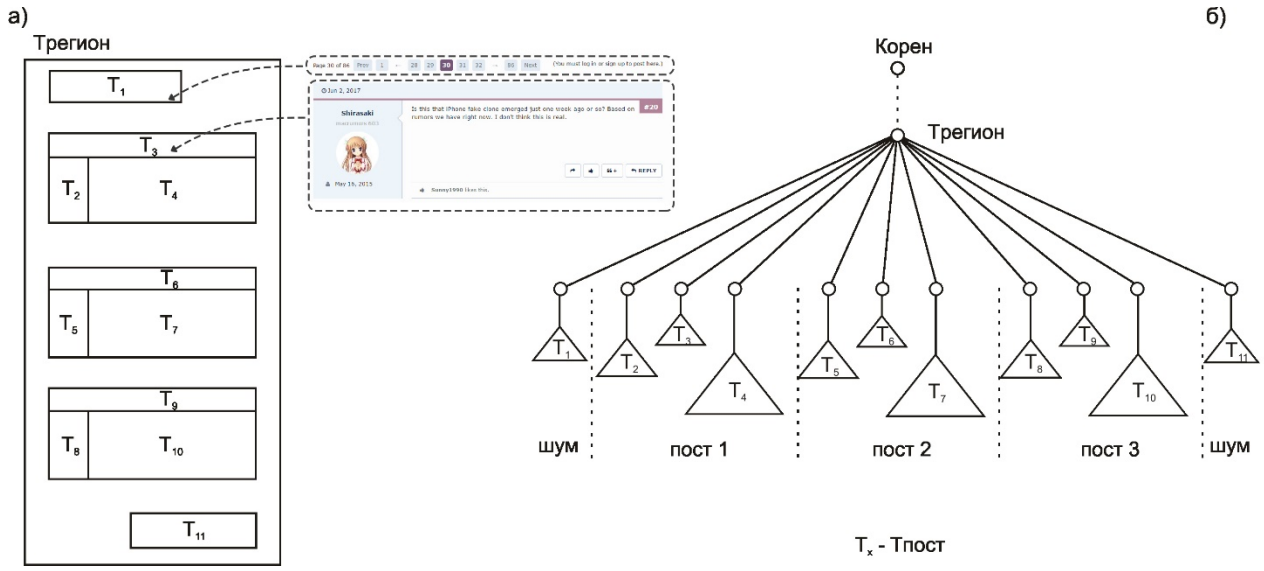
Стране дискусија су увек сачињене од репетитивних региона у којима се садрже постови са кориснички генерисаним садржајем (Слика 5.4). $T_{регион}$ представља минимално под стабло које садржи корисничке постове. На Слици 5.4 је приказан $T_{регион}$ на веб страни и $T_{регион}$ као HTML DOM стабло. Под корисним визуелним карактеристикама, аутори овог система подразумевају:

- Да су сви постови на страни поређани вертикално, имају исту ширину и са леве стране им се налазе додатне информације као што су ауторово име, профилна слика, URL адреса корисниковог налога, пол, локација, датуми регистрације и последње активности и сл.
- Шаблон једног поста на страни је идентичан за сваки пост

- Елементи који престављају исте објекте у оквиру постова су слични по изгледу, позицији и фонту, али се разликују у кориснички генерисаном садржају.

Као карактеристике HTML DOM стабла се издвајају следећа опажања:

- Сваки пост се састоји од једног или више под стабала $T_{регион}$-а који заједно формирају један пост. Ова под стабла се у даљем тексту називају $T_{пост}$.
- Сваки пост на једној стани садржи исти број $T_{пост}$-ова.
- Семантички редослед $T_{пост}$-ова је увек исти код сваког поста.
- Поред $T_{пост}$-ова постоје и делови $T_{регион}$-а који чине шум и који се морају одстранити. Обично се овај шум налази на почетку или крају $T_{регион}$-а и представља, на пример, пагинационе линкове.



Слика 5.4 – Генерализација случаја постова са шумом на веб страни [49] *на основу Слике 3.3: а) регион постова на веб страни б) HTML DOM под стабло региона постова*

5.2.2 Екстракција постова

Екстракција постова се састоји од три фазе: детекције региона који садржи постове, отклањање шума и, на крају, детекције граница самих постова. Детекција региона постова је метод сличан методу представљеном у [53] одакле се уводе следећа опажања која се користе приликом екстракције региона: (1) регион окупира велику површину на страни тј. више од половине стране је окупирано овим регионом (2) лоциран је централно (3) садржи много карактера (4) садржи барем три репетитивна региона који заједно представљају постове. Као додатак, аутори WeRE-а уводе и чињеницу да ови репетитивни региони садрже и датуме који су сортирани.

Да би се утврдиле границе постова, и отклонио шум, прво је потребно израчунати сличност два стабла. Аутори WeRE-а су предложили свој алгоритам поређења два стабла, који се разликује од претходних решења као што су они представљени у [54], [55]. Претходна решења имају метрику где се узима у обзир само сличност HTML тагова, док су у решењу представљеном у овом раду аутори урачунали и визуелне информације као што је фонт, јер се

мора узети у обзир и семантичка репрезентација тага. Два стабла са истом семантиком би требала бити сличнија у структури него стабла са другачијом семантиком.

Основна идеја која стоји иза овог алгоритма јесте да што год се чворови на већој дубини стабла више поклапају, то су и та два стабла сличнија. Алгоритам се састоји од два рекурзивна под-алгоритма базирана на динамичком програмирању која израчунавају: (1) максималну сличност стабала и (2) максимално поклапање секвенце чворова. Сличност два стабла се мери као сума њихових нивоа $N(a,b)$ и сличности секвенце под чворова. Ниво $N(a,b)$ два стабла a и b се рачуна на следећи начин:

$$N(a,b) = \exp\left(\frac{nivo(a,b) - pDubina(T_{region})}{pDubina(T_{region})}\right) \quad (5.1)$$

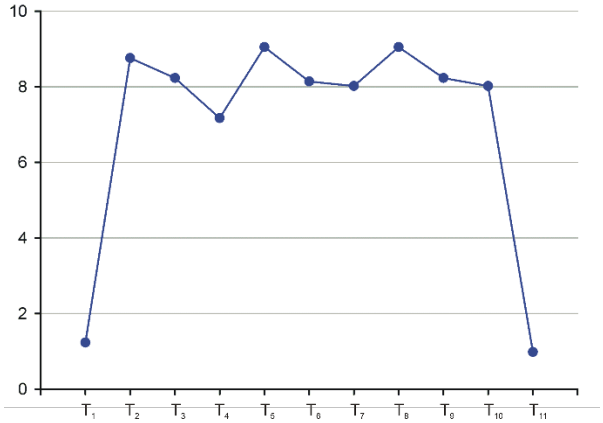
где $nivo(a,b)$ преставља дужину пута од почетка $T_{регион}$-а до a или b, док *pDubina* преставља просечну дубину под стабала $T_{регион}$-а. Што је $nivo(a,b)$ већи, а *pDubina* мања, сличност два стабла је већа. Резултат функције максималне сличности стабла је увек позитиван, и постаје већи ако стабла која се мере имају велики број чворова. $nivo(a,b)$ се не користи директно да би се израчунао ниво између и a и b јер се са *pDubina* урачунавају различитости између шаблона веб страна. Са друге стране, максимално поклапање секвенце чворова два стабла мери сличност секвенце два низа чворова потомака та два стабла, проналазећи максимално поклапање кроз динамичко програмирање. Ако су два стабла иста, резултат овог алгоритма ће генерисати велики број, док у случају не поклапања овај број је јако мали или можда чак 0, у случају да не постоји никаква сличност.

Користећи резултат ове функције, могуће је пронаћи под стабла региона која не садрже постове, него шум који треба одстранити. Идеја се заснива на томе да се сличности свака два под стабла $T_{регион}$-а измере, тако да ако постоји n под стабала, свако под стабло добија $n-1$ мерења са преосталих $n-1$ под стабала. Од сваког под стабла и његових $n-1$ мерења се узима максимум, који се назива и глобални максимум сличности GS_i под стабла i. На тај начин, шум који се налази на крајевима региона ће имати јако мали глобални максимум у поређењу са постовима, чија стабла су већа и имају много већи глобални максимум сличности. У Табели 5.2 су приказане вредности сличности између свака два под стабла као и њихови глобални максимуми. Може се уочити да су вредности симетричне у односу на главну дијагоналу табеле, што произилази из чињенице да се сличност између свака два под стабла мери два пута (T_i и T_j односно T_j и T_i).

ТАБЕЛА 5.2
СЛИЧНОСТИ БИЛО КОЈА ДВА ПОД СТАБЛА РЕГИОНА СА СЛИКЕ 5.4

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	GS_i
T1		**1,23**	0,99	0,71	1,11	0,89	0,69	1,17	0,93	0,76	0.79	**1,23**
T2	1.23		5,54	2,13	**8,73**	6,01	1,70	8,29	1,94	4,14	0,95	**8,73**
T3	0,99	5,54		3,13	5,53	8,22	2,66	4,99	**8,27**	2.95	0,72	**8,27**
T4	0,71	2,13	3,13		4,28	3,59	7,22	3,90	3,13	**7,27**	0,80	**7,27**
T5	1,11	8,73	5,53	4,28		7,01	3,67	**9,04**	6,92	2,77	0,96	**9,04**
T6	0,89	6,01	8,22	3,59	7,01		2,17	5,19	**8,11**	2,41	0,55	**8,11**
T7	0,69	1,70	2,66	7,22	3,67	2,17		3,71	1,98	**8,02**	0,81	**8,02**
T8	1,17	8,29	4,99	3,90	**9,04**	5,19	3,71		5,75	7,89	0.65	**9,04**
T9	0,93	1,94	**8,27**	3,13	6,92	8,11	1,98	5,75		3,32	0,91	**8,27**
T10	0,76	4,14	2.95	7,27	2,77	2,41	**8,02**	7,89	3,32		0,67	**8,02**
T11	0,79	0,95	0,72	0,80	**0,96**	0,55	0,81	0.65	0,91	0,67		**0,96**

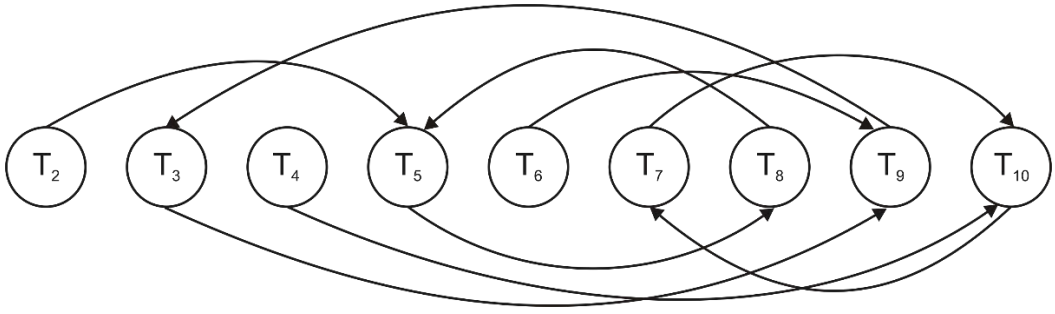
Међутим, чисто посматрање резултата поређења сличности и глобалних максимума није довољно, нити се може са сигурношћу одредити нека граница која би означила да је неко под стабло шум или не. Разлог за то је што сличности под стабала два шума могу бити у границама између *(a,b)*, док су сличности под стабала постова између *(c,d)*, где се може десити да је $a < c < b < d$. Додатно се мора посматрати и стопа промене глобалних максимума сличности. Ако се посматра Слика 5.5, те ако се узме да се стопа промене рачуна као GS_i/GS_{i-1}, тада највећа позитивна промена представља први пост коме претходи шум, док највећа негативна промена представља последњи пост после којег долази опет шум. Ово све иде уз претпоставку да шум постоји. За случајеве када шум не постоји, да се случајно не би одстранила валидна под стабла постова, аутори уводе праг толеранције одређен експерименталним путем од 2.5 испод којег, ако не падне сличност, под стабло се све једно не одбацује.



Слика 5.5 – Пример промене тренда сличности на основу глобалног максимума под стабала са Слике 5.4

После уклањања шума, детектују се границе постова. Да би се исправно детектовале границе постова, прво је потребно утврдити колико узастопних под стабала $T_{регион}$-а формира један пост. На пример, са Слике 5.4б се може утврдити да пост формирају 3 под стабла.

Да би се утврдио тачан број под стабала која представљају један пост, прво се формира усмерени граф који се састоји од чворова T_i и грана (T_i,T_j), где грана са T_i на T_j постоји само ако је глобални максимум сличности од T_i такође и сличност између T_i и T_j. Овако формиран граф аутори називају GSRG (*Global Similarity Relationship Graph*). У GSRG графу, број излазних грана сваког чвора је 1, док је број улазних грана у распону $[0, n-1]$, и пример таквог графа на основу Слике 5.4 дат је на Слици 5.6.



Слика 5.6 – Илустрација усмереног GSRG графа под стабала на основу Слике 5.4

Дистанца за сваки чвор T_i се рачуна као $d_i = |i - j|$, где је T_j најсличније под стабло T_i. Даље је потребно пронаћи најмањи заједнички делилац свих дистанци у графу, који представља број под стабала који формирају један пост. Са Слике 5.6, дистанце за $\{T_2,...,T_{10}\}$ су d = {3,6,6,3,3,3,3,6,3} где је њихов најмањи заједнички делилац 3, одакле произилази да сваки пост чине три узастопна под стабла.

5.2.3 Екстракција кориснички генерисаног садржаја из поста

Када се одреди број под стабала који формира један пост, потребно је одредити које под стабло садржи кориснички генерисан садржај и потом извршити његову екстракцију. Ово није тривијалан задатак, јер региони који садрже кориснички садржај могу да се разликују што у структури што у дужини текста.

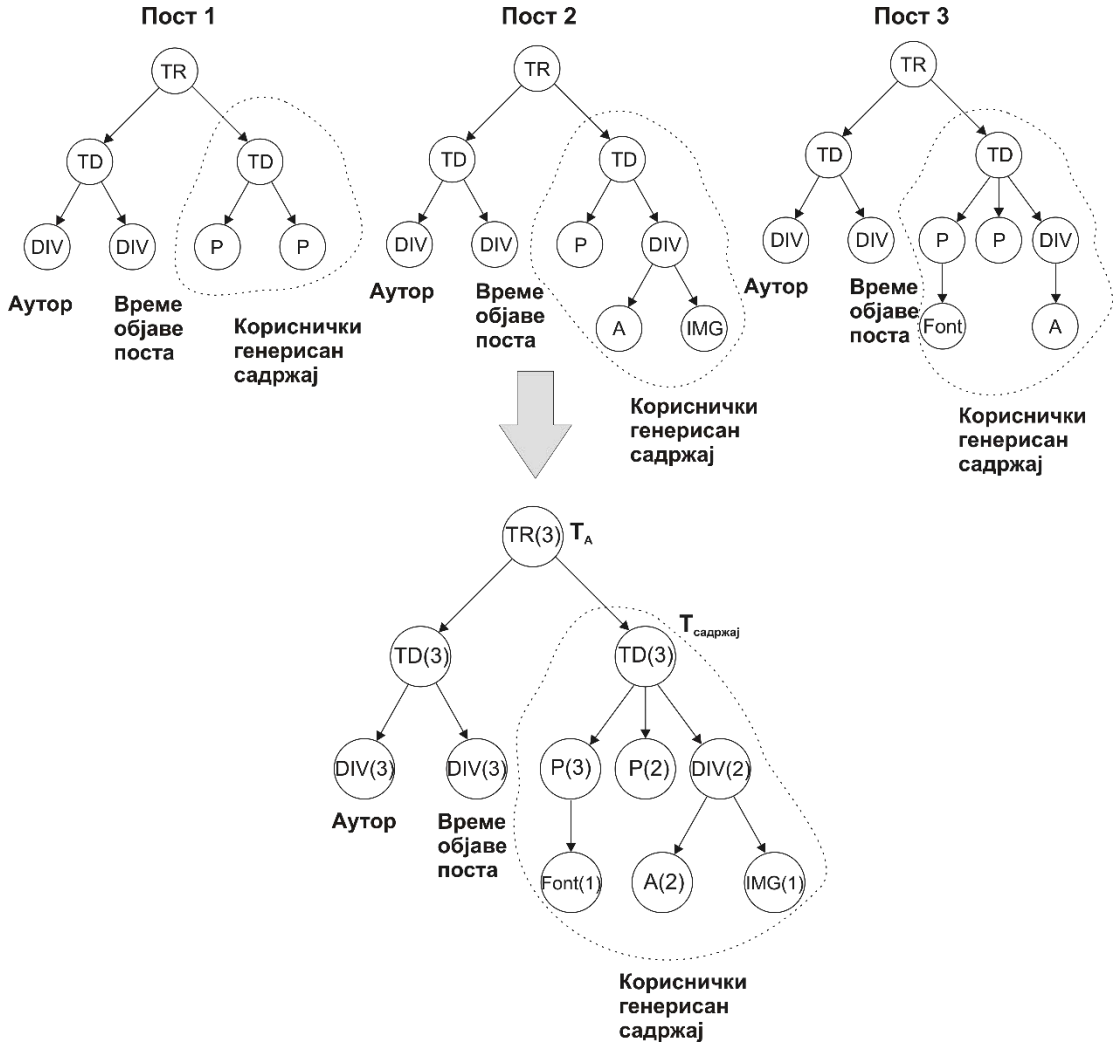
Да би се детектовало које под стабло садржи кориснички генерисан садржај, прво се сва под стабла групишу, тако да свако под стабло исте семантике заврши у истој групи G. Ово је могуће урадити јер је у претходном кораку одређен тачан број под стабала који формирају један пост. Тако се из горњег примера могу добити 3 групе $\{G_1,G_2,G_3\}$ састављене од под стабала $\{T_2,T_5,T_8\}$, $\{T_3,T_6,T_9\}$ и $\{T_4,T_7,T_{10}\}$ респективно. На основу већ поменутих опсервација, под стабла која садрже кориснички генерисан садржај ће се налазити у истој групи. Да би се одредила група у којој се налази тражени садржај користи се формула за израчунавање стандардне девијације сваке групе G_i:

$$SD(G_i) = \sqrt{\frac{\sum_{j=1}^{|G_i|}(N_j - \bar{N} + 1)^2 (W_j - \bar{W})^2}{|G_i|}} \tag{5.2}$$

где $|G_i|$ представља број под стабала у групи G_i, N_j је број чворова j-тог под стабла групе G_i, $\bar{N}$ је просечан број чворова под стабала групе G_i, W_j је дужина текста j-тог под стабла групе G_i и $\bar{W}$ је просечна дужина текста под стабала групе G_i. Група која има највећу стандардну девијацију се узима као група која садржи под стабла са кориснички генерисаним садржајем.

Када се детектује под стабло које садржи кориснички генерисан садржај, потребно је извршити екстракцију чистог садржаја. Обично ово није само један чвор овог под стабла, него неко мање, али опет комплетно стабло, састављено од више чворова. Међу свим семантичким деловима стабла поста, део који садржи овај садржај је најмање конзистентан у смислу дужине текста и структуре стабла. Да би овај процес екстракције био успешан, пролази се кроз три корака (1) прво се формира супер-стабло на основу свих под стабала групе G_i која је одређена као група која садржи кориснички садржај (2) мери се конзистенција сваког чвора у добијеном супер-стаблу, на основу чега се потом мери конзистенција сваког његовог под стабла (3) врши се екстракција садржаја. Ова три поступка се укратко описују у даљем тексту.

Формирање супер-стабла од више стабала је процес поравнања тих стабала да би се конструисало најмање стабло које садржи сва дата стабла. Цео процес је илустрован на Слици 5.7. У овом процесу се користи алгоритам поравнања два стабла предложен у [56]. Поступак је такав да се прво сва под стабла селектоване групе G_i сортирају опадајуће по величини, тј. броју чворова.



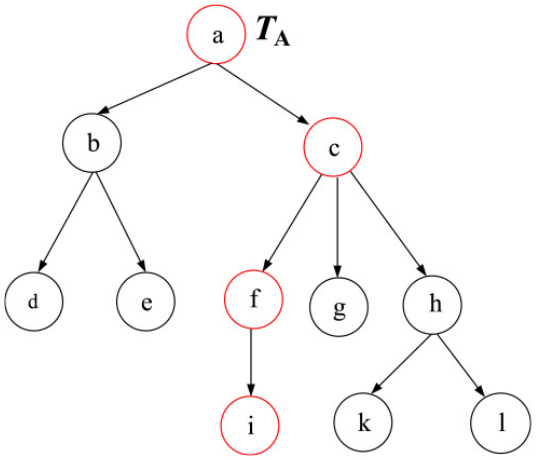
Слика 5.7 – Пример креирања супер-стабла од више под стабала [49]

Потом се прва два најмања под стабла поравнавају у једно стабло. Овако добијено стабло се поравнава са новим, следећим по величини стаблом из групе. Овај процес се понавља док год има преосталих под стабала у групи G_i. Преостало једино стабло на крају је тражено супер-стабло. Приликом процеса поравнања за сваки чвор се памти колико пута је био поклопљен, тј. колико се пута тај чвор појављује у осталим под стаблима изабране групе. Ово је битна информација у преосталом делу процеса. На Слици 5.7 је овај број представљен у сваком чвору у загради.

Да би се одредило под стабло супер-стабла у којем се налази чист садржај, довољно је пронаћи његов корен. Ако је супер стабло обележено са T а стабло садржаја $T_{садржај}$, онда се ово супер стабло може поделити на два стабла $T_{садржај}$ и T-$T_{садржај}$. Постоје две разлике између ових стабала; прва, да је број поклапања чворова обично мањи у $T_{садржај}$-у него у другом. Друго, ако се посматра чвор у $T_{садржај}$ његов текст се обично разликује од стабла до стабла из групе G_i. Да би се направила разлика између $T_{садржај}$ и T-$T_{садржај}$ дефинише се конзистенција чвора на следећи начин:

$$consis(a) = \frac{m}{n}\left(\sum_{i=1}^{m} -\frac{|W_i|}{|W|}\ln\frac{|W_i|}{|W|}\right) \tag{5.3}$$

Где је a чвор из супер-стабла, n је укупан број постова, m број поклапања чвора a, $|W_i|$ дужина текста чвора a из i-тог поста и $|W|$ укупна дужина текста a, из свих постова. Први део једначине, m/n мери конзистенцију чвора a у структури стабла, док други део једначине мери ентропију, тј. количину конзистенције дужине текста чвора a. Ако се узме у обзир конзистенција чворова из $T_{садржај}$-а, онда важи $consis(a_i) < consis(a_j)$ где је a_i из $T_{садржај}$-а а a_j из T-$T_{садржај}$. На основу конзистенције чвора, дефинише се конзистенција стабла као просек конзистенција свих његових чворова $consis(T)$.



Слика 5.8 – Пример екстракције и детекције корена стабла у којем се налази само кориснички генерисан садржај [49]

Користећи конзистенцију стабла, могуће је пронаћи корен стабла у којем се налази чист садржај. Међу свим могућим под стаблима супер-стабла управо $consis(T_{садржај})$ има највећу конзистенцију. На Слици 5.8 је илустрован пример рада поменутог алгоритма. Почевши од корена формираног супер-стабла a, рачунају се консистенције за његова два под стабла чије корене чине његови синови b и c. Пошто је $consis(b) < consis(c)$ памти се чвор c и силази се у њега. Потом се поступак понавља за његова два сина. Овај поступак се понавља док се не дође до листа. Запамћени чворови су a,c,f и i. Највећу конзистенцију међу њима има чвор c, које се узима за корен стабла кориснички генерисаног садржаја.

6 Приказ предложеног система

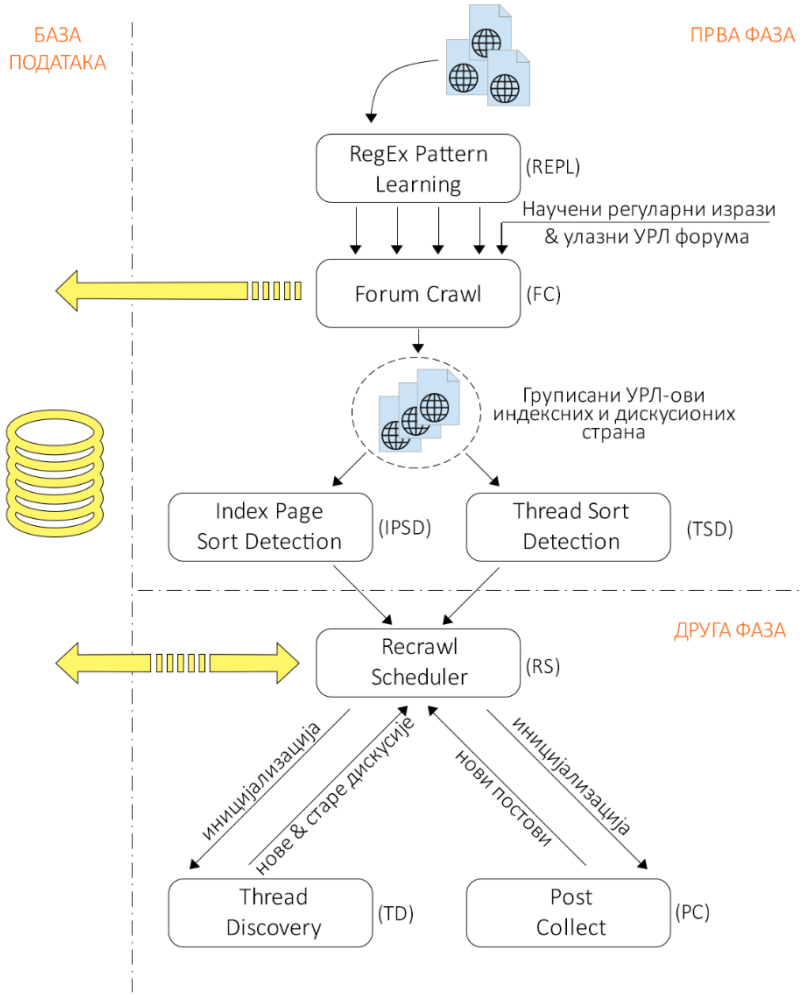
На Слици 6.1 је приказана структура предложеног система за претраживање SInFo (*Structure-driven incremental forum crawler*) чије се функционисање састоји од две фазе. Прва фаза је почетна фаза претраживања која се обавља само једном, када се форум посећује први пут, или након велике промене као што је промена технологије форума. Пратећи скелетне линкове форума, SInFo прикупља све доступне податке и учи редослед сортирања на странама за конкретан веб форум, који се касније користи приликом новог циклуса претраживања. Друга фаза је фаза инкременталног претраживања. Зависно од редоследа сортирања откривеног у првој фази и доступних навигационих опција технологије форума, бирају се адекватне технике за циљање најновијег садржаја приликом следећег претраживања. У сваком новом циклусу поновног претраживања постојеће дискусије се ажурирају новим постовима (свежина), док се на индексним странама претражују нове дискусије (покривеност). Заједно са Сликом 6.1, на Слици 6.2 је дат и псеудо код за боље сагледавање предложеног система. Бројеви линија у псеудо коду на Слици 6.2 наводе се у даљем тексту у заградама.

На почетку прве фазе, Модул REPL (*RegEx Pattern Learning*) учи обрасце регуларних израза који се касније користе за детекцију URL-ова који циљају индексне стране, стране са темама или пагинационе линкове. REPL модулу се даје случајно одабрана страна веб форума који се претражује, уз помоћ које он потом проналази URL почетне стране форума одакле креће са претраживањем. За имплементацију REPL модула су коришћени неки елементи архитектуре претраживача FoCUS који је детаљно описан у претходном поглављу.

На основу случајно изабране стране, REPL модул прво открива почетни URL веб форума (линија 1), јер ова улазна тачка може да варира од форума до форума. Класификатори страна су изграђени уз помоћ SVM-а са подразумеваним линеарним језгром, користећи особине засноване на изгледу страна, одлазним URL-овима, мета подацима и особинама циљаних записа на страни. Неке од главних особина страна које SVM користи су изабране на основу истраживања представљеног у [3] и приказане су у Табели 5.1. Са унапред изграђеним класификаторима страна, почевши од почетне стране, врши се детекција индексних и дискусионих страна, а њихови URL-ови се групишу у скупове који ће касније бити коришћени за тренинг. Овако формирани скупови URL-ова се користе за обуку како би се научили регуларни изрази, који се касније користе да би се уз помоћ њих могли препознати тражени URL-ови. Као што је већ описано, овај алгоритам полази од генеричког узорка „*,“ и проналази све обрасце који одговарају URL-овима у тренинг скупу. Обрасци се рекурзивно пречишћавају док на крају не остану само они који се користе касније као регуларни изрази. Овај алгоритам није превише стриктан, у том смислу да може узимати у обзир и негативне примере, уз додатни праг толеранције за одбацивање URL адреса са слабом покривеношћу. Генеришу се четири врсте регуларних израза, за проналажење индексних страна, индексних пагинационих линкова, дискусионих страна, и пагинационих линкова на дискусијама (линија 1). Ови регуларни изрази се користе од стране претраживача у обе фазе како би се детектовали и позивали искључиво скелетни линкови форума и на тај начин избегле небитне стране. Генерисани регуларни изрази се прослеђују модулу за претраживање форума - FC (*Forum Crawl*), заједно са откривеним почетним URL-ом форума (линија 2).

FC модул врши комплетно претраживање веб форума користећи стратегију претраге по ширини [57]. Полазећи од почетне стране форума, претраживач сакупља све URL адресе које се поклапају са генерисаним регуларним изразима. Нађени URL-ови се затим постављају у ред

за касније претраживање. Овај поступак се понавља све док у реду нема више URL-ова. Сви пронађени URL-ови индексних и дискусионих страна се чувају у бази података заједно са постовима (линија 3). Сваки пост се третира као независни објект у бази података и за сваки пост се, применом *hash* функције, израчунава јединствени идентификатор. Ова *hash* функција користи датум објаве, име аутора и текст поста као улаз за израчунавање јединственог идентификатора. За сваку индексну или дискусиону страну се чува URL стране заједно са последњим датумом претраживања (линије 4-5). URL основне стране представља URL без параметара за страничење, а последње време претраживања представља време последње посете тој одређеној индексној или дискусионој страни.



Слика 6.1 – Структура предложеног система

Поред комплетног претраживања форума, FC модул аутоматски извлачи постове са дискусионих страна користећи софистициране технике. У ту сврху, користе се неки елементи WeRE [49] система, једноставног и свеобухватног решења за екстракцију постова велике тачности који је детаљно описан у претходној секцији. Процес екстракције је подељен на два под задатка: детекција региона постова и екстракција садржаја из сваког поста. Да би издвојили постове, шум на дискусионој страни се прво уклања, а границе поста откривају. Ово се врши алгоритмом подударања стабала по нивоима који израчунава сличности између два стабла. За екстракцију садржаја из поста мери се конзистентност сваког чвора поста у DOM стаблу [49]. На основу измерних конзистентности, издваја се минимално под стабло у којем се налази кориснички садржај.

Након завршетка претраживања форума, прикупљени URL-ови се класификују у две групе, URL-ови индексних страна и URL-ови дискусионих страна, одакле се после користе у

модулу за детекцију сортирања индексних страна IPSD (*Index Page Sort Detection*) и модулу за детекцију сортирања дискусионих страна TSD (*Thread Sort Detection*), како би се касније редослед сортирања садржаја исправно могао искористити приликом претраживања (линија 6). Детектоване информације о редоследу сортирања даље се прослеђују у инкрементални део претраживача, који представља другу фазу (линија 7) процеса претраживања (Слика 6.1).

Друга фаза започиње распоређивачем инкременталног претраживања RS (*Recrawl Scheduler*) (линије 7-16). Овај модул открива нове дискусије посећујући поново сваку индексну страну форума (покривеност) и одржава све претходно преузете дискусије ажурним прикупљањем нових постова са њих (свежина). У случају свежине, RS модул ће иницијализовати поновно претраживање постојеће дискусије помоћу модула за прикупљање постова PC (*Post Collect*). PC модул ће прикупити само нове постове, генерисане након последњег датума претраживања (линије 13-14). Нови постови се чувају у бази података преко RS модула, где се и ажурира последње време претраживања дискусије када су преузети (линија 15).

У случају покривености, користи се модул за откривање дискусија TD (*Thread Discovery*) заједно са PC модулом. Прво, TD модул открива нове и старе дискусије које садрже постове настале након последњег датума претраживања индекс стране (линије 9-12). Прикупљени URL-ови се затим шаљу у PC модул на прикупљање постова. Нове дискусије се у потпуности претражују, док се старе само освежавају.

IPSD, TSD, TD и PC модули су детаљно описани у наредним секцијама. Већина ових модула користи моделе машинског учења за екстракцију и нормализацију датума на стандардизовани формат, као и проналажење одређених URL-ова унутар пагинационих линкова. Ови модели и алгоритми нису приказани на Слици 6.1, која глобално приказује структуру решења, али су такође детаљно описани у наредним секцијама.

Глобални алгоритам функционисања система

Input / output: *db*: reference to the database
let *pages* be randomly sampled pages of a given web forum;
let *indexURLs* and *threadURLs* be groups, consisting of *indexURL* and *threadURL* URLs respectively;
1. find *entryPageURL* and generate *RegEx* with **REPL** using *pages;* //прва фаза
2. collect all *indexURLs*, *threadURLs* and their *postsData* with **FC** using *entryPageURL* and *RegEx*;
3. store all tuples consisting of logically corresponding (*indexURL*, *threadURL*, *postsData,* hash(*postsData*)) into *db*;
4. **foreach** collected pair of (*indexURL*, *threadURL*) **do**
5. extract pair (*baseIndexURL*, *baseThreadURL*) and store with their corresponding (*lcrawlDate*,*TcrawlDate*) into *db*;
6. *indexSort* = **IPSD**(*indexURLs*); *threadSort* = **TSD**(*threadURLs*);
7. **while** true **do** //друга фаза
8. wait for the *nextCrawl* schedule;
9. **if** *nextCrawl*.*type* equals to “coverage” **then**
10. find new and old *threadURLs* with **TD** using *nextCrawl*.*baseIndexURL*, *indexSort and nextCrawl*.*lcrawlDate*;
11. *nextCrawl*.*threadURLs* = *threadURLs*;
12. **end_if**
13. **foreach** *turl* in *nextCrawl*.*threadURLs* **do**
14. collect only new *postsData* with **PC** using *turl*.*baseThreadURL*, *turl*.*lastTPL*, *turl*.*TcrawlDate* and *threadSort*;
15. store logically corresponding tuples(*indexURL*, *threadURL*, *postsData,* hash(*postsData*), *lcrawlDate*, *TcrawlDate*) into *db*;
16. **end_while**

Слика 6.2 – псеудо код прегледа система

6.1 Екстракција датума

Екстракција датума је важан део SInFo система. Представљени модули у прегледу система морају детектовати датуме на основу којих потом откривају редослед сортирања и одређују најбољи приступ за прикупљање најновијег садржаја са индексних или дискусионих страна. Поступак обраде датума се састоји из два дела: први је сама детекција датума на страни, док је други део трансформација датума у стандардизовани радни формат. Као што ће бити у даљем тексту приказано, због великог броја свих могућих комбинација формата и језика у којима датуми могу да се појаве, модели машинског учења имају предност у коришћењу наспрам класичних метода као што су алгоритми упаривања или регуларни изрази.

6.1.1 Детекција различитих формата датума

Различите технологије веб форума, као и њихове различите језичке варијанте, користе различите формате записа датума. Иако су сви ови формати предефинисани [58], [59], увек може да се појави нека варијација од стране аутора веб форума или локална језичка варијанта која није била предвиђена. Неки од формата који су уочени за време писања овог рада налазе се у Табели 6.1.

ТАБЕЛА 6.1
НЕКИ ОД ПРИМЕРА ФОРМАТА ДАТУМА НА ВЕБ ФОРУМИМА

Бр.	**Формат**	**Коментар**
1.	Нов 25, 2019 или Јул 11, 2020	
2.	Недеља, 05 Октобар 99 или Недеља, 05 Октобар 1999	
3.	Јануар 30, 2015 или Понедељак, Јан 30, 2015	
4.	Децембар 13.	(Без године)
5.	Јан. 21/99	
6.	2019 11 13	(година месец дан)
7.	30 02 2020	(дан месец година)
8.	05 02 1999	(месец дан година)
9.	16 10 12	(година месец дан)
10.	2009-Децембар-11 или 2009-Дец-11	
11.	Феб-Уто-23-99	
12.	15 Јул	(без године)
13.	Сеп 31, '98	
14.	2016Јануар05	(без сепаратора)
15.	13 Јун 2020	
16.	Суб Нов 1 2013	(са тачком или без после скраћенице)
17.	Пон 13 Окт '97	

Као што се може видети из Табеле 6.1, број варијација формата који може да се појави је изузетно велик, поготово ако се узме у обзир да неки од њих могу бити и на различитим језицима. Додатно, у свим представљеним форматима, размак између дана, месеца или године може уместо обичног размака бити и неки од симбола /, \, -, ., , а године написане са четири цифре могу бити са две и обратно. Тако нпр. формат број 7. може имате следеће облике:

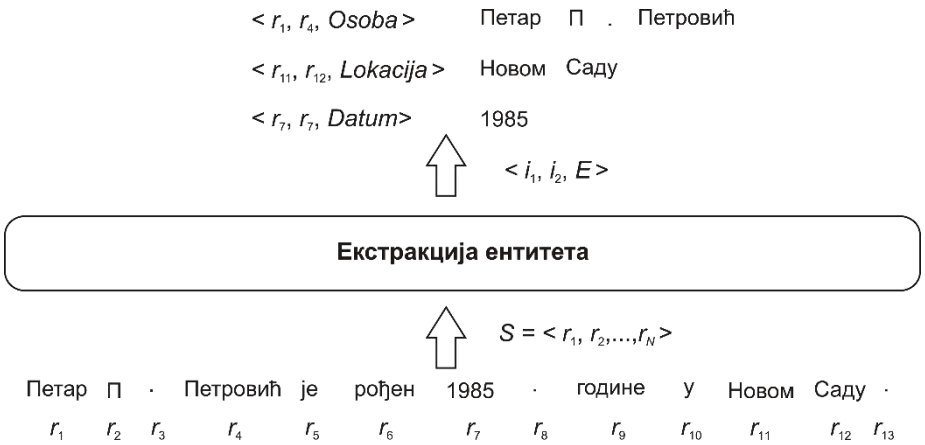
- 30 02 2020
- 30/02/2020

- 30\02\2020
- 30-02-2020
- 30.02.2020

Додатно неки формати немају годину као параметар, него само дан и месец. Иако једноставна метода представљена у [60] даје јако добре резултате, ипак не може да покрије све случајеве који могу да се појаве. Разлог је што ова метода доста зависи од способности програмера да направи добре регуларне изразе који ће да покрију све комбинације. Идеја из поменутог рада је да се направе обрасци, као што је нпр. *дд%мм%гггг*, који ће да се поклапају са форматом на страници, где % може бити било које понављање симбола као што су /, \, -, ., , и где делови датума *дд*, *мм* и *гггг* могу да мењају места и узимају текстуалне облике. Чак и када се ови шаблони независно пишу и на другим језицима, опет је потребно пре сваког претраживања новог сајта, преконтролисати све шаблоне који постоје у систему и поправити или додате нове, чиме се корисниковом интервенцијом губи на аутоматизацији целокупног процеса. Постоји и проблем приликом детекције релативних датума јер се претпоставља да за датуме који су написани у формату ‘5 дана раније’, или ‘пре једног сата’ постоји адекватан прави временски запис у HTML тагу, што на великом броју веб форума није случај.

Коришћење готових софтверских пакета као што су Python библиотека *Datefinder* [61], или *SUtime* [62] Java библиотеке објављене под лиценцом Станфорд Универзитета, су базиране на регуларним изразима и подржавају ограничен број формата, као и мали број језика. У моменту писања овог рада, од стране *SUtime* библиотеке били су подржани Енглески, Шведски и Шпански језик. Са друге стране, постојећа комерцијална решења као што је [63], која покривају већи број формата, нису исплатива за неограничен број коришћења. Такође, за ово конкретно решење не постоји јасна спецификација подржаних формата и језика, што онемогућава планирање претраживања ширих размера.

Да би успеле да се покрију све могуће варијанте датума, па чак и оне које никада пре нису виђене, у овом раду се користи NER (*Named Entity Recognition*) систем машинског учења. NER екстракција ентитета из текста је метод који покушава да пронађе унапред предефинисане ентитете у тексту као што су организација, производ, датум, време, особа и слично [64]. NER системи играју битну улогу у процесирању природних језика и применама као што је екстракција информација IE (*Information Extraction*). На Слици 6.3 се може видети илустрован пример NER система.



Слика 6.3 – Пример NER система

NER, или именовање ентитета, је поступак који детектује реч или фразу која се јасно издваја од скупа других речи са којима има сличне атрибуте [65]. Задатак NER система је да детектује ове речи или фразе и да их класификује у предефинисане категорије. Формално гледано, NER систем добија као улаз низ речи $S = \{r_1,r_2,\ldots,r_n\}$, а као излаз даје низ уређених торки (i_1,i_2,E) где i_1 и i_2 представљају почетни и крајњи индекс у S док E представља унапред одређен ентитет којем низ између i_1 и i_2 припада [66]. Тренинг сет NER система се састоји од анотираног текста, где се детектују ентитети и остале речи које нису битне за детекцију (Табела 6.2.) Остале речи су анотиране тагом O, која означава "***O**utside of a named entity*", тј. изван именованог ентитета. Пошто тражени ентитет може да се састоји од више речи, када се детектује, означава се са B-ENT или I-ENT, где B означава почетак ентитета "***B**eginning of a named entity*", I означава део ентитета који треба придружити и који је спојен на B-ENT "***I**nside of a named entity*", док ENT представља таг за сам ентитет као што је локација (LOC), особа (PER), година (DATE) и сл.

ТАБЕЛА 6.2
ПРИМЕР АНОТИРАНОГ ТЕКСТА ТРЕНИНГ СЕТА ЗА NER СИСТЕМ

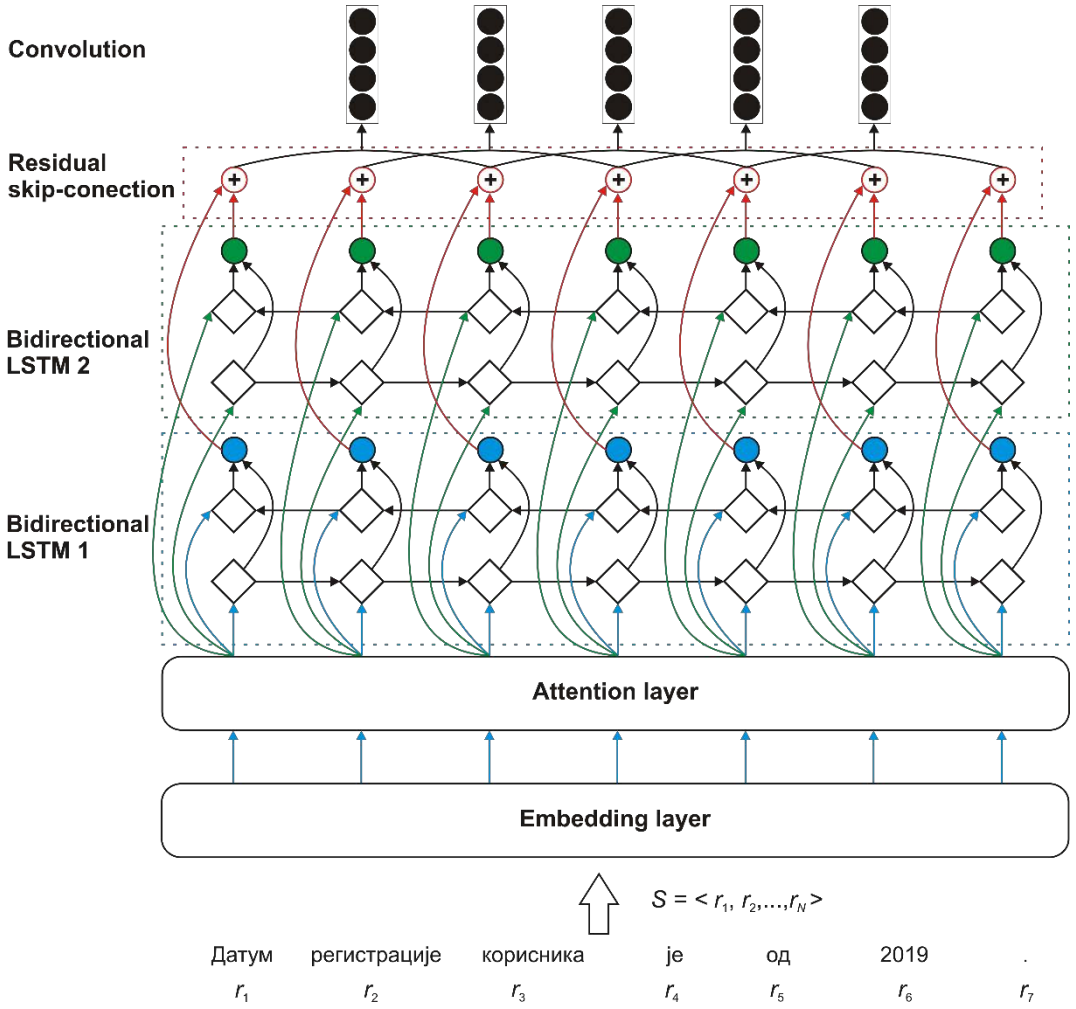
Анотиран излаз	B-PER	I-PER	I-PER	I-PER	O	O	B-DATE	O	O	O	B-LOC	I-LOC
Улазни текст	Петар	П	.	Петровић	је	рођен	1985	.	године	у	Новом	Саду

Добар преглед постојећих NER архитектура се може пронаћи у [66], где се сви представљени системи могу поделити у: (1) системе засноване на правилима којима нису потребни обележени подаци, него се ослањају на ручно креирана правила; (2) системе са ненадгледаним учењем, који не користе ручно анотиране тренинг скупове; (3) системе са надгледаним учењем; (4) системе базиране на дубоком учењу. Преглед јавно доступних већ истренираних NER система приказан је у Табели 7.6. Као што је показано у експериментима, иако веома добри, ови NER системи нису у стању да задовоље задатак дефинисан у овом раду, док неки од наведених система чак и не подржавају екстракцију ентитета датума. Из тог разлога, за потребе овог рада је коришћена индивидуално креирана и засебно тренирана NER архитектура. Од свих познатих архитектура које постоје, у овом раду је коришћена модификација Attention-BiLSTM-CNN [67]. Ова архитектура има на почетку *Attention* слој [68], [69], који се надовезује на Би-дирекциони LSTM (BiLSTM *Bidirectional LSTM*) [70] и на крају на излазу садржи конволуциони слој CNN [71].

Иако се у пракси најчешће користи архитектура где се свака реченица прочита од почетка до краја и обрнуто користећи BiLSTM слој, где се потом резултат шаље у CRF (*Conditional Random Fields*) [72] слој, комбинација BiLSTM-CNN се показала као ефикаснија варијанта [73], јер оваква структура омогућава да модел научи изузетно комплексне релације на великом сету података. Међутим, сваким правцем читања реченице BiLSTM слојем се види само прва половина секвенце која је прочитана у сваком временском кораку. За сваку реч, LSTM читањем са лева на десно види само прошлост, док за сваку реч читања LSTM-а са десна на лево се види само будућност. Ово је прихватљиво за системе као што су класификација текста или машинско превођење, док са друге стране за NER системе ово може да представља ограничење. Разлог је што свака реч садржи своје тренутно скривено стање које није у могућности да моделује шаблоне укрштања будућег и ранијег текста на основу чега би се касније семантика речи могла боље утврдити и доделити исправном ентитету. Додавањем *Attention* слоја на почетак BiLSTM-CNN архитектуре, овај модел је у могућности да у сваком временском кораку ухвати интеракцију између прошлости и будућности у реченици, чиме се прецизније могу одредити релације између самих речи и припадности одређеним ентитетима [74].

Слика 6.4 представља архитектуру NER система коришћену у овом раду. Идеја је да се користе два BiLSTM слоја чији се резултат спаја пре CNN слоја. Ова конекција је инспирисана резидуалним заобилазним конекцијама ResNets (*residual skip connections*) [75]. Овако креирана

конекција омогућава моделу да научи да мапира бијективну функцију, прескачући други BiLSTM слој ако има потребе, али такође омогућава и да научи нешто додатно. То значи да NER систем за време учења може сам да бира пут кроз модел којим ће да се креће. Даље, дужина улазне секвенце у речима која улази у NER системе је увек унапред предодређена и представља фиксну дужину, тако да се дуже реченице одсецају на почетку или крају, док се краће реченице допуњују са специјалним симболом [пад], и то обично на почетку секвенце, тако да се остале речи поравнавају са десне стране. У NER систему представљеном у овом раду, дужина улазне секвенце је 512. Комплетан модел је написан у *Keras* [76] софтверском пакету, и позив његове функције *summary()* за испис архитектуре модела се може видети на Слици 6.6. Овај испис даје спецификацију свих слојева, њихову повезаност и димензије.



Слика 6.4 – Илустрација NER система коришћеног у овом раду

Наведени модели за екстракцију ентитета се најчешће користе над текстуалним подацима, тј. текстом који има одређену семантику, док код размотавања HTML DOM стабла и претраге за датумима на веб страни то не мора увек да буде случај. Некада датум може бити записан у HTML коду и није видљив за корисника у самом старту, или једноставно текст нема очекивану семантику.

Примери случајева који могу да се појаве су:

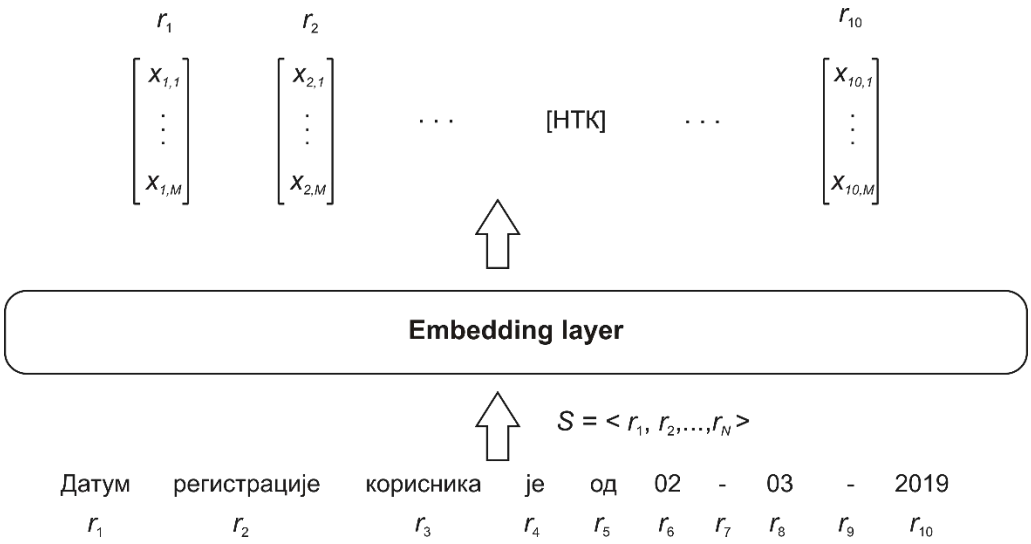
1. `User registration date was on 02-03-2019 07:01`
2. `Post » alan smithee # 05 Sep 2005, 10:48 @tagged`
3. `<tbody datetime="22-04-2020" pubdate="" onmouseover="">3 weeks ago by EllasHoplite</tbody>`

4. `<section class="single"> <article> <time class="pub-date" datetime="2020-4-22 15:3" pubdate=""> <span class="date">22.04.2020</span> |<span class="time">15.03</span> </time>`

Пример под 1. је једини који има семантику природног језика и из које ће сваки добро тренирани NER систем да препозна датум. Пример 2. је пример где и даље NER систем добија текст на улазу али са смањеном очекиваном семантиком, што може представљати проблем за правилну екстракцију ентитета. Примери 3. и 4. су са HTML кодом, где је пример 3. комбиновани текст и HTML код где се у тексту налази релативан, а у тагу прецизан датум, док је пример 4. чист HTML код одакле треба да се препознају ентитети датума.

На основу великог број опсервација приликом писања овог рада, уочено је да се највише појављују примери под 2. и 3. У највећем броју случајева се датум на посту појављује у формату 2., док ако је релативан, онда је потребно скенирати и HTML код око тог релативног датума да би се проверило да ли се у неком од тагова налази и прецизан датум. Случај који има највећу језичко семантичку исправност под 1. је у исто време и најређи.

Приликом слања наведених реченица у NER систем, могу да се појаве два проблема. Први је да реченица нема довољну семантичку логику природног језика да би систем могао правилно да препозна ентитете. Други проблем је да систем не може да препозна специјалне симболе и речи које се појављују у улазном тексту или HTML коду - Табела 6.3. Сваки NER систем, пре него што почне да обрађује реченицу, мора да конвертује улазне речи и симболе у бројеве тј. у векторску репрезентацију како би могао да врши калкулације. Да би ово могло да се изведе, пре свих слојева који су описани, постоји слој за превођење речи у векторску репрезентацију који се зове *Embedding layer*, и који користи предефинисани вокабулар као извор речи. Вокабулар за сваку реч или симбол садржи идентификатор у облику броја, тако да се пре прослеђивања реченице у модел, све речи и симболи замене са бројевима тј. векторима (Слика 6.5). NER систем на тај начин ради искључиво са бројевима и тако разуме реченицу и врши тражене предикције. Бројевни идентификатор у најпростијем случају може бити само један број, или чак низ бројева (вектор) који представља значење одређене речи.



Слика 6.5 – Илустрација embedding слоја

Постоје две варијанте формирања овог вокабулара. Први је да се вокабулар сам формира приликом тренинга, на основу чега ће спектар речи искључиво зависити од количине тренинг података одакле ће се извући речи. Други начин је да се користи већ готов вокабулар који је јавно доступан и који је формиран на основу великог корпуса речи користећи неки

обиман извор као што је *Wikipedia*. Предност коришћења готовог вокабулара је што су такви вокабулари тренирани на огромном сету података и садрже изузетно велики корпус речи, чија је семантика у напред подешена. Мана је што, ако се такви вокабулари користе за неке специфичне задатке као што је овај, где се прослеђује HTML код који не садржи валидне речи из говорног језика, постоји могућност да тражена реч или симбол неће бити пронађена. За речи које се не пронађу у вокабулару се ставља посебан симбол [нтк] тј. непознати токен. Вокабулар тако додатно, поред свих речи и симбола, садржи два специјална токена [пад] и [нтк].

Разлог зашто се користи низ бројева уместо једног броја као идентификатора, је да се реч представи са вектором значења у векторском простору семантике говорног језика. Величина овог вектора осликава комплексност значења и речи са сличним значењем унутар тог векторског простора имају сличне векторе. Одавде систем може да предвиди значење речи и пронађе сличну у датом векторском простору, нпр. аналогија "краљ је краљици исто што и човек жени", се кодира у векторском простору као једначина *краљ – краљица = човек – жена*. Оваква евалуативна шема фаворизује моделе који производе димензије са значењем, самим тим представљајући мулти–кластер идеју дистрибуиране репрезентације [77]. У овом раду је коришћен модификован готов, јавно доступан и у напред истрениран вокабулар – GloVe [78]. Величина вектора репрезентације речи у GloVe векторском простору вокабулара, у зависности од жељене комплексности може бити 25, 50, 100, 200 или 300 димензиони вектор. У овом раду се користи GloVe "*Common Crawl*" вокабулар, сачињен од 2.2 милиона речи и 300 димензионог простора [79]. Додатно, овакви вокабулари се углавном формирају независно за сваки језик, јер ако би се спајали, постоји могућност да неке речи које имају исти запис у различитим језицима имају другачије значење у векторском простору, што може да доведе до неправилних предикција NER система.

На основу свега изложеног, за проблем детекције датума који се покушава решити у овом раду користећи NER систем, везане су следеће чињенице:

1. Постоје реченице које могу имати семантику говорног језика и из тог разлога се користи GloVe вокабулар који својим векторским простором представља природан језик.
2. Велики број улазних реченица може садржати по пар непознатих речи или симбола [нтк]. Ово је углавном случај када је улаз у NER систем комбиновани HTML код са деловима текста који могу имати семантичко значење природног језика.
3. Велики број реченица које немају семантичко значење ће садржати у себи један или више датума на различитим локацијама.
4. Неке реченице, као што је чист HTML код, ће садржати јако велик број непознатих [нтк] токена где ће само пар речи и бројева бити правилно кодирано у број или вектор.
5. Веб форуми могу бити на различитим језицима па се самим тим и датуми појављују на тим језицима, сем ако нису садржани у неком атрибуту тага HTML кода.
6. Ако је датум у релативном формату, онда се он налази у тексту, са или без HTML кода. Релативан датум се скоро никада не садржи у неком атрибуту тага HTML кода.

Да би се решио проблем 5., за сваки језик се користи независан вокабулар. Различити GloVe вокабулари се могу пронаћи на [80]. У пракси се обично за сваки језик тренира посебан модел, јер када се један модел тренира на више језика скоро увек има лошије перформансе од једно-језичког модел [81]. Да не би у предложеном систему имали више модела који би се

мењали у зависности од типа језика веб форума, и на томе евентуално губили на аутоматизацији приликом детекције језика, уводи се вишејезични модел који је трениран на више језика користећи *Transfer Learning* [82]. *Transfer Learning* омогућава да се модел који је већ трениран за један задатак, искористи како би минималним прилагођавањем могао да ради сличан задатак. У овом случају то је разумевање додатног језика. Модел представљен у овом раду је прво трениран на енглеском језику, па се потом додатно тренира за остале језике користећи *Transfer Learning*. Даље, да би се решио проблем где једнојезични модели скоро увек имају боље перформансе од вишејезичних користећи овакав тип тренирања, користе се предложена решења у [81]. Финим подешавањем параметара овако тренираног вишејезичког модела добијају се боље перформансе од једнојезичних модела. Представљени модел у овом раду је трениран на следеће језике: енглески, руски, турски, шпански, румунски, француски, немачки, холандски, италијански, чешки, бугарски, пољски, шведски, кинески (поједностављен), кинески (традиционални), корејски и јапански. Списак језика за тренинг је формиран као сет најчешће коришћених језика форума на светском нивоу. Иако није рађена евалуација за остале језике, у пракси се овај избор показао као довољан да се покрију многе језичке под-варијанте и дијалекти тих истих језика на другим говорним подручјима, као и у потпуности неки језици као што је српски, на који модел није трениран.

Да би се решио проблем под 1. и 6. и да би покрили семантичке делове који могу да се појаве и под 2., као тренинг сет се користи скуп различитих познатих корпуса за тренирање NER система. Ови корпуси су пре 2005 били углавном тривијалног садржаја и садржали су се од анотираног текста преузетог из новинских чланака са малим бројем ентитета. У Табели 7.1 је дат списак свих корпуса коришћених за тренирање предложеног модела. Бирани су корпуси који поред осталих ентитета садрже и датум, и где су на крају сви остали ентитети били одстрањени. Неки од ових корпуса такође садрже релативне временске одреднице и релативне датуме, па се на тај начин покрива и проблем под 6. Иако се може десити да модел трениран са овим корпусом почне да препознаје релативне временске одреднице типа “прошли четвртак”, “следећа недеља” и сл., ипак се показало да ово не утиче на прецизност система. Разлог је да веб форуми углавном не генеришу времена у овако формираном релативном временском запису, па самим тим модел не добија такве записе за детекцију.

Да би решили проблем под 3. прво је преузет велики корпус садржаја постова са одабраних веб форума који се могу видети у Табели 7.8. Одавде су ручно издвојене реченице кратког садржаја које могу садржати датум. Потом су све те реченице ручно анотиране на датуме. Уместо дугих реченица биране су кратке реченице без семантике, јер је то углавном случај када реченица није из HTML кода а садржи датум, као што је већ био дат пример “*Post » alan smithee # 05 Sep 2005, 10:48 @tagged*”. Додатно, за детекцију оваквих кратких реченица са датумима се није користио неки други систем, како тренинг сет не би зависио од излаза другог система. Да би се све ове реченице ручно изабрале и анотирале, коришћен је комерцијални сервис AMT (*Amazon Mechanical Turk*) [83], који се ослања на људске ресурсе и кориснике зарад постизања квалитета садржаја. Да би се додатно осигурао квалитет жељеног садржаја, подаци су редундантно слати различитим корисницима изнова. Број добијених реченица као анотираних датума је дат у Табели 7.2.

Да би се додатно осигурало да модел научи да препознаје релативне датуме, примењен је исти приступ као и за проблем под 3. користећи AMT. Прво се са истих веб форума, користећи ручно написан скрипт, сакупио сав текст који садржи релативан датум, као и околни HTML блок дужине 20 речи/симбола са леве и десне стране. У емпиријским тестовима се ово показало довољно јер ако и постоји прецизан датум унутар HTML блока, који осликава релативан датум у тексту ограниченом са тим HTML блоком, он се неће налазити даље од 20 речи/симбола од тога релативног датума у тексту. Овако формиран текст се ручно анотирао и прочистио користећи AMT.

Приликом сваког анотирања текста, користећи AMT, бележен је формат датума на који се наишло из разлога да би се на крају сакупљања утврдило колико формата фали, и који су формати фаворизовани у односу на друге. Ово је битан корак у креирању боље дистрибуције тренинг сета када је у питању разноврсност формата. Да би се направила добра дистрибуција формата датума за све сетове података који су били формирани, сем за под 1., фаворизовани анотирани датуми замењени су форматима који су фалили, или су једноставно пребачени у друге формате. Цео процес је урађен аутоматски, користећи скрипту написану у *Python* програмском језику и софтверском пакету *faker* [84].

Summary

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 128)]	0	
input_2 (InputLayer)	[(None, 128)]	0	
embedding (Embedding)	(None, 128, 50)	20009000	input_1[0][0] input_2[0][0]
attention (Attention)	(None, 128, 50)	0	embedding[0][0] embedding[1][0]
bidirectional (Bidirectional)	(None, 128, 200)	120800	attention[0][0]
bidirectional_1 (Bidirectional)	(None, 128, 200)	240800	bidirectional[0][0]
add (Add)	(None, 128, 200)	0	bidirectional[0][0] bidirectional_1[0][0]
conv1d (Conv1D)	(None, 128, 32)	19232	add[0][0]
batch_normalization	(None, 128, 32)	128	conv1d[0][0]
dense (Dense)	(None, 128, 128)	4224	batch_normalization[0][0]
dropout (Dropout)	(None, 128, 128)	0	dense[0][0]
batch_normalization_1	(None, 128, 128)	512	dropout[0][0
dense_1 (Dense)	(None, 128, 100)	12900	batch_normalization_1[0][0]
dropout_1 (Dropout)	(None, 128, 100)	0	dense_1[0][0]
dense_2 (Dense)	(None, 128, 5)	505	dropout_1[0][0]

Total params: 20,408,101
Trainable params: 20,407,781
Non-trainable params: 320

Слика 6.6 – Позив функције summary(), модела написан у Keras софтверском пакету

Софтверски пакет *faker* служи за генерисање и симулацију података који су налик правим. Ова процедура је коришћена и приликом генерисања датума који у свом формату имају речи уместо цифара и који су се касније користили за формирања других језичких варијанти тренинг сета.

Да би се решили проблеми под 2. и 4., где се морају користити [нтк] токени који замењују специјалне симболе који се могу наћи у HTML коду а не садрже се у GloVe вокабулару, сам вокабулар је допуњен са овим симболима и речима које недостају. Да би овај процес био урађен што исправније, прво се формирао локални вокабулар сачињен од речи и симбола тренутних података. Потом се направио пресек GloVe и овако формираног вокабулара где су све речи и симболи који не постоје у GloVe сачувани, и где се за сваку сачувану реч и симбол формирао вектор. Скоро све речи добијене на овај начин су припадале HTML спектру. За ове векторе се пазило да буду груписани у независној констелацији семантике језичког векторског простора, и да сви узимају сличан правац и интензитет. Иако једноставан приступ, у тестовима се показао као веома ефикасан и робустан. Ова процедура је поновљена за све језичке варијанте. У Табели 6.3 се може видети излаз из *Embedding* слоја, пре и после додавања специјалних симбола у GloVe вокабулар.

ТАБЕЛА 6.3
ПРИМЕР [НТК] ТОКЕНА У HTML КОДУ КОРИСТЕЋИ СТАНДАРДНИ И ДОПУЊЕНИ GLOVE ВОКАБУЛАР

Улазни текст	<tbody datetime=\"22-04-2020\" pubdate=\"\" onmouseover=\"\">3 weeks ago by EllasHoplite</tbody>
Проширен GloVe	<tbody datetime=\"22-04-2020\" pubdate=\"\" onmouseover=\"\">3 weeks ago by [унн] </tbody>
Улазни текст	<tbody datetime=\"22-04-2020\" pubdate=\"\" onmouseover=\"\">3 weeks ago by EllasHoplite</tbody>
Стандардни GloVe	<[унн] [унн]=\"22-04-2020\" [унн]=\"\" [унн]=\"\">3 weeks ago by [унн]</ [унн]>

Овако формиран и трениран модел се користио за детекцију датума унутар текста или HTML кода. Овај излаз се потом прослеђује у модел базиран на машинском учењу за нормализацију датума који конвертује било који дати формат датума у већ предефинисан и који је описан у следећој секцији.

6.1.2 Нормализација датума

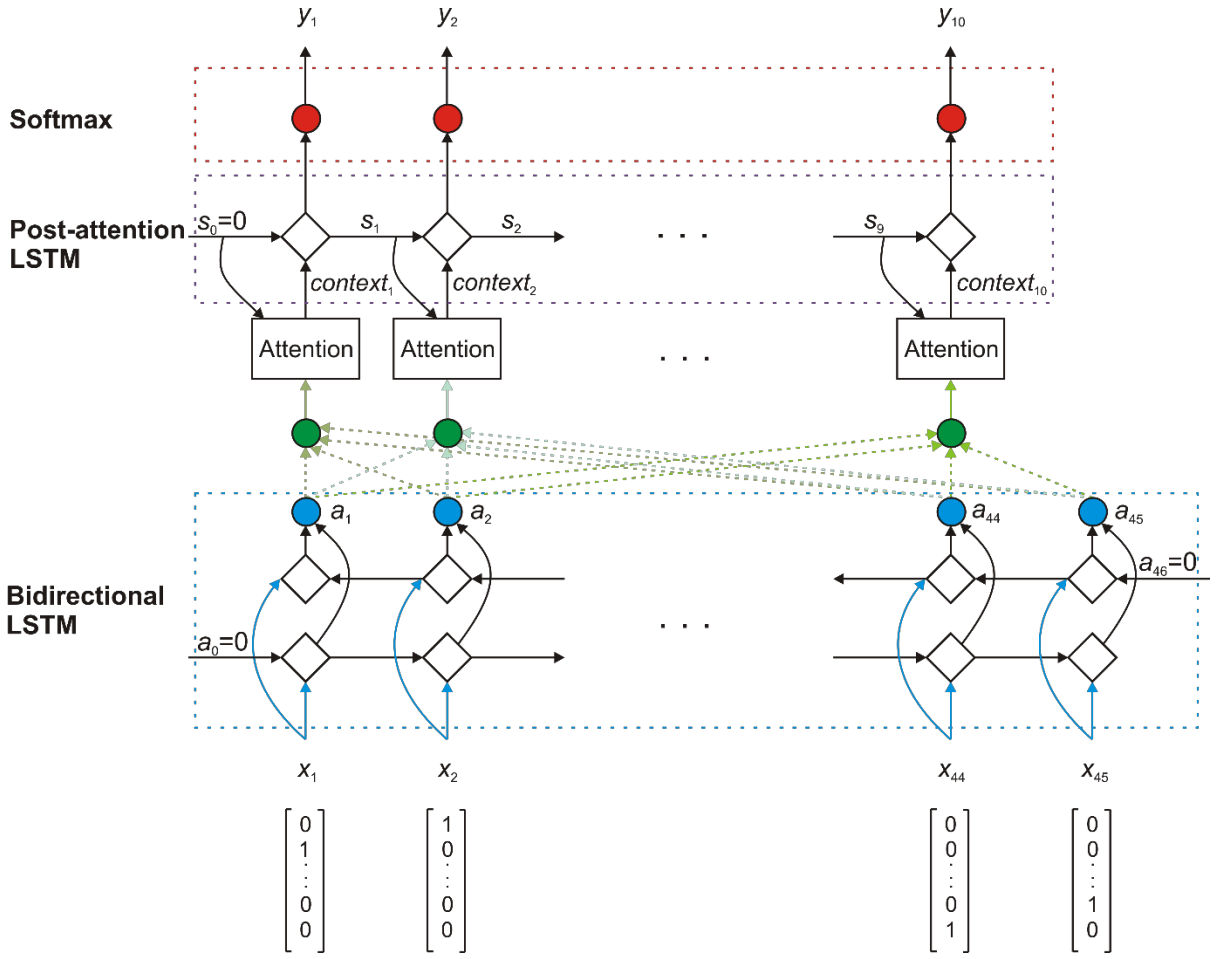
Као што је приказано у претходној секцији, датуми које NER систем за екстракцију ентитета може да пронађе на страни, могу бити у различитим форматима, што прецизним што релативним. Сваки датум који се пронађе је потребно нормализовати и конвертовати у стандардни формат који систем касније може да препозна и који памти у бази података. Усвојено је да овај нормализовани формат буде ГГГГ-ММ-ДД (година-месец-дан).

Иако неки од поменутих готових софтверских решења као што су *Datefinder, SUtime* или *Dateparser* могу да препознају или конвертују ове датуме у предефинисани формат, овде ћемо користити модел базиран на машинском учењу због већ поменутих недостатака који ти пакети имају. Модел који ће бити коришћен у даљем излагању је представљен и развијен у сарадњи са *nVidia Deep Learning* институтом [85] за потребе специјалистичког курса из машинског учења на *deeplearning.ai* [86] веб сајту. Архитектура модела је представљена на Слици 6.7 и инспирисана је *Attention* идејом механизма представљеном у [69].

Улазна секвенца S за овај модел која садржи чист датум је дужине 45 карактера $S = \{X_1,\ldots,X_{45}\}$. Свака улазна секвенца у овом моделу се разбија на карактере уместо на речи, и карактери се прослеђују за сваки временски корак LSTM ћелијама. Из тог разлога овај модел не користи *Embedding* слој, јер вокабулар не представљају речи него карактери. За случај када уместо речи модел на улазу има карактере, уместо идентификатора конкретне речи који би био прочитан из вокабулара, прослеђује се вектор, који има дужину колико има и свих могућих карактера који могу да се појаве на улазу. Свака координата овог вектора је мапирана на карактер који може да се појави, тако да улазни вектор на свим својим координатама има вредност 0, сем на оној координати која представља тај карактер и на којој има вредност 1. Додатно, као и код класичног вокабулара NER система, једна координата је предодређена за

[нтк] тј. непознати карактер, и [пад], за допуњавање секвенце. Ова техника се назива *one-hot encoding* [87] и доста је коришћена у области машинског учења тако да се неће даље описивати. Дужина вектора предложеног модела је 130, и сви карактери су представљани у Табели 6.4. Добијени карактери су сви они карактери који су потребни за формирање датума за све поменуте језике и све њихове формате над којима је модел трениран.

Пре него што се улазна секвенце конвертује у векторе, прво се сва велика слова пребаце у мала, како би величина улазног вектора била мања. Пребацивање великих слова у мала не утиче на семантичко значење датума, а знатно смањује величину улазног *one-hot encoding* вектора. Спектар језика на које је трениран овај модел је исти као и за модел који се користи за детекцију датума, и са свих 130 карактера вокабулара су покривени сви језици и формати које подржава NER систем за екстракцију ентитета датума.



Слика 6.7 – Архитектура модела машинског учења за нормализацију формата датума [86]

Улазна секвенца од 45 карактера је оцењена као довољна, јер се временски део датума одсеца користећи регуларне изразе, и само се чист датум без времена шаље на улаз модела. На основу великог броја прегледаних веб форума, ова дужина је довољна да прихвати и најдуже секвенце датума као што су *Saturday, 19th of November, 1998*. Регуларни израз који је коришћен да би детектовао и склонио временски део датума је:

```
([\s\\\-./,@:]*(\d{1,2}[:]\d{1,2}[\\\-./\s,]*\d*[:\\\-./\s,\d]*[AaPp]?[Mm]?))
```

Разлог зашто се временски део датума не прослеђује моделу је што формата записа времена углавном има у малом броју [58]. Ово омогућава да се временски делови датума детектују регуларним изразом и директно конвертују користећи неке од стандардних библиотека за рад са временом као што је на пример библиотека *time* у *python*-у. После детекције времена од стране наведеног регуларног израза, одстрањено време се прослеђује скрипти написаној у *python*-у где се конверзијом стандардизује у формат *hh:mm:ss*, који преставља сате, минуте и секунде респективно.

ТАБЕЛА 6.4
КОРИШЋЕНИ ВОКАБУЛАР КАРАКТЕРА И СИМБОЛА ПРЕДЛОЖЕНОГ МОДЕЛА ЗА КОНВЕРЗИЈУ ФОРМАТА ДАТУМА

карактер		'	(	)	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	\	^
ид.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
карактер	_	a	b	c	D	e	f	g	h	i	j	k	l	m	n	o	p	r	s	t	u
ид.	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
карактер	v	w	y	z	Á	â	ä	å	ç	é	ì	í	ö	ú	û	ü	ý	ă	ą	č	ě
ид.	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
карактер	ğ	ı	ł	ř	Ś	ş	ź	ţ	а	б	в	г	д	е	и	й	к	л	м	н	о
ид.	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
карактер	п	р	с	т	У	ф	ц	ч	ъ	ь	ю	я	一	七	三	九	二	五	八	六	十
ид.	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
карактер	周	四	土	年	日	星	曜	月	期	木	水	火	週	金	금	년	목	수	요	월	일
ид.	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
карактер	토	화	[нтк]	[пад]																	
ид.	126	127	128	129																	

Предложен модел са Слике 6.7 као излаз има секвенцу $Y = \{Y_1,\ldots,Y_{10}\}$ од 10 карактера која представља датум у жељеном конвертованом формату, ГГГГ-ММ-ДД. Постоје два одвојена LSTM слоја у датом моделу *Pre-Bidirectional LSTM,* и *post-attention LSTM*. Први се налази пре *Attention* слоја, док се други налази после овог слоја. Један временски корак *t Attention* механизама је детаљно приказан на Слици 6.8. Улазна секвенца *S* се у *Pre-Bidirectional LSTM* слоју чита бидирекционо, са лева на десно и обрнуто користећи LSTM ћелије и пролази кроз свих 45 временских корака.

Вектори $\vec{a}_t$ се добијају читањем улазне секвенце *S* у напред, док се вектори $\overleftarrow{a}_t$ добијају читањем секвенце *S* у назад. У сваком временском интервалу *t,* ова два вектора се спајају у вектор $a_t = [\vec{a}_t, \overleftarrow{a}_t]$, који се користи као улазни вектор у *Attention* слој.

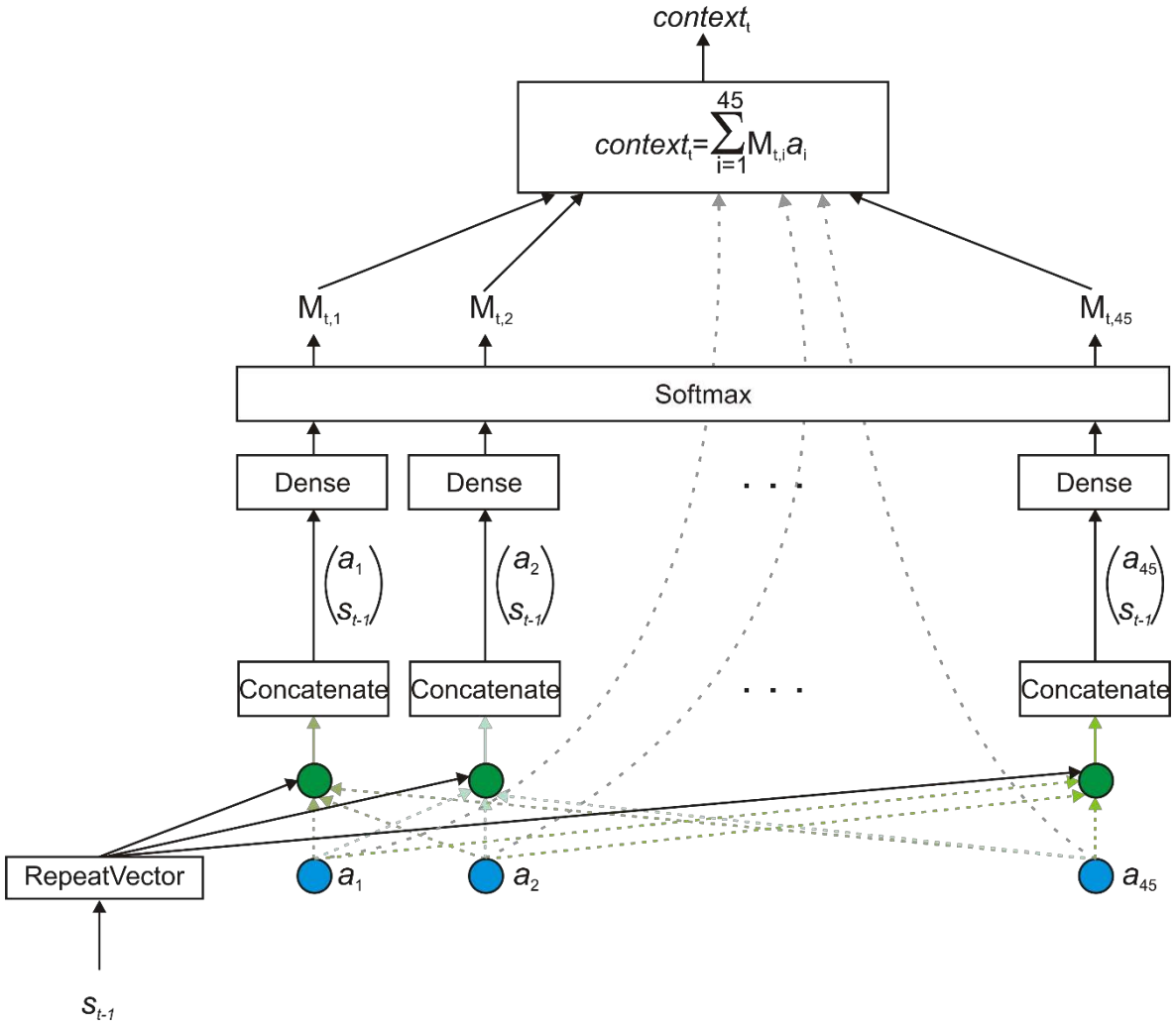
Излаз из *Attention* слоја се користи као улаз у *post-attention LSTM* слој, који пролази кроз 10 временских корака и генерише излазне карактере користећи *softmax* [88] функцију. Ова функција узима улазни вектор од *N* реалних бројева, и нормализује га у дистрибуцију вероватноће која се састоји од *N* вероватноћа пропорционалних експоненту улазних бројева. У овом случају, улазни вектор има 10 координата и представља бројеве од 0 до 9 и симбол '-', који се користе за формирање излазне секвенце *Y*. После примене *softmax* функције на улазни вектор, све вредности се распоређују у интервал (0,1) и могу се интерпретирати као вероватноће, одакле се узима вредност са највећом вероватноћом и мапира у еквивалентни карактер на излазу. Овај поступак се понавља за свих 10 излаза чиме се формира излазна секвенца.

У стандардним применама генерисања текста и машинског превођења, користећи LSTM и сличне механизме, обично се излаз генерисан у временском интервалу *Y(t)* прослеђује на улаз LSTM ћелије временског интервала $Y_{(t+1)}$ јер се претпоставља да свака будућа

генерисана реч или карактер зависи од претходне. У овом моделу то није случај, него се на улаз LSTM ћелије временског интервала $Y_{(t+1)}$ прослеђују скривено стање ћелије $S_{(t)}$ генерисано у претходном кораку. Разлог је што у излазном формату, тј. генерисаној секвенци ГГГГ-ММ-ДД, не постоји строга зависност од суседних карактера или речи као што је то случај у генерисању природног језика.

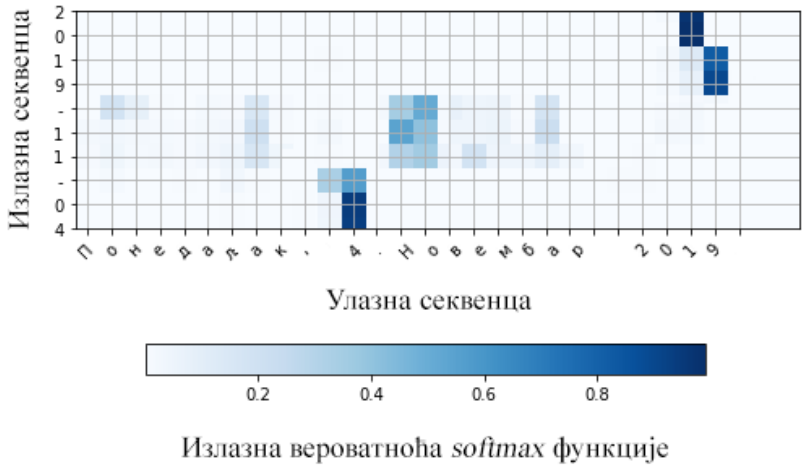
На Слици 6.8 је детаљно приказан *Attention* механизам, док пролази кроз један временски интервал *t*. Идеја овог механизма је да генерише вектор контекста $context_t$ који дефинише колико и којим тачно улазним векторима a_t у тренутку *t*, треба посветити пажње. Идеја се базира на чињеници да само одређени вектори улазне секвенце утичу на генерисање излазног вектора у одређеном временском тренутку, што овај слој учи током тренирања.

Битан параметар $M_{t,i}$ је податак о томе колико контекста ће зависити од активација које се добијају из различитих временских периода. Да би се израчунао $M_{t,i,}$ прво се користи *RepeatVector* да би се улазни вектор $s_{(t-1)}$ ископирао на сваки од *a* вектора са којим се врши спајање. Пошто се природно намеће да $M_{t,i}$ зависи од ова два параметра, али без да се зна тачно у којем односу, за рачунање $M_{t,i}$ се користи мала неуронална мрежа *Dense*, која учи функцију пресликавања из $[s_{(t-1)}, a]$ у $M_{t,i}$. На крају се $context_t$ дефинише као сума свих излаза $M_{t,i}$ распоређена са активационим функцијама *a*.



Слика 6.8 – Attention механизам модела за конверзију формата датума [86]

Да би се лакше визуелно представио *Attention* механизам, приказан је пример где се датум "*понедељак, 4. Новембар 2019*" конвертује у "*2019-11-04*" на Слици 6.9. Предност овог механизма је што зна да сваки део излазне секвенце, као што је месец, зависи од малог дела улазне секвенце, а то су у овом случају почетни карактери који описују месец. Слика 6.9 представља визуелизацију шта *Attention* механизам посматра и у ком делу улазне секвенце да би генерисао тражени излаз. Занимљиво је приметити како модел потпуно игнорише реч *понедељак*, јер је научио да она нема никаквог ефекта на генерисање исправног датума на излазу. Такође се може уочити да је број 4. исправно конвертован у 04, а *Новембар* у 11, где само првих пар карактера има утицај на одређивање броја месеца. За годину је научено да нема потребне обраћати пажњу на прве две цифре, и да су последње две довољне да се утврди која је година у питању, тако да *Attention* модел фокусира тежине на бројеве 19 чиме се генерише 2019. Разлог овако научене логике је што су године које су коришћене у тренинг сету у распону од 1970 до 2050.



Слика 6.9 – Визуелизација зависности излазних карактера од улазне секвенце

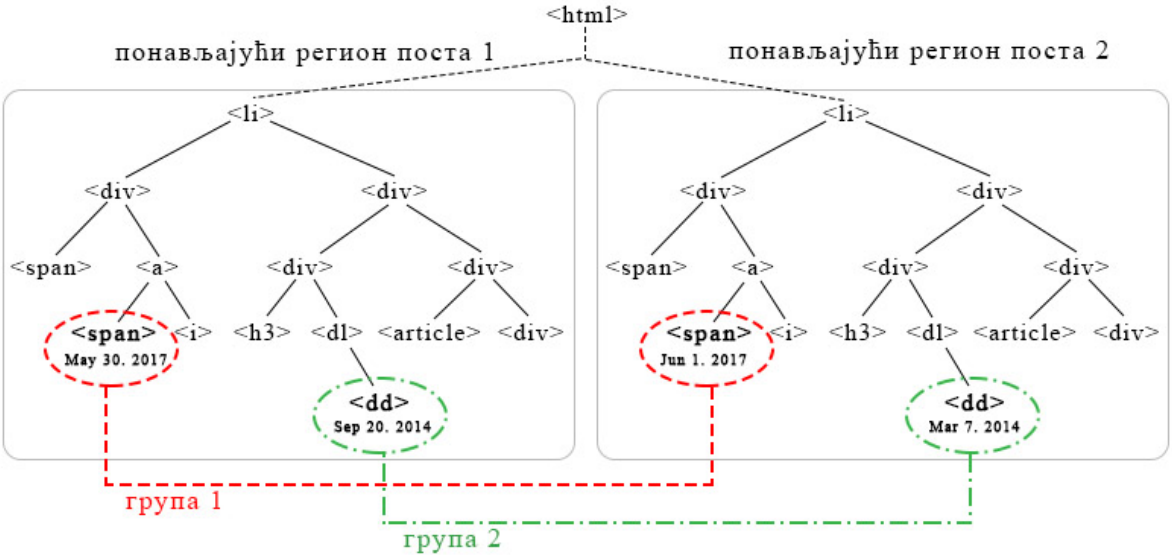
За тренинг је коришћен скуп формиран од јавно доступних сетова и додатно вештачки креираних датума користећи *python* библиотеку *faker*. Сви коришћени датуми су били у форматима који су представљани у Табели 6.1 и на наведеним језицима. Распон коришћених година је од 1970 до 2050, јер се претпоставља да године датума објаве постова на веб форуму не могу бити ван овог опсега. Као што је приказано на овом примеру, овај модел је успевао да препозна и датуме језика за које није трениран, као што је српски.

6.1.3 Детекција датума на дискусијама и индексним листама

Детекција датума на дискусијама. Као што је приказано на Слици 3.3, сваки пост на страни дискусије има свој датума креирања. Постови такође могу да садрже и друге врсте датума, као што су датум регистрације корисника, или датум последње активности корисника на форуму. Ови датуми се сматрају шумом на страни и аутоматски се одбацују предложеном техником.

Заједничка карактеристика свих корисничких постова су редослед генерисања на страни и начин груписања у блок или табелу која заузима средиште стране. Унутар овог блока или табеле налази се HTML DOM под стабло са постом и представља понављајући шаблон у изворном коду странице, у чијим се чворовима датум и други елементи поста налазе у складу са њиховим визуелним изгледом - Слика 6.10. Поштујући ове структуриране информације о

распореду, при чему се сваки датум поста може наћи на потпуно истом чвору сваког HTML DOM под стабла поста, сви датуми из истог чвора се сакупљају са различитих постова у једну групу уместо да се посматра сваки датум појединачно. Резултат је уређена група која се састоји од датума поређаних у потпуно истом редоследу као и на страни. Да би се идентификовао регион стране где су постови угнеждени и саме границе постова на страни, имплементирана је софистицирана технику предложена у [49]. Ова техника користи алгоритам подударања стабла по нивоима за израчунавање сличности између два под стабла, одбија непотребне елементе као што су URL-ови за пагинацију и открива тачну границу блока који садржи постове.



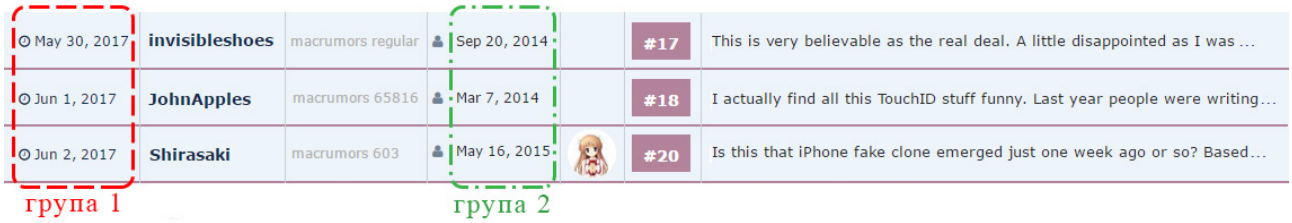
Слика 6.10 – Пример HTML DOM под стабла базираног на прва два поста са Слике 3.3

Након што су границе региона поста откривене, њихов HTML садржај се парсира у HTML DOM стабло, на које се потом примењује делимично поравнавање [47], [48] чиме се креирају структуре налик табелама – Слика 6.11. Ова техника је интензивно изучавана и коришћена претходних година тако да неће бити детаљно образлагана у даљем тексту. Из добијене табеле, селектују се само колоне које садрже датуме у свакој ћелији, где се детекција и нормализација датума врши моделима машинског учења представљеним у претходној секцији.

Свака колона са детектованим датумима је селектована као група кандидат. Пошто су постови генерисани и сортирани по редоследу времена креирања, група која садржи датуме поста мора да задовољи опадајући или растући редослед сортирања, иначе се сматра шумом и одбацује се. У овом правилу се дозвољава да један датум може да буде ван редоследа ако је на почетку секвенце. Ово је правило додато јер на неким форумским технологијама изабрани или важни (*sticky*) пост може бити постављен испред свих осталих, и на тај начин нарушити сортирани редослед датума. Група 1 на Слици 6.11 представља редослед сортирања постова на датој дискусији.

Детекција датума на индексној листи. Овај алгоритам ради слично као и претходни тако да ће бити направљен осврт само на битне разлике. Алгоритам започиње тако што примењује делимично поравнавање HTML DOM стабла на индексној страни. Резултат ове операције је да URL-ови који се налазе на индексној страни буду смештени у колоне структуре налик табели. У датој табели, заједно са URL-овима дискусија остале колоне могу садржати и ћелије са информацијама везаним за дискусије као на пример; датум започињања дискусије, број одговора, број прегледа, датум последњег поста, URL последње активности итд. Бирају се само оне колоне у којима су детектовани датуми моделима машинског учења. Овај алгоритам може пронаћи једну или две групе датума, у зависности од доступних информација

на дискусији. Постоје две врсте група датума; једна група може представљати датуме последњих активности дискусија, док друга представља датуме започињања дискусија. Група која представља редослед сортирања индекс стране као и тип сортирања се детектује у модулу за детекцију сортирања индекс страна.



Слика 6.11 – Пример делимичног HTML DOM поравњања коришћен како би се добиле структуре налик табелама на основу постова са Слике 3.3

Оба алгоритма функционишу на сличан начин - Слика 6.13. Главна разлика је у томе што алгоритам за откривање датума на индекс страни враћа једну или две групе датума, док алгоритам за детекцију датума на дискусијама даје само једну групу датума. Делимично поравнање HTML DOM стабла имплементира се за репетитивне регионе и на индекс и на дискусионим странама како би се створиле структуре налик табелама. Колоне кандидати се бирају детекцијом датума у свакој ћелији коришћењем модела машинског учења. Да би се изабрала колона која садржи датум креирања дискусије, она мора бити у континуираном растућем (ASC - *ascending*) или опадајућем (DSC - descending) редоследу. На овај начин одбацују се колоне које садрже шум, попут датума регистрације корисника или последње активности корисника. За откривање редоследа сортирања на индекс странама бирају се све колоне које садрже датум без детектовања типа сортирања, јер IPSD модул који је задужен за откривање редоследа сортирања захтева на улазу све колоне које садрже датум. Покушај одабира само једне колоне која има опадајући или растући редослед сортирања може бити погрешан, јер све групе датума могу имати исти редослед сортирања. Изабране колоне се касније конвертују у групе датума са истим редоследом у којем су били поређани на страни форума.

6.2 Проналажење одређених URL-ова унутар линкова за страничење

У првој фази претраживања, REPL модул генерише четири врсте регуларних израза; два за препознавање URL адреса индекс и дискусионих страна и додатна два за препознавање URL адреса за њихову пагинацију. Будући да последња два препознају само тип URL адресе а не и где тачно тај конкретан URL показује, то није довољно за каснију и прецизнију навигацију приликом претраживања коју модули из овог система захтевају. Док инкрементално претражују веб форум, модули TD и PC користе много сложеније навигационе методе како би дошли до најновијег садржаја. Да би се постигло задовољавајуће прецизно претраживање и циљање садржаја, док се претражују индекс и дискусионе стране покушава се са проналаском следећих URL адреса:

Слика 6.12 – Примери пагинације без текстуалне интерпретације. а) комбинација слика и бројева - bfmtv.forumactif.org б) само бројеви страна - www.alphaspain.es

- *next (следећа)* – ако се претраживач налази на *n*-тој страни индекса или дискусије, потребно је пронаћи следећу (*n + 1*) страну у низу ако постоји.
- *previous (претходна)*– ако се претраживач налази на *n*-тој страни индекса или дискусије, потребно је пронаћи претходну (*n - 1*) страну у низу ако постоји.
- *first (прва)*– URL који упућује на прву страну индекса или дискусије. Ово може бити и URL основне стране без параметара.
- *last (последња)* – URL који упућује на последњу страну индекса или дискусије.
- *URL последње активности дискусије* – исто као и *last*, осим што се односи на индивидуални URL у блоку дискусије на индекс страни који садржи URL дискусије и њене додатне информације – Слика 3.2ц.

Као што је приказано на Слици 3.2, навигациони блок може садржати одређене URL адресе као што су следећа, претходна, прва, последња или URL последње активности теме, али се оне не могу користити директно интерпретацијом значења текста. Први разлог је у језичкој разлици између веб форума, на пример на форуму који је на шпанском језику, „следећи“ би се означавало као „*siguiente*“ итд. Тумачење свих језика и разлика у дијалектима није ефикасно. Додатно, текстови унутар HTML блока URL-а попут „следећи“ или „претходни“ се могу представити у сличним варијантама као што је „следећих 20 постова“ или „претходних 25 дискусија“ итд.

Све форумске технологије не подржавају или не приказују текстуално значење представљеног URL-а, уместо тога може постојати нестандардно представљање пагинационих страна са само бројевима и сликама - Слика 6.12. Такође, на неким форумским технологијама, блок дискусије на индексној страни не садрже секцију са пагинацијом (Слика 3.2б), него само URL последње активности (Слика 3.2ц) заједно са датумом последњег поста или именом аутора.

Уместо тумачења текстуалног дела URL-а, детектују се одређени URL-ови користећи специфичности URL формата и визуелних информација блока дела стране за пагинацију. У Табели 6.5 су приказане разлике између формата URL-а основне стране и формата URL-а који циља одређену страну са укљученим параметрима. Запажања су:

1. *Специфични број* дефинише број одређене индекс или дискусионе стране, а додаје се на URL-у основне стране као део параметра или само као један број. На пример, https://ubuntuforums.org/forumdisplay.php?f=460&page=2, параметар „page=2“ има специфичан број 2.
2. Специфичан број може бити укључен у URL адресу основне стране. Касније, приликом пагинације, овај број се мења у одређени померај који презентује ту конкретну страну.

3. Специфичан број не мора бити у облику броја циљне стране, већ може бити параметар тј. померај препознатљив само тренутној технологији форума. На пример, URL https://investireinborsa.club/forum/index.php/board,14.50.html има померај специфичног броја 50, што у тренутној технологији форума представља другу страну.
4. Сви URL-ови у групи пагинационих URL-ова садрже исти URL основне стране.
5. Параметри URL-а индекс или дискусионе стране који нису повезани са пагинацијом, остају исти кроз све URL-ове страна приликом њихове пагинације.
6. На основу 1 - 5 следи да се груписани URL-ови за пагинацију страна разликују само у специфичном броју.
7. На индекс страни, ако су пагинациони URL-ови дискусије расположиви налазе се у блоку у којем је пронађен URL основне дискусионе стране.
8. URL последње активности дискусије може бити присутан на индекс страни у блоку у којем се налазе информације о дискусији, чак и када група URL-ова за пагинацију те дискусије није приказана. Такође, URL последње активности дискусије може имати текст који није повезан са бројем стране на коју упућује. Текст може садржати датум поста, име аутора итд., али URL и даље садржи URL основне стране дискусије.
9. Ако URL-ови за пагинацију, груписани у HTML-у блок стране, садрже бројеве у својим текстовима, онда су они приказани у уређеном редоследу. Додатно, на *n*-тој индекс или дискусионој страни, следећа или претходна URL адреса стране ће се налазити поред URL-а који упућује на тренутну страну.

Алгоритам 1: Детекција датума на дискусији

```
input: threadUrl: thread URL from which to extract the post dates
output: grdates: group of post dates ordered as appeared on the thread
1. let grdates be Ø;
2. page = Download (threadUrl);
3. postsRegion = extract HTML region inside which posts are nested on the page;
4. table = align HTML DOM sub trees of all posts in postsRegion;
5. foreach gr in table.columns do
6.         if all gr cells contain date and gr is in dsc or asc sort order then grdates = gr;
7. end_foreach
8. return grdates;
```

Алгоритам 2: Детекција датума на индекс листи

```
input: indexUrl: index page URL from which to extract dates groups
output: dates: group(s) of dates ordered as appeared on the index page
1. page = Download (indexUrl);
2. table = align HTML DOM sub trees of all thread rows from page;
3. foreach gr in table.columns do
4.         if all gr cells contain date then add gr to dates;
5. end_foreach
6. return dates;
```

Слика 6.13 – Детекција датума на дискусијама и индекс листи

На основу ових запажања се предлажу два алгоритма за проналажење специфичних URL-ова; алгоритам за откривање последње активности дискусије - Слика 6.14 и алгоритам за откривање навигационог URL-а за дати правац - Слика 6.15.

ТАБЕЛА 6.5
ПРИМЕРИ ПАГИНАЦИОНИХ URL-ОВА СА ОЗНАЧЕНИМ РАЗЛИКАМА

Форум	Технологија форума	Тип URL-а	URL формат
ubuntuforums.org	vBulletin	Index page (base page URL)	/forumdisplay.php?f=460
ubuntuforums.org	vBulletin	Index page 2nd page	/forumdisplay.php?f=460**&page=2**
ubuntuforums.org	vBulletin	Thread (base page URL)	/showthread.php?t=2357780
ubuntuforums.org	vBulletin	Thread 2nd page	/showthread.php?t=2357780**&page=2**
bbs.bmeol.com	Discuz!	Index page (base page URL)	/forum-28-**1**.html
bbs.bmeol.com	Discuz!	Index page 2nd page	/forum-28-**2**.html
bbs.bmeol.com	Discuz!	Thread (base page URL)	/thread-147-**1**-1.html
bbs.bmeol.com	Discuz!	Thread 2nd page	/thread-147-**2**-1.html
investireinborsa.club	SMF	Index page (base page URL)	/forum/index.php/board,14.**0**.html
investireinborsa.club	SMF	Index page 2nd page	/forum/index.php/board,14.**50**.html
investireinborsa.club	SMF	Thread (base page URL)	/index.php/topic,250.**0**.html
investireinborsa.club	SMF	Thread 2nd page	/forum/index.php/topic,250.**10**.html
www.u-mama.ru	Customized	Index page (base page URL)	/forum/kids/brothers-sisters/
www.u-mama.ru	Customized	Index page 2nd page	/forum/kids/brothers-sisters/**2.html**
www.u-mama.ru	Customized	Index page (base page URL)	/forum/kids/brothers-sisters/758057/
www.u-mama.ru	Customized	Index page 2nd page	/forum/kids/brothers-sisters/758057/**2.html**

Алгоритам за откривање последње активности дискусије на линији (1) проналази све URL адресе из датог HTML блока дискусије. Линије (2-4) упоређују све пронађене URL-ове са пагинационим регуларним изразима наученим у REPL модулу. Ако је URL препознат као пагинациони URL стране, убацује се у промењиву *pagingLinks*. Линија (5), на основу запажања 1 - 6, извлачи специфичан број поравнавањем свих URL-ова из *pagingLinks*-а и проналажењем пресека, тј. разлике. Ако HTML блок дискусије на индекс страни садржи пагинационе URL-ове, онај са највећим специфичним бројем који такође одговара URL-у последње активности ће бити пронађен. Ако URL-ови за пагинацију нису приказани у блоку дискусије, URL последње активности ће бити пронађен као једини URL који одговара највећем специфичном броју. Ако ни пагинациони URL-ови ни URL последње активности нису присутни у блоку дискусије, специфичан број који се враћа је онај који одговара URL-у основне стране.

Алгоритам 3: откривање последње активности дискусије

input: *threadRowHTMLBlock:* thread row from the index page;
output: *laurl*: thread last activity URL if exists
1. *urls* = extract all URLs from *threadRowHTMLBlock;*
2. **foreach** *u* in *urls* **do**
3. **if** *u* matches thread page flipping RegEx **then** insert *u* in *pagingLinks*;
4. **end_foreach**
5. *laurl* = get URL with the largest value of specific number from all URLs in *pagingLinks*;
6. **return** *laurl*;

Слика 6.14 – Алгоритми за откривање последње активности дискусије

Откривање навигационог URL-а за дати правац на линији (1) прво извлачи све групе URL-ова поравнавањем HTML DOM стабла. Скуп URL-ова се сматра групом ако су у HTML DOM стаблу сви URL-ови на истом нивоу и имају највише један родитељски чвор. На линијама (2-6) се врши провера сваке групе и бира се прва која садржи URL адресе детектоване од стране регуларних израза за пагинацију. Регуларни израз за пагинацију који се користи одговара типу стране дате алгоритму - индекс или дискусија. Линије (7-16) узимају у обзир сва 4 могућа смера и проналазе адекватан URL са специфичним бројем из издвојене секвенце.

Исто као и за претходни алгоритам, специфични број се проналази поређењем свих прикупљених URL-ова и проналажењем разлике. На линијама (12) и (15) користи се тренутни URL стране за издвајање тренутног специфичног броја. Ово се постиже привременим уметањем тренутног *curUrl*-а у групу са пагинационим URL-овима *pflgr* и проналажењем

специфичног броја који одговара тренутном URL-у. Ако није пронађен ниједан специфични број значи да је тренутни URL заправо URL основне стране који упућује на прву страну.

Алгоритам 4: откривање навигационог URL-а за дати правац

input: *page:* index or thread page HTML; *curUrl*: index or thread page URL pointing to *page*;
direction: wanted direction (options: next, previous, last, first)
output: *url*: URL for wanted direction
1. *URLgr* = align HTML DOM tree of the *page*, and extract all URL groups;
2. **foreach** *ugr* in *URLgr* **do**
3. **if** all URLs in *ugr* match page flipping RegEx which corresponds to *page* type **then**
4. *pflgr = ugr;* **break**;
5. **end_if**
6. **end_foreach**
7. **if** *direction* eq 'last' **then**
8. *url* = get URL in *pflgr* with the largest specific number;
9. **elseif** *direction* eq 'first' **then**
10. *url* = get URL in *pflgr* with none or smallest specific number;
11. **elseif** *direction* eq 'next' **then**
12. *cursn* = extract current specific number from *curUrl* URL by aligning it with *pflgr* URLs;
13. *url* = get URL from *pflgr* with next specific number in sequence compared to *cursn*;
14. **elseif** *direction* eq 'previous' **then**
15. *cursn* = extract current specific number from *curUrl* URL by aligning it with *pflgr* URLs;
16. *url* = get URL from *pflgr* with previous specific number in sequence compared to *cursn*;
17. **end_if**
18. **return** *url;*

Слика 6.15 – Алгоритми за откривање навигационог URL-а за дати правац

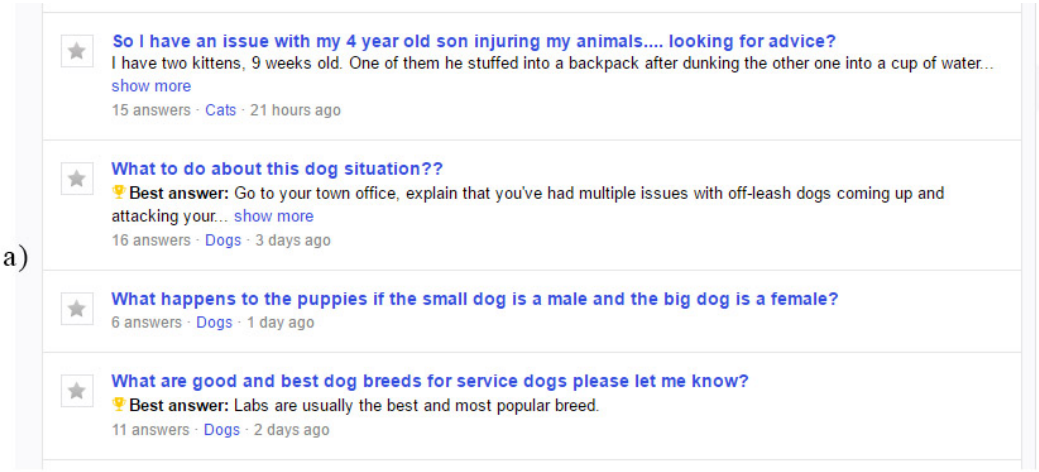
6.3 Детекција сортирања на индекс страни

Откривање редоследа сортирања на индекс страни није једноставан задатак због његове недоследности у распореду и репрезентацији информација кроз различите технологије форума. На основу редоследа сортирања, у овом кораку се одређује како ефикасно претражити индекс страну у TD модулу за откривања дискусија. Редослед сортирања на индекс страни одговара једном од два типа датума који се односе на дискусије са те индекс стране: датум креирања или датум последње активности. На основу запажања на великом броју форумских технологија, установљено је да без обзира на тип датума по ком се сортира индекс страна, тај датум се увек приказују у опадајућем редоследу. У већини случајева датуми креирања и последњих активности приказани су као на Слици 3.2. Постоје и неки случајеви где је приказан само један тип датума или приказани тип датума није тип датума по ком је сортирана индексна страна - Слика 6.16. Запажања су:

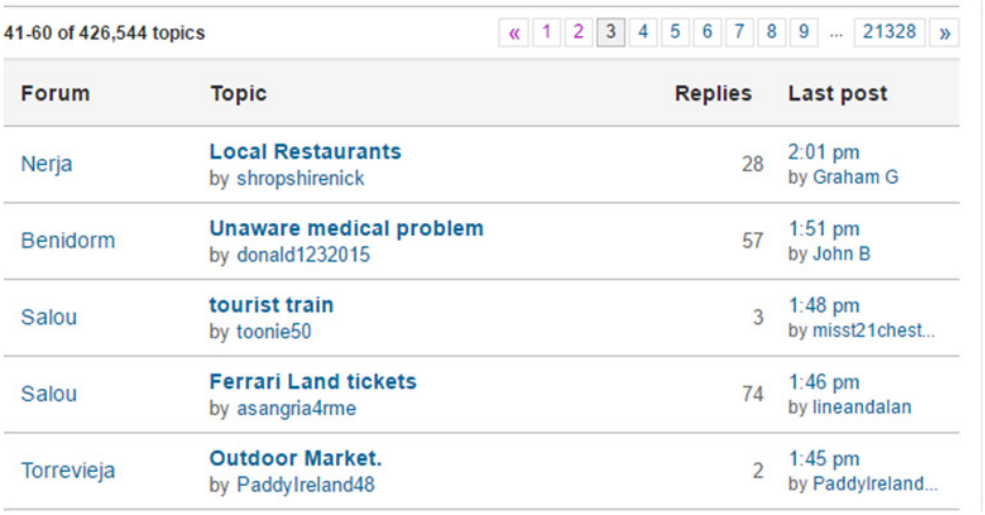
1. Ако су датуми по којима је сортирана индекс страна приказани, они су у опадајућем редоследу.
2. У случају да се два датума појављују на индекс страни, старији датум представља датум креирања дискусије, а новији датум представља последњу активност на дискусији. У случају једног поста на дискусији, ова два датума су иста.
3. Ако индекс страна садржи датуме оба типа, редослед сортирања може да се одреди директно посматрајући који су датуми у опадајућем редоследу.
4. У случају само једног типа датума на индекс страни, потребна је најмање једна посета дискусији како би се одредио тачан тип датума, тј. да ли датум из блока дискусије

припада првом посту или последњем посту те дискусије. Тачан тип датума не може се утврдити директним посматрањем индекс стране, јер индекс страна може бити сортирана према различитом типу датума од оног који је приказан.

а)



б)



в)

Слика 6.16 – Примери сортирања индексне стране. а) answers.yahoo.com – сортирана по датуму последње активности, али је само датум започињања дискусије приказан б) tripadvisor.com - сортирана по датуму последње активности, који је такође приказан в) intopic.it – сортиран по датуму започињања дискусије, али само датум последње активности приказан

На Слици 6.17 представљен је алгоритам за детекцију сортирања на индекс страни, где се користе дата запажања. Линија (2) генерише групу са URL-овима дискусија са дате индекс стране. Потом следи поравнање HTML DOM стабла индекс стране и креирање структуре налик на табелу. Ослањајући се на идеју из [3], претпоставља се да је колона која садржи URL-ове дискусија, колона са најдужим укупним текстом. Одабрана колона се парсира у групу URL-ова дискусија, сортираних као што је приказано на индекс страни. Линија (3) генерише једну или две групе датума користећи промењиву *the Index*.

На линији (4) псеудо-случајни број *n* се генерише и задржава током извршавања алгоритма. Одабрани број *n* представља *n*-ти блок дискусије на индекс страни са URL-ом дискусије и датумима. Случај са две групе датума приказан је на линијама (5-10). Разликовање између групе која представља датуме започињања теме crd (*Creation Date*) и групе која представља датуме последње активности lad (*Last activity date*) се врши поређењем два датума, где је сваки из друге групе али са истим редним бројем - *n* (6-8) . Исти редни број гарантује да су из истог блока дискусије, јер су датуми у обе групе поређани онако како су се појављивали на страни. На основу запажања 1, група која има опадајући редослед је група по којој су сортиране дискусије на индекс страни. Тип датума у одабраној групи узима се као начин сортирања индекс стране (9-10).

У случају само једне групе датума (11-17), прво се URL из групе користи да се дохвати дискусиона страна где се потом са ње врши екстракција датума објаве постова (12) користећи моделе машинског учења. Од добијених датума, први датум поста се упоређује са датумом из групе датума који одговара том одређеном URL-у дискусије (13). Ако су датуми исти, тада група датума детектована на индекс страни представља датуме започињања дискусија (14), у супротном представља датуме последње активности (15). Такође, ако је група датума у редоследу константног опадања, тип датума у овој групи узима се као тип редоследа сортирања индекс стране (16). У супротном, други преостали тип узима се као тип редоследа сортирања (17).

Алгоритам 5: детекција сортирања на индекс страни

```
input: ind: index page URL
output: sort: detected sort - crd or lad
1. page = Download(ind);
2. threads = align HTML DOM tree of the page and extract threads;
3. grp = extract thread dates group(s) from the ind;                    // Алгоритам 2
4. let n ∈ [1, grp.dates1.count];
5. if grp.count equals 2 then
6.       if grp.dates1[n] < grp.dates2[n] then
7.             grp.dates1.type = crd; grp.dates2.type = lad;
8.       else  grp.dates1.type = lad; grp.dates2.type = crd;
9.       if grp.dates1 is in dsc order then return grp.dates1.type;
10.      else return grp.dates2.type;
11. elseif grp.count equals to 1 then
12.      datesGroup = extract post dates from threads[n];               // Алгоритам 1
13.      if grp.dates1[n] equals to datesGroup[1] then
14.            grp.dates1.type = crd;
15.      else grp.dates1.type = lad;
16.      if grp.dates1 is in dsc order then return grp.dates1.type;
17.      else return inverted grp.dates1.type;
18. end_if
```

Слика 6.17 – Алгоритам за детекцију сортирања на индекс страни

Када се упоређује редослед датума допушта се да за низ датума који су на почетку, специфичних за сваки форум, буду ван сортиране секвенце. Разлог је могуће постојање одабраних/фиксираних (*sticky*) дискусија, које се могу појавити на почетку индекс стране и пореметити редослед сортирања датума.

6.4 Детекција сортирања на дискусији

Овај модул користи алгоритам за детекцију датума на дискусијама да би издвојио групу која се састоји од датума објаве постова. Датуми у издвојеној групи су поређани по редоследу као што су приказани на страни дискусије. Детекција врсте сортирања датума у тој групи одређује редослед сортирања постова на дискусији. Овај редослед сортирања може бити опадајући или растући, и у зависности од ове информације касније се у PC модулу за претраживање користе адекватни приступи. Такође је и задат праг од три поста, колико дискусија мора минимално да их садржи, како би се у опште узела у разматрање. Два поста на страни нису довољна за поређење, јер један од њих може бити фиксирани (*sticky*) пост који се појављује на врху свих осталих постова и на тај начин нарушава редослед сортирања.

Алгоритам за откривање типа сортирања дискусије функционише на следећи начин: Линије (1-2) преузимају страну дискусије, поравнавају HTML DOM стабло и издвајају постове користећи методе предложене у [49]. Страна се узима у обзир само ако на њој постоје 3 или више постова. Линија (4) издваја групу датума помоћу алгоритма за детекцију датума на постовима дискусија. Линије (5-8) упоређују свака два узастопна датума поста унутар генерисане групе и инкрементирају адекватне бројаче сортирања (*asccount* или *dsccount*). Метода гласања већине је усвојена за одређивање типа сортирања, јер може доћи до појаве фиксираних постова који могу променити редослед сортирања датума (9-10).

Алгоритам 6: детекција сортирања на дискусији

```
input: threadUrl: thread page URL on which to detect the sort
output: sort: detected sort order asc or dsc
let i be 1; asccount be 0; dsccount be 0;
1.  page = Download(threadUrl);
2.  posts = align HTML DOM tree, and extract the posts from the page;
3.  if posts.count < 3 then return;
4.  datesGroup = detect thread post dates from threadUrl;
5.  while i < datesGroup.count-1 do
6.          if datesGroup[i] < datesGroup[i+1] then asccount++;
7.          else dsccount++;
8.  end_while
9.  if asccount > dsccount then return asc;
10. else return dsc;
```

Слика 6.18 – Алгоритам за детекцију сортирања на дискусији

6.5 Откривање дискусија

У модулу за откривања дискусија (TD) користе се два различита приступа у зависности од типа редоследа сортирања на индекс страни, који може бити по датуму креирања или према датуму последње активности дискусије.

У првом случају индекс страна је сортирана према датуму креирања дискусија. На основу запажања 1 из одељка 6.4, без обзира који тип датума се користи за сортирање индекс стране, ти су датуми сортирани у опадајућем редоследу. Додатно, индекс страна сортирана по датуму креирања има нове дискусије које се појављују на почетку прве индекс стране. На основу последња два тврђења, ако је индекс страна сортирана према датуму креирања сами

датуми на индекс страни нису битни, јер се све нове дискусије могу пронаћи претраживањем првих пар индекс страна док се не наиђе на старе - Слика 3.4а.

Алгоритам започиње прикупљањем свих URL-ова дискусија са прве индекс стране - Слика 6.19. Ако ниједан од тих URL-ова није већ прикупљен у једном од претходних претраживања, користећи URL-ове за пагинацију, алгоритам преузима следећу индекс страну и са ње прикупља URL-ове следећих дискусија. Овај поступак се понавља све док се не открије барем један URL дискусије из претходних претраживања. Када претраживач наиђе на стари URL неке дискусије цео процес се зауставља, из разлога што су сви наредни URL-ови који би се појавили већ прикупљени у претходним претраживањима.

У другом случају URL-ови дискусије на индекс страни су сортирани према датуму последње активности. Индекс стране сортиране по датуму последње активности могу бити састављене од микса старих и нових дискусија. Нове дискусије се креирају након последњег датума претраживања индекс стране, а старе добијају нове постове генерисане од истог тог датума последњег претраживања - Слика 3.4б. Овај случај захтева упоређивање датума последњих активности дискусија са датумом последњег претраживања те индекс стране, јер се жели избећи прикупљање старих URL-ова дискусија које нису ажуриране. Алгоритам почиње с првом индекс страном и прикупља све URL-ове. Затим, за све прикупљене URL-ове посматрају се њихови датуми последње активности.

Алгоритам 7: Откривања тема

```
input: idu: index base page URL; sort: previously detected sort order of idu - lad or crd
       iplcd: last crawled date of idu; DB: reference to the database of the previous crawls
output: threads: collected thread URLs, with last activity URLs if exist
let page be Ø and threadCandidateRows be Ø;
1. page = Download(idu);
2. while page not equal to Ø do
3.      threadCandidateRows = align DOM tree of the page and extract thread rows;
4.      Find all thread URLs (as url), their corresponding dates (as lad) and
        thread last activity URLs if exist (as lpl) in threadCandidateRows;
5.      foreach th in threadCandidateRows do
6.          if sort equals to crd and th.url not in DB then
7.              add th.url and th.lpl into threads;
8.          if sort equals to lad then
9.              if th.lad equals to Ø then
10.                 th.lad = find newest date from th.url;
11.             if th.lad is newer than the iplcd then
12.                 add th.url and th.lpl into threads;
13.     end_foreach
14.     if last th.url is inserted into threads then
15.         idu = find the 'next' index page URL of idu; page = Download (idu);
16.     else
17.         page = Ø;
18. end_while
19. return threads;
```

Слика 6.19 – Алгоритам за откривања тема

Ако су сви датуми новији од датума претходног преживања индекс стране алгоритам ће преузети следећу индекс страну и поновити поступак. Алгоритам се зауставља када је бар један нађени датум последње активности старији од датума последњег претраживања на тој индекс страни. Дискусије које имају датум последње активности старији од датума последњег претраживања се не сакупљају. Касније у PC модулу нове дискусије се потпуно претражују, а старе се само ажурирају најновијим постовима, па се зато прикупљени URL-ови у бази података означавају као нови или стари, на основу већ постојећих URL-ова дискусија у самој бази.

Алгоритам за откривања тема је укратко описан на Слици 6.19. На линијама (3-4), користећи поравнање HTML DOM стабла индекс стране, алгоритам издваја и прикупља URL-ове дискусија користећи исти приступ као у алгоритму за откривање типа сортирања индекс страна. Поред сваког URL-а дискусије, датум последње активности и URL последње активности (ако је доступан) се такође издвајају помоћу алгоритама за откривање датума индекс стране и алгоритама за детекцију URL-а последње активности (4). Користећи URL последње активности дискусије, PC модул може ефикасно да приступа најновијем садржају на старим дискусијама. Линије (6-7) - ако се утврди да је редослед сортирања индекс стране према датуму креирања дискусија (crd), URL се преузима само ако није сакупљен у претходним претраживањима. Ако се утврди да је редослед сортирања према датуму последње активности (lad), URL дискусије се прикупља само ако је датум последње активности новији од датума последњег претраживања те индекс стране - линије (11-12). Ако датуми последњих активности нису приказани на индекс страни, URL-ови дискусије се преузимају, а датуми објављивања постова се издвајају алгоритмом за откривање датума постова на дискусији. Датум последњег поста на дискусији се бира као датум последње активности (9-10). Следећи логику редоследа сортирања, овај поступак се понавља само ако последња прикупљена URL адреса дискусије са индекс стране испуњава захтев за уметање у базу података (14) или док се не дохвати последња страна индекс листе (15-17).

6.6 Сакупљање постова

Модул за сакупљање постова (PC) се користи за индексирање нових постова са дискусије. У случају нове дискусије овај модул ће обићи све њене стране користећи URL-ове за пагинацију и индексирати све постове. За већ постојеће дискусије у бази података, PC модул ће се кретати преко страница дискусије у потрази за новим садржајем на најефикаснији начин, и то на основу откривеног редоследа сортирања дискусије и URL-а последње активности. Препознају се три случаја за пролазак страница дискусије: (1) постови су у опадајућем редоследу, (2) постови су у растућем редоследу, и URL адреса последње активности доступна је на индекс страни, (3) постови су у растућем редоследу а URL последње активности дискусије није доступан - Слика 6.21.

У случају када су постови у опадајућем поретку, последњи пост биће приказан на врху прве стране, а најстарији (тј. први који је креиран на дискусији) биће приказан на дну последње стране дискусије. У овом случају PC модул ће кренути од прве стране и прикупити све постове који нису прикупљени у неком од претходних претраживања - Слика 6.21а. Постови се издвајају као што је предложено у [49], а датуми се откривају алгоритмом за откривање датума на дискусији. Сваки датум поста упоређује се са датумом претходног претраживања. Ако су сви датуми постова на страни новији од датума задњег претраживања користећи пагинационе URL-ове и алгоритам за откривање навигационог URL-а, овај модул ће пронаћи следећу страну дискусије и поновити поступак. Ако је бар један датум поста са следеће стране старији од последњег датума претраживања процес се зауставља. PC модул се такође зауставља када се више не могу пронаћи следеће стране, тј. када се досегне последња страница дискусије.

У случају растућих постова на дискусији, последња страна дискусије ће садржати последњи пост. У овом случају модул користи URL последње активности који проналази модул за откривање дискусија - Слика 6.21б. PC модул почиње са сакупљањем нових постова са последње стране. Сви датуми објаве постова са те стране се упоређују са последњим датумом претраживања дискусије. Ако су сви датуми новији од датума последњег претраживања, помоћу пагинационих URL-ова и алгоритма за откривање навигационих URL-ова, овај модул проналази URL претходне стране дискусије. Процес екстракције поста и

датума понавља се на свакој страни све док се не нађе барем један датум старији од последњег датума претраживања. Овај алгоритам се зауставља и када више нема претходних страна за пронаћи (тј. када се досегне прва страна дискусије).

Алгоритам 8: Сакупљање постова

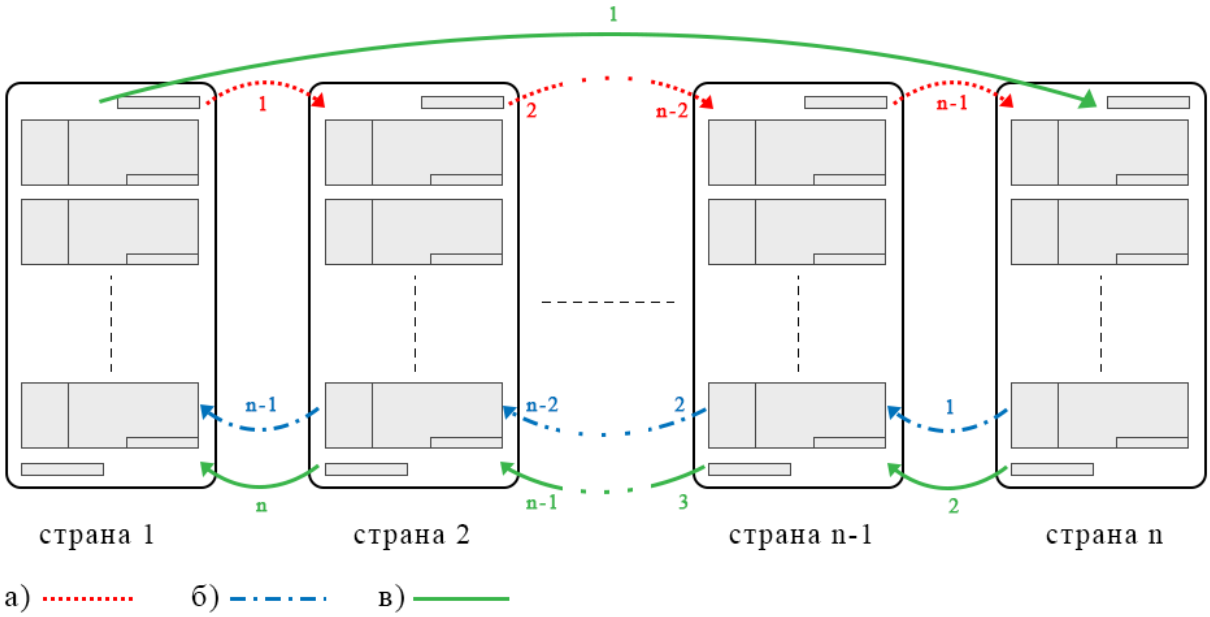
```
input: th: base thread URL; thlpl: thread last activity URL of th if exist; sort: asc or dsc;
       thlcd: last crawled date of th;
output: posts: collected new posts
let page be Ø; allFoundDates be Ø; direction be 'next'      // 'next' – подразмевани правац за случај са Слике 6.21
1.  if sort equal to dsc or thlpl equal to Ø then
2.      page = Download(th);
3.  if sort equal to asc then
4.      if thlpl equal to Ø then
5.          thlpl = using page, try to find URL to the 'last' page of th;
6.      if thlpl not equal to Ø then
7.          direction = 'previous'; page = Download(thlpl); // правац за случајеве са Слике 6.21б и Слике 6.21в
8.  while page not equal Ø do
9.      allposts = extract all posts from the page;
10.     foreach post in allposts do
11.         postdate = extract post date;
12.         insert postdate into allFoundDates;
13.         if postdate is newer than thlcd then
14.             insert post into posts;
15.     end_foreach
16.     if any date from allFoundDates is older than thlcd then
17.         break;
18.     nextPageUrl = find the next thread page for the given direction using pagination URLs;
19.     page = Download(nextPageUrl);
20. end_while
21. return posts;
```

Слика 6.20 – Алгоритам за сакупљање постова

У трећем случају постови на дискусији су поређани у растућем редоследу и URL последње активности није доступан. Постоје два разлога зашто TD модул не може да нађе URL последње активности. Први је да генерисање URL-а последње активности на индекс страни није подржано од стране тренутне форумске технологије или да још увек нема довољно постова на дискусији да би се генерисале додатне стране. У другом случају PC модул ће преузети једину постојећу страну и прикупити само нове постове. У случају када технологија форума не подржава генерисање URL-а последње активности на индекс страни, преузима се прва страна дискусије одакле алгоритам покушава да нађе URL последње стране користећи алгоритам за откривање навигационог URL-а - Слика 6.21ц. Након преузимања последње стране приступ је исти као код случаја растућих постова на дискусији. Преузимају се узастопне претходне странице и извлаче се сви постови и њихови датуми све док постоје нови постови.

PC алгоритам је приказан на Слици 6.20. Овај алгоритам има два смера; *next* - за тумачење случаја приказаног на Слици 6.21а; и *previous* - за случајеве приказане на Слици 6.21б и Слици 6.21в. Улази алгоритма су URL адреса дискусије, откривени тип сортирања, URL последње активности дискусије (ако постоји) и датум последњег претраживања. Линије (1-2) преузимају прву страну дискусије у случају растућег типа сортирања или ако URL последње активности није доступан, при чему је смер подразумевано *next*. У случају растућег редоследа сортирања и када није доступан URL последње активности дискусије линија (5) покушава да пронађе URL последње стране дискусије са тренутно преузете стране користећи алгоритам откривања URL-а последње активности дискусије. Ако је URL последње активности дат алгоритму или нађен на линији (5) смер се мења у *previous* и преузима се последња страна дискусије (6-8). За сваку преузету страну извлаче се постови (10), а из сваког

поста се издваја датум (12) користећи методе већ описане у претходним одељцима. Пост се преузима само ако је датум његове објаве новији од датума последњег претраживања (14-15). Модул се зауставља када се на страни нађе најмање један датум објаве поста који је старији од последњег датума претраживања (17-18), у супротном алгоритам преузима наредну страницу према одређеном правцу (19-20) користећи алгоритам за откривање навигационих URL адреса.



Слика 6.21 – Пролазак страна дискусије када је тип сортирања а) опадајући б) растући, URL последње активности познат в) растући, URL последње активности није познат

7 Експерименти

7.1 Евалуација модела за детекцију и конверзију датума

У овој секцији су описани методи тестирања модела као и процес тренирања са коришћеним подацима. Узимајући у обзир да је за тренинг модела потребно много више времена него за само извршавање, за ова два процеса су коришћене различите хардверске конфигурације. За тренирање је коришћена јача хардверска конфигурација са више процесора и графичком картом, док је за само извршавање коришћен класичан десктоп рачунар са графичком картом високих перформанси прилагођеном за извршавање модела машинског учења.

7.1.1 Тренирање модела

Оба модела су тренирана користећи процесоре и GPU (*Graphic Processing Unit*). Коришћена GPU је специјализована графичка карта за потребе рада са великим бројем калкулација. Модел графичке карте коришћен у ове сврхе је *nVidia Tesla K80* [89] са 12GB радне меморије. Овај модел располаже са 2496 *nVidia CUDA* језгара, и може да обави калкулације до 2.91 тера-флопса двоструке прецизности. Рачунар на којем је инсталирана графичка карта је садржао 4 *Intel's Broadwell* процесора радног такта од 2.7 GHz, као и 61GB радне меморије. Ова хардверска конфигурација је рентирана на AWS *(Amazon Web Services*) веб сервису [90] у форми резервисане EC2 инстанце.

Као што је већ напоменуто, за формирање једног дела тренинг сета коришћени су скупови познатих корпуса за тренирање NER система. У Табели 7.1 је дат списак свих корпуса коришћених за формирање тренинг сета, одакле су сви анотирани ентитети који нису датум или време били одстрањени. Овако добијен тренинг сет је поред осталих речи 'O' садржао ентитете за датум B-DATE и I-DATE, као и независне ентитете за време B-TIME и I-TIME. Прво су сви датуми анотирани са B-DATE и I-DATE издвојени у чист скуп датума, одакле ће се даље формирати тренинг сет за модел за конверзију датума. Остале реченице се користе као један део тренинг скупа за NER систем. Подаци издвојени из наведених корпуса су већински на енглеском језику, па су самим тим за оба модела генерисани додатни формати датума представљени у Табели 6.1 на следећим језицима: енглески, руски, турски, шпански, румунски, француски, немачки, холандски, италијански, чешки, бугарски, пољски, шведски, кинески (поједностављен), кинески (традиционални), корејски и јапански.

Даље је, као што је већ објашњено у секцији 6.1.1, додатно преузет велики корпус садржаја постова са одабраних веб форума из Табеле 7.8, одакле су ручно издвојене реченице кратке дужине до 100 карактера, које су можда садржале датум, и које су потом ручно анотиране користећи Амазонов комерцијални сервис AMT [83]. Додатно се са истих веб форума ручно написаним скриптом покупио сав текст који садржи релативан датум, као и

околни HTML блок дужине 20 речи/симбола са леве и десне стране. Овако формиран текст је ручно анотиран и пречишћен користећи AMT.

У Табели 7.2 се може видети број добијених података коришћених за тренирање као и сетови који су коришћени за њихово формирање. Из јавно и комерцијално доступних корпуса (Табела 7.1) је за NER систем издвојено 200.000 података. Из ових података за модел за конверзију датума су издвојени само датуми који су коришћењем AMT сервиса ручним анотирањем добили своје еквиваленте у траженом излазном формату. Користећи *pyhton faker* библиотеку изгенерисано је додатних 300.000 датума за модел за конверзију датума, и 200.000 датума за NER модел. Овако генерисани датуми су били униформно распоређени по дефинисаним форматима из Табеле 6.1 као и по наведеним језицима. За генерисање HTML варијанти података, са веб форума је преузето 200.000 примерака, док је ручним анотирањем потврђено да само 85.112 има датум. Од укупног коначно добијеног броја података, 5% случајно изабраних се користило као валидациони сет, док се додатних 5% изабраних користило као тестни сет. Осталих 90% добијених података су коришћени за тренинг. У овом раду се валидациони сет користи како би се израчунале перформансе модела после сваке епохе, док се тестни сет користи само једном, на крају завршеног тренирања, како би се финално утврдили исти ови параметри на невиђеном сету података.

ТАБЕЛА 7.1
ПРЕГЛЕД КОРПУСА ПОДАТАКА КОРИШЋЕНИХ ЗА ФОРМИРАЊЕ ЈЕДНОГ ДЕЛА ТРЕНИНГ СЕТА NER СИСТЕМА

Корпус	Година	Извор текста	URL
MUC-6	1995	Wall Street Journal texts	https://catalog.ldc.upenn.edu/LDC2003T13
MUC-6 Plus	1995	Additional news to MUC-6	https://catalog.ldc.upenn.edu/LDC96T10
MUC-7	1997	New York Times news	https://catalog.ldc.upenn.edu/LDC2001T02
W-NUT	2015 - 2018	User-generated text	http://noisy-text.github.io
BBN	2005	Wall Street Journal texts	https://catalog.ldc.upenn.edu/ldc2005t33
OneNotes	2007 - 2012	Magazine, news, conversation, web	https://catalog.ldc.upenn.edu/LDC2013T19
WikiFiger	2012	Wikipedia	https://github.com/xiaoling/figer
DFKI	2018	Business news and social media	https://dfki-lt-re-group.bitbucket.io/product-corpus/

NER модел за детекцију ентитета датума је трениран тако да минимизује *cross-entropy* грешку са Adam [91] оптимизацијом, униформним фактором учења 0.001, величином серије од 64 података и 40% варијабилном *Dropout* [92] техником. Модел је трениран 7 епоха.

Модел за конверзију датума је трениран на само 4 епохе што се у експериментима показало као довољан број да модел научи да са великом сигурношћу конвертује улазну секвенцу у исправан датум. Модел је трениран тако да минимизује *categorical-crossentropy* [93] грешку са Adam оптимизацијом и фактором учења 0.006. Такође је коришћена техника *learning decay rate* [94] са фактором од 0.0009 која служи да после сваке епохе успори фактор учења како се минимум тражене функције не би прескочио.

Пошто је наведена рачунарска конфигурација за тренирање модела садржала поред четири процесора и специјализовану графичку карту за комплексне калкулације, за сваки модел је био покренут и тренинг на графичкој карти и исти тај тренинг на процесорима. Резултати времена тренирања се могу видети у Табели 7.3.

ТАБЕЛА 7.2
ПРЕГЛЕД КОЛИЧИНЕ И ИЗВОРА ПОДАТАКА КОРИШЋЕНИХ ЗА ГЕНЕРИСАЊЕ ТРЕНИНГ СЕТОВА

	Тренинг сет			Укупно података
	корпуси	Вештачки генерисани	HTML	
NER модел	200.000	200.000	85.112	485.112
Модел за конверзију датума	200.000	300.000	/	500.000

Време тренирања за оба модела се очекивано показало боље ако се користи графичка карта. У случају модела за конверзију датума, добијено убрзање је дупло, док је у случају NER модела добијено убрзање од чак 5 пута. Разлог боље искоришћености графичке карте од стране NER модела лежи у чињеници да је архитектура овог модела таква да се већи број паралелних калкулација може извршити од једном, што није случај са моделом за конверзију датума.

ТАБЕЛА 7.3
ВРЕМЕ ТРЕНИРАЊА У САТИМА ЗА ОБА МОДЕЛА КОРИСТЕЋИ ГРАФИЧКУ КАРТУ ИЛИ ПРОЦЕСОРЕ

	Време тренирања у сатима користећи графичку карту	Време тренирања у сатима користећи процесоре
NER модел	7,5	39
Модел за конверзију датума	4	8,5

7.1.2 Евалуативна метрика за тестирање модела

NER модел за детекцију ентитета датума је тестиран како се и обично тестирају NER системи у већини радова, а то је поређење излаза са очекиваним ручно анотираним сетом података. Поређење очекиваног анотираног резултата са излазом модела се у неким радовима може третирати као парцијално или комплетно поклапање. Иако су на MUC-6 [95] и ACE [96] конференцијама дефинисане релаксиране парцијалне евалуативне метрике, где су чак решени проблеми као што су парцијално поклапање погрешног типа и предложени под типови ентитета, у овом раду се ипак користи комплетно поклапање излаза са ручно анотираним подацима. Разлог је што иако су дефинисане метрике за парцијално поклапање добре, оне су и даље проблематичне, јер су компаративне само када су параметри фиксирани [64], [97], [98]. Комплексне евалуативне методе нису интуитивне и отежавају крајњу анализу грешака. Додатно, парцијалне евалуативне методе нису често коришћене у скоријим NER студијама [66]. Модел за конверзију датума из једног формата у други је такође евалуиран комплетним поклапањем излаза са већ предефинисаним тестним примерком.

Евалуативна метрика за оба модела се конципира на комплетном поклапању граница ентитета. Ентитет се сматра комплетно поклопљеним само ако се обе границе и тип у потпуности поклапају [99], [100]. Као евалуативна мера за модел за конверзију датума се узима само тачност, док се за NER систем узимају прецизност, сензитивности и F1-мера јер NER систем ради на нивоу класификатора тј. исправног препознавања, док модел за конверзију датума ради чисту предикцију излаза. Наведене мере се рачунају на основу броја тачно позитивних (TP – *true positives*), лажно позитивних (FP – *false positives*), лажно негативних (FN – *False Negatives*) и тачно негативних (TN – *True Negatives*) података.

- Тачно позитивни (TP): датуми који су комплетно тачно препознати од стране NER система или датуми који су тачно конвертовани од стране модела за конверзију датума.
- Лажно позитивни (FP): речи или токени који су препознати од стране NER система али нису тачни и не представљају праве датуме или датуми који нису тачно конвертовани од стране модела за конверзију датума.
- Лажно негативни (FN): ручно анотирани датуми који нису били препознати од стране NER система, тј. препознати су као нешто што није датум или су само парцијално били препознати.
- Тачно негативни (TN): остале речи или токени који нису датум, и који су исправно били препознати као такви од стране NER система.

Прецизност мери могућност NER система да предвиди искључиво тачне датуме, док сензитивност мери могућност NER система да детектује све датуме који се налазе у задатом сету:

$$Прецизност = \frac{TP}{TP + FP} \tag{7.1}$$

$$Сензитивност = \frac{TP}{TP + FN} \tag{7.2}$$

F1-мера се рачуна као хармонијска средина прецизности и сензитивности:

$$F1 - мера = 2\frac{прецизност * сензитивност}{прецизност + сензитивност} \tag{7.3}$$

Иако предложен NER модел врши детекцију две врсте ентитета, да би израчунали F1-меру, NER модел се третира као да врши детекцију искључиво једног ентитета – датума. Иако је у даљем тексту извршена евалуација перформанси ентитета времена, овде ипак нису мерене макро-просечне и микро-просечне F1-мере које се обично користе код NER система који врше детекцију више различитих ентитета. Разлог је једноставност ентитета времена чији су формати коначно дефинисани. За модел за конверзију датума се уместо прецизности и сензитивности прилаже мера тачности, чије мерење има више смисла:

$$тачност = \frac{TN + TP}{TN + TP + FN + FP} \tag{7.4}$$

Тачност представља пропорцију тачних резултата, и мери степен истинитости теста. С обзиром на то да се лажно негативни и тачно негативни узорци не могу мерити од стране

модела за конверзију датума, јер то није класификациони модел и има само тачно или нетачно конвертован датум, ове две мере узимају вредност 0 приликом рачунања тачности.

Перформансе оба модела су приказане у Табели 7.4 и Табели 7.5. За оба модела су приказане перформансе измерене на тренинг и тест сету. За потребе NER система су мерени прецизност и сензитивност, док је за потребе модела за конверзију датума измерена само тачност. Све три мере NER модела су посебно извршене за ентитет датума а посебно за ентитет времена.

ТАБЕЛА 7.4
ПРЕГЛЕД ПЕРФОРМАНСИ NER МОДЕЛА НА ТЕСТ И ТРЕНИНГ СКУПУ

NER модел	Ентитет датума			Ентитет времена		
	Прецизност	сензитивност	F1-мера	Прецизност	сензитивност	F1-мера
Тест сет	0.99	0.98	0.988	0.99	0.965	0.979
Тренинг сет	0.99	0.99	0.99	0.99	0.99	0.99

NER систем је показао високу прецизност и сензитивност на оба евалуативна сета, што је резултовало високо задовољавајућом F1-мером за оба ентитета. Иако се користио мали број тренинг епоха, ово је било довољно да модел јако брзо дође да изузетно високих перформанси. 40% варијабилна *Dropout* техника је спречавала да модел преучи и да се превише прилагоди тренинг сету, што се може видети на перформансама тестног сета, који је скуп примера који модел до тада никада пре није видео. Због великог број тренинг примерака, без коришћења *Dropout* технике, модел може да постигне сличне перформансе после само три епохе, али то може да резултује јако лошим перформансама на тестном скупу.

ТАБЕЛА 7.5
ПРЕГЛЕД ПЕРФОРМАНСИ ТАЧНОСТИ МОДЕЛА ЗА ЕКСТРАКЦИЈУ ЕНТИТЕТА ДАТУМА

	Тренинг сет	Тест сет
Модел за конверзију датума	0.99	0.984

Мерења за модел за конверзију датума такође показују изузетно високу тачност приликом конвертовања улазне секвенце у датум, што се може видети по резултату перформанси тестног сета података. Приликом одабира примерака тестног сета, пазило се да се међу одабраним примерима нађе равномерна дистрибуција свих невиђених формата у равномерној дистрибуцији подржаних језика. На тај начин је могуће измерити перформансе модела подједнако за све случајеве који се могу појавити а да их опет модел пре тога није видео.

За потребе тренирања овог модела није коришћена *Dropout* техника, јер због великог броја тренинг примерака, као и њихове разноликости, модел није преучио на наведеном број епоха на које је трениран.

7.1.3 Поређење NER система са постојећим готовим решењима

У Табели 7.6 су дати постојећи NER системи који су већ у напред истренирани на великом броју тренинг примерака за детекцију различитих врста ентитета [66]. Неки од ових *off-the-shelf* готових система су упоређени са предложеним NER системом користећи дефинисане метрике прецизности, сензитивности и F1-мере, док је коришћен сет за евалуацију био већ унапред формиран тестни сет. Мерења су урађена независно за оба ентитета датума и времена.

ТАБЕЛА 7.6
ПРЕГЛЕД ЈАВНО ДОСТУПНИХ NER СИСТЕМА И ПАКЕТА [66]

Име система	URL	Тестиран
Stanford Core NLP	https://stanfordnlp.github.io/CoreNLP/	+
OSU Twitter NLP	https://github.com/aritter/twitter_nlp	-
Neuro NER	http://neuroner.com/	-
NER suite	http://nersuite.nlplab.org/	-
Polyglot	https://polyglot.readthedocs.io	-
Gimli	http://bioinformatics.ua.pt/gimli	-
spaCy	https://spacy.io/	+
NLTK	https://www.nltk.org	-
Apache Open NLP	https://opennlp.apache.org/	+
LingPipe	http://alias-i.com/lingpipe-3.9.3/	-
IBM Watson	https://www.ibm.com/watson/	-

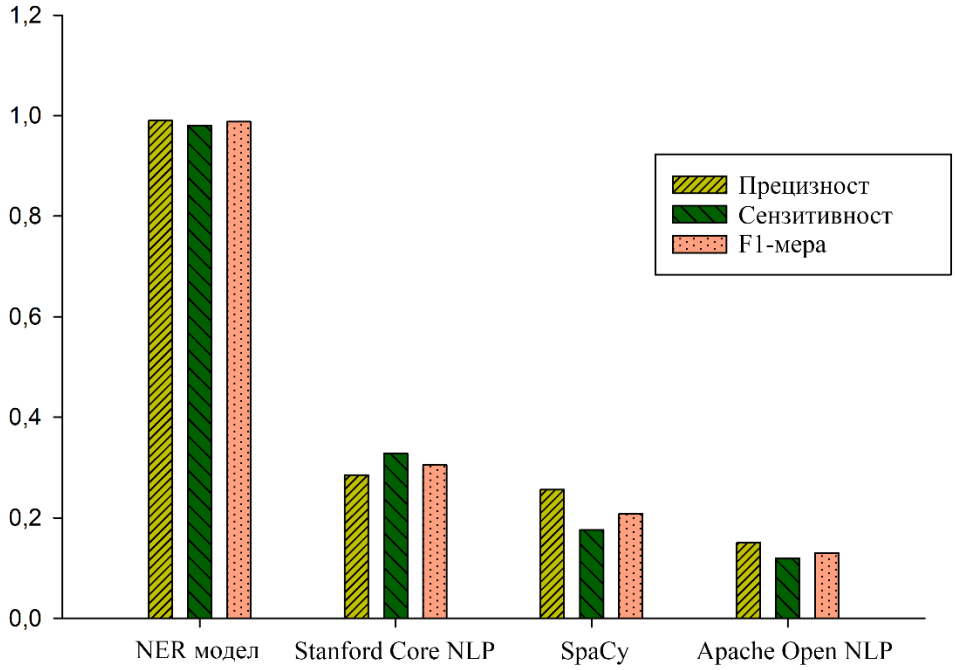
За сваки од наведених NER система из Табеле 7.6 су издвојене кратке напомене, опис система и разлог зашто није тестиран ако је то био случај.

- *Stanford Core NLP* је систем који је показао најбоље резултате од свих представљених у Табели 7.6 за екстракцију времена као ентитета. Иако није постигао резултате као предложени NER систем из овог рада, показао се као систем који може да препозна изузетно велики спектар формата времена. Што се тиче екстракције датума, лош резултат је продукт чињенице да овај систем раздваја реченице користећи тачку као терминални симбол краја реченице, која код датума може бити интегрални део формата записа. Овако разбијени датуми у делове не могу бити исправно детектовани. Додатно, систем препознаје као датум доста парцијалних ентитета као што су *Sep, once, sunday, current* и сл. што доводи до грешке.
- *OSU Twitter NLP* систем, поред разних ентитета које подржава са јако добрим перформансама, ентитете датума и времена не подржава, тако да његова евалуација није узета у обзир. Са друге стране, овај систем подржаве 10 других ентитета *Person, Geo-Loc, Company, Facility, Product, Band, Sportsteam, Movie, Tv-Show* и *Other*.
- *NeuroNER* систем не подржава ентитете датума и времена. Подржава екстракцију само четири врсте ентитета: *Organization, Person, Geo-Loc* и *Misc*.
- *NER suite, LingPipe* и *Gimli* системи су специјализовани за екстракцију ентитета везаних за биологију, па самим тим не подржавају ентитете датума и времена.
- *Polyglot* подржава само три ентитета међу којима нису датум и време: *Organization, Person* и *Geo-Loc*.
- *SpaCy* подржава екстракцију и датума и времена. Слично као и са *Stanford Core NLP* системом, детекција времена је била боља у смислу прецизности и сензитивности од детекције датума, јер формата времена има коначно мање и

формати датума могу бити неисправно протумачени због симбола које могу садржати.

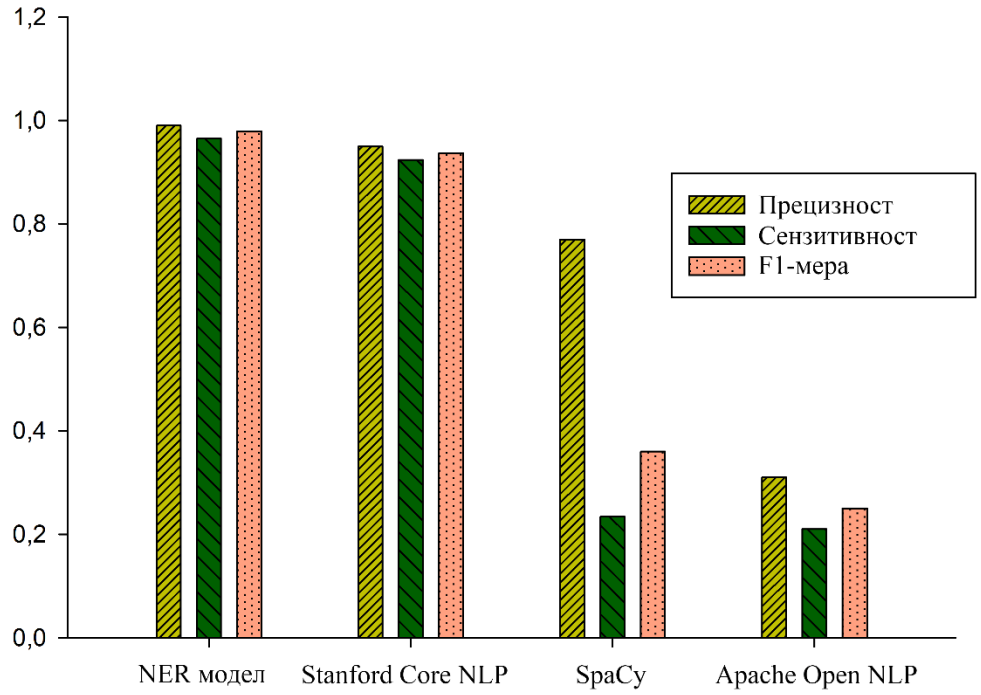
- *NLTK* сам по себи не може да детектује ентитете датума и времена, него користи *Stanford's Named Entity Tagger* библиотеку, уз помоћ које може да изврши детекцију. С обзиром да се ова библиотека такође користи и у *Stanford Core NLP* систему, *NLTK* није тестиран у овој евалуацији.
- *Apache Open NLP* је показао доста лошије резултате од NER система предложеног у овом раду. Као прво, у највећем броју случајева ентитет времена је детектован као датум, што доводи до лошијих резултата мерења за оба ентитета. Додатно, делови датума се некада парцијално детектују а остатак занемарује, на пример, од целог датума може да се деси да је само година детектована, док остали делови датума нису.
- *IBM Watson* као и већина описаних система, не подржава детекцију датума и времена као ентитета.

Као што се може видети на Слици 7.1 NER систем предложен у овом раду има боље перформансе од свих тестираних модела када је у питању екстракција датума са прецизношћу, сензитивношћу и F1-мером од 99%, 98% и 98.8% респективно.



Слика 7.1 – Перформансе детекције датума

Stanford Core NLP, *SpaCy* и *Apache Open NLP* NER системи имају доста лошије резултате, где је највећу F1-меру имао *Stanford Core NLP*. NER модел предложен у овом раду је имао боље перформансе за 4 и више пута од осталих наведених система.



Слика 7.2 – Перформансе детекције времена

Што се тиче резултата екстракције ентитета времена, NER систем представљен у овом раду је показао најбоље перформансе где су прецизност, сензитивност и F1-мера били 99%, 96.5% и 97.9% респективно – Слика 7.2. По перформансама на другом месту се издваја *Stanford Core NLP* који прати предложени NER систем са F1-мером од 93.7%. Иако *SpaCy* и *Apache Open NLP* NER системи имају боље перформансе детекције времена него датума и даље сам резултат није добар да би систем могао да се користи приликом детекције времена. Додатно, очекивано је било да наведени системи покажу боље перформансе на детекцији времена него датума, јер се формат времена увек појављује у коначном броју комбинација и није зависан од језика као што је то случај са датумима.

7.1.4 Перформансе извршавања

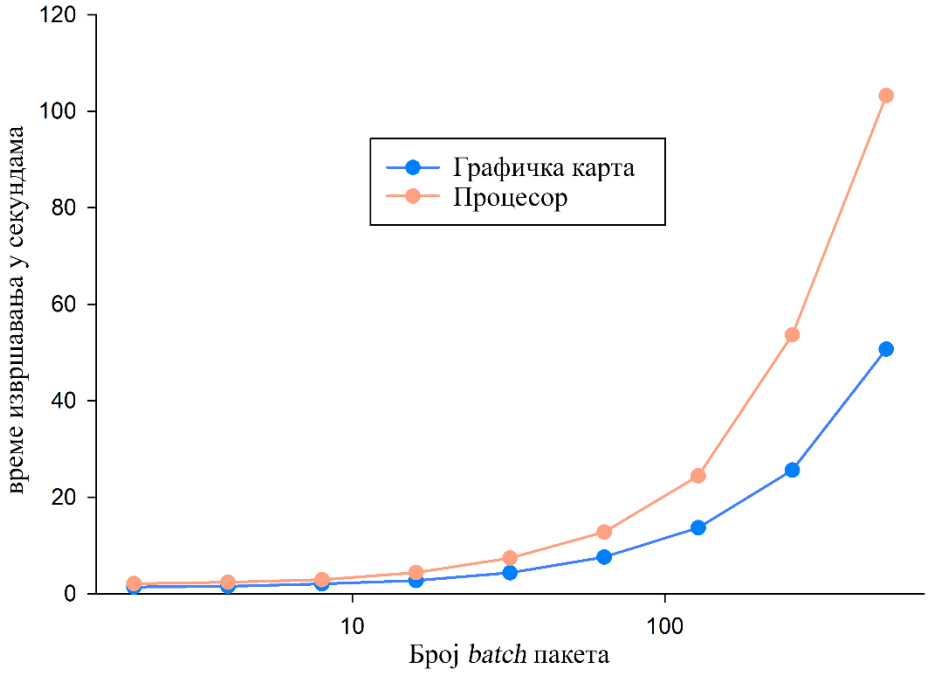
Скоро сви модели машинског учења имају могућност паралелног рачунања улазних података, што важи и за моделе представљене у овом раду. Уместо само једног датума који треба конвертовати или пронаћи у реченици, могуће је процесирати више њих од једном. Ова техника се зове *batching* и може да доведе до знатног убрзања броја обрађених података у јединици времена. Перформансе извршавања су битна мера, јер мора постојати праг колико модели могу да конвертују или детектују датума у датој јединици времена. Ово је битан параметар из разлога што се може десити да претраживач много брже проналази датуме које је потребно послати на обраду него што то модели могу да подрже, чиме може доћи до преоптерећења система, тј. успоравања претраживања. Саме перформансе извршавања евалуиране су у две варијанте, са по једним позивом и са више њих (*batching*), где је свака варијанта засебно тестирана на графичкој карти и процесорима. Перформансе извршавања појединачног позива по минути су дате у Табели 7.7.

ТАБЕЛА 7.7
БРОЈ ИЗВРШАВАЊА У МИНУТИМА ОБА МОДЕЛА НА ГРАФИЧКОЈ КАРТИ И ПРОЦЕСОРИМА ПРИЛИКОМ ПОЈЕДИНАЧНОГ ПРОСЛЕЂИВАЊА ПОДАТКА

	Тип извршавања		
	Процесор	Графичка карта	Фактор убрзања
NER модел	200	520	2.60 x
Модел за конверзију датума	285	375	1.31 x

Мерења су изведена на рачунару који покреће *Intel Core i7* и који садржи 32 GB RAM меморије са графичком картом *nVidia Tesla K80*. Као што се може видети из претходне табеле, NER модел постиже много веће убрзање извршавајући се на графичкој карти него модел за конверзију датума. Са друге стране, извршавање у *batch* моду и одговарајуће време извршавања се може видети на Слици 7.3 и Слици 7.4 респективно за оба модела.

За тест је узето 1024 податка, који су подељени у *batch* пакете величине 2, 4, 8, 16, 32, 64, 128, 256 и 512 података, што значи да ће број *batch* пакета за те величине бити 512, 256, 128, 64, 32, 16, 8, 4 и 2 респективно. Као што се може видети на представљеном графику, перформансе тј. брзина извршавања се повећава како се величина *batch* пакета повећава за исти број датума које треба конвертовати. 1024 датума је много брже конвертовати ако се користе два *batch* пакета од по 512 датума него 512 пакета од по два датума јер ће модел сваки *batch* пакет да процесира од једном.



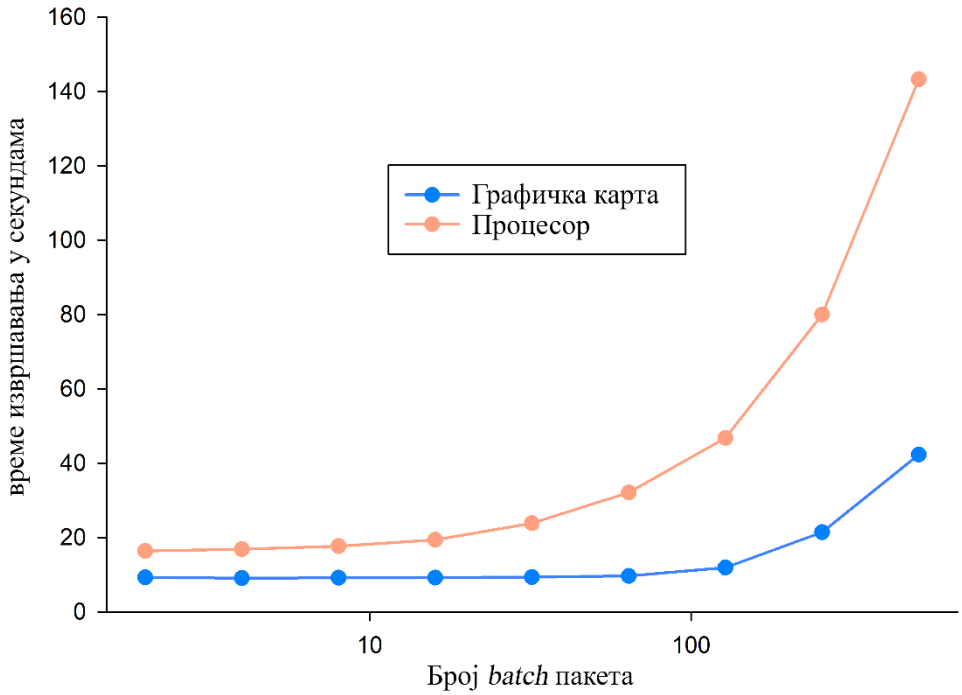
Слика 7.3 – Перформансе модела за конверзију датума приликом паралелног процесирања података у batch моду

Ефекти убрзања модела за конверзију датума код веће величине *batch* пакета се могу видети и на процесору и на графичкој карти. Даље, модел постиже много боље перформансе на графичкој карти него на процесору када је у питању мања величина *batch* пакета. За 512 *batch* пакета, процесорско време извршавања је било 103 секунде, док је исти процес на

графичкој карти трајао 50 секунди. Како се величина *batch* пакета повећава, тако се ове перформансе изједначавају за исти број датума, тако да за два *batch* пакета од по 512 датума, потребно је и на процесорима и на графичкој карти 2 секунде за извршавање. Ово доводи до закључка да је свеједно који хардверски ресурс ће модел за конверзију датума да користи за извршавање докле год је *batch* пакет довољно велики. На Слици 7.4 се може видети време извршавања за NER модел.

Резултати су слични као и за модел за конверзију датума, са разликом да овај модел константо постиже боље резултате извршавања на графичкој карти чак и за већу величину *batch* пакета. Док је за 512 *batch* пакета, процесорско време извршавања било 143 секунде, на графичкој карти је за исти број *batch* пакета било потребно 42 секунде, што је убрзање од 3,4 пута. Када се број *batch* пакета смањи на два, а величина пакета износи 512 података, време извршавања на процесору је 17 секунди, док је не графичкој карти 9 секунди, што и даље преставља убрзање од 1.9 пута.

Како ова два модела приликом претраживања у реалном времену морају паралелно да се извршавају, одлучено је да се моделу за конверзију датума као хардверски ресурс додели процесор, док је као хардверски ресурс за NER модел одређена графичка карта. Са оваквом расподелом добила се боља искоришћеност хардверских ресурса, као и ефикасније и брже време извршавања, чије се перформансе могу видети у следећем поглављу.



Слика 7.4 – Перформансе NER модела приликом паралелног процесирања података у batch начину извршавања

У експерименталним резултатима се показало да већа величина *batch* пакета са мањим бројем захтева за обраду смањује време извршавања због паралелног израчунавања, па је самим тим систем прилагођен *batching* начину извршавања. Да би се постигао *batching* начин извршавања, направљен је бафер, тј. ред од 512 слотова за датуме, који се прво попуњавао пре него што је слат на конверзију.

7.2 Експериментална поставка и коришћени подаци за тестирање претраживача

Експерименти су изведени на рачунару који покреће *Intel Core i7* и који садржи 32 GB RAM меморије са инсталираном графичком картом *nVidia Tesla K80*. Табела 7.8 приказује скуп података који је коришћен у наведеним експериментима. Овај скуп података укључује 14 веб форума, пажљиво одабраних да покрију све неопходне случајеве за евалуацију алгоритама као и са укљученим свим репрезентативним форумским технологијама потребним за смислену евалуацију.

Покривено је пет најпопуларнијих веб форумских технологија: phpBB, vBulletin, Discuz, SMF и XenForo, које су у време писања овог рада покривале око 82,5% свих коришћених софтверских пакета веб форума [9]. Такође су укључена и индивидуална форумска софтверска решења како би се проширили тестови и обухватили нестандардни случајеви потребни за тестирање. Да би се правилно спровела евалуација, свих 14 одабраних веб форума је најпре пресликано у локалну базу података. Да би се прикупили сви подаци са одабраних веб форума, за сваки форум, због њихове јединствености технологије и тематског приказа, морали су бити направљени уникатни прилагођени претраживачи у облику скрипти. Сви претраживачи су индексирали форуме по ширини, неограниченој дубини и били су лимитирани на конкретни домен. Затим су за сваку софтверску технологију (за стандардне форумске технологије као и индивидуална решења) изграђени симулатори који генеришу садржај као и оригиналне форумске технологије.

Сваки симулатор има могућност да одговори на URL захтев који шаље SInFo претраживач и да на основу тог захтева генерише садржај и изглед имитирајући оригинални веб форум. Да би у овој евалуацији било могуће подржати инкрементално претраживање, симулатори су такође могли да репродукују садржај форума онакав какав је био у одређеним временским тренуцима у прошлости. Симулатор користи временски интервал за генерисање стања веб форума који одговара оригиналном стању форума у том периоду, а које касније може да се претражује од стране SInFo претраживача.

7.3 Евалуација ефикасности модула за детекцију типа сортирања

Да би се проценила ефикасност модула за детекцију сортирања индекс страна и детекцију сортирања дискусија, узето је 100 насумичних страна од оба типа са сваког одабраног веб форума, које су касније пропуштене кроз модуле. Добијени резултати детекције су прегледани ручно и представљени у Табели 7.9.

Већина одабраних веб форумских технологија има датуме записане у релативном формату (нпр. пре 3 месеца), и сви одговарајући прецизни датуми који постоје у атрибутима HTML тагова су исправно препознати од стране модула осим на форуму tripadvisor.com и на индекс страни yahoo.com форума. Иако ова два веб форума садрже релативне датуме на постовима с прецизним форматом унутар HTML таг атрибута, они не примењују исту прецизност на индекс страни, што може довести до погрешног откривања редоследа сортирања. Што се тиче iiyi.com форума, три од свих насумично изабраних страна садрже више од 40 изабраних (*sticky*) дискусија од укупно 50 колико се приказује по страни, а где је 50 максимални број дискусија на једној индекс страни. Будући да је већина ових датума представљала микс без константног опадајућег или растућег редоследа, IPSD модул није могао правилно да детектује тип редоследа за сортирање. Каснија инспекција ових страна је показала

да је број изабраних (*sticky*) дискусија ажуриран на мање од 10, што имплицира на погрешно кратко стање ове три индекс стране на iiyi.com.

ТАБЕЛА 7.8
ПРЕГЛЕД ФОРУМА КОРИШЋЕНИХ ЗА ТЕСТИРАЊЕ. LAD – СОРТИРАНО ПО ДАТУМУ ПОСЛЕДЊЕ АКТИВНОСТИ. CRD - СОРТИРАНО ПО ДАТУМУ КРЕИРАЊА. ASC – РАСТУЋЕ. DSC - ОПАДАЈУЋЕ

Бр.	Адреса форума	Технологија форума	Сортирање на индексној страни (приказан датум)	Сортирање постова	Пагинација на индексној страни	URL последње активности
1	http://forums.bigfishgames.com	phpBB	LAD	ASC	+	+
2	http://www.airliners.net/forum/	phpBB	LAD	ASC	+	+
3	http://bbs.chinadaily.com.cn/forum.php	Discuz!	LAD	ASC	+	+
4	http://bbs.iiyi.com/	Discuz!	LAD	ASC	+	+
5	http://www.doityourself.com/forum/	vBulletin	LAD	ASC	+	+
6	https://forum.avast.com/	SMF	LAD	ASC	+	+
7	https://forums.macrumors.com/	XenForo	LAD	ASC	+	+
8	https://forum.parallels.com/	XenForo	LAD	ASC	+	+
9	https://www.tripadvisor.com/ForumHome	custom	LAD	ASC	+	+
10	https://complaintwire.org	custom	CRD	-	-	+
11	https://answers.yahoo.com	custom	LAD (CRD)	DSC	-	-
12	https://www.generation-nt.com/entraide.html	custom	CRD	ASC	-	-
13	http://club.beaute-addict.com/forum/	custom	LAD	ASC	+	-
14	https://www.afb.org/messageboards.aspx	custom	CRD	DSC	+	-

Што се тиче модула за откривање типа сортирања на дискусијама, најлошија прецизност постигнута је на форуму afb.org, што се догодило због малог броја постова по дискусији. Већина одабраних дискусија са овог форума имала је само један или два поста. Прецизност на complaintwire.org је била 82% због саме технологије форума која омогућава одговор у виду коментара директно на пост, што као ефекат даје угњеждену HTML структуру коментара унутар поста. Ово доводи до неправилног делимичног поравнања HTML стабала и погрешног откривања редоследа сортирања од стране модула. Што се тиче форума airliners.net, 4 насумично одабране дискусије су имале само један пост. Мали број постова по дискусији такође је пронађен у примерима форума под редним бројевима 5, 6 и 12 у Табели 7.8.

ТАБЕЛА 7.9
ЕВАЛУАЦИЈА ПРЕЦИЗНОСТИ МОДУЛА ЗА ДЕТЕКЦИЈУ РЕДОСЛЕДА СОРТИРАЊА НА ИНДЕКСНИМ И ДИСКУСИОНИМ СТРАНАМА

Форум	Bigfishgames.com	airliners.net	chinadaily.com.cn	iiyi.com	doityourself.com	avast.com	macrumors.com	parallels.com	tripadvisor.com	complaintwire.org	yahoo.com	generation-nt.com	beaute-addict.com	afb.org
IPSD	100%	100%	100%	97%	100%	100%	100%	100%	92%	100%	89%	100%	100%	100%
TSD	100%	96%	100%	100%	99%	99%	100%	100%	100%	82%	100%	97%	100%	76%

Укупна прецизност детекције редоследа сортирања на индексним странама је износила 98,4%, док је укупна прецизност детекције редоследа сортирања на дискусионим странама 96,4%, што говори о робусности и високој отпорности целокупног система на разноликост презентација у односу на технологије форума представљене у оквиру скупа података.

7.4 Евалуација претраживача

Ова студија је специфична по томе што процењује циљање најновије генерисаног садржаја у инкременталном претраживању на веб форумима. У наредној секцији ће бити приказан покушај целовите и објективне процене предложеног система, имајући у виду да постојећи специјализовани претраживачи форума нису дизајнирани као инкрементални или нису добро документовани да би се могли симулирати слични приступи циљању најновијег садржаја у инкременталним циклусима претраживања.

Дизајн предложеног и тестираног дела система за претраживање и индексирање SInFo одговара периодичном претраживачу са фиксном фреквенцијом поновног претраживања. Иако континуирани претраживачи са променљивом фреквенцијом могу повећати свежину садржаја [101], у овом раду је коришћен периодичан начин рада због једноставности евалуације и јер се у даљим експериментима не евалуира свежина података. Садржај на веб форумима се ретко мења или брише, али се стално додаје, што значи да једном прикупљени садржај са веб форума нема потребе за освежавањем у регуларним случајевима.

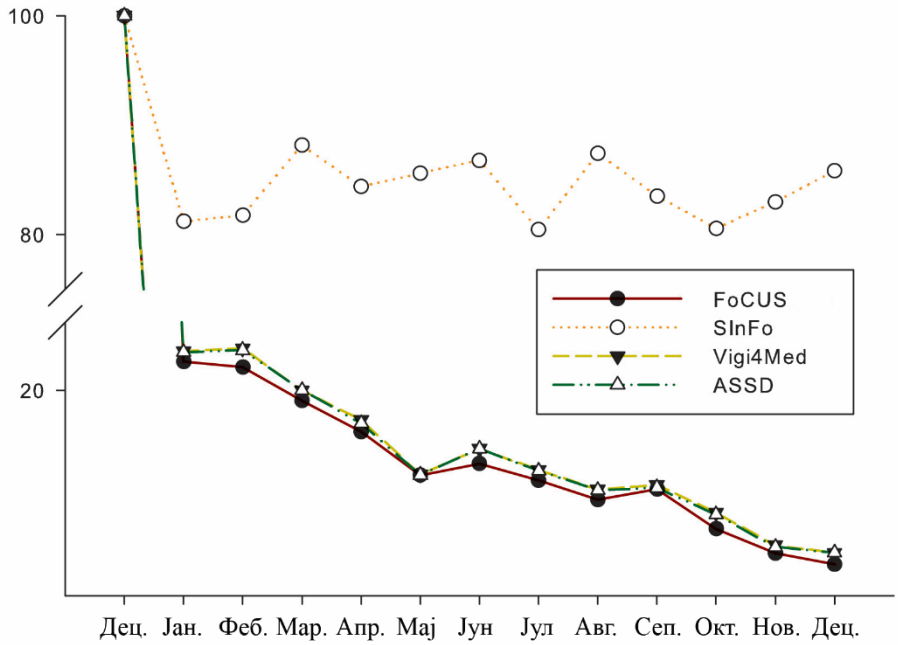
Процена комплетног претраживања великих размера је извршена током једногодишњег временског периода 2016 године. Почев од 1. јануара, сваког првог дана у месецу, извршено је једно инкрементално претраживање, до укупно 12 претраживања. У сваком кораку појединачног претраживања систем је прикупио све нове генерисане податке од датума последњег циклуса претраживања. Никаква тежина нити приоритет није дат при преузимању веб страна, а систем за претраживање је преузимао странице у петљи, све док није било више страна са новим садржајем за преузимање. Претраживач је такође постављен за иницијалну фазу претраживања 1. децембра 2015, када је покренут ради прикупљања свих података генерисаних на форуму од почетка рада форума до тог датума. На сваком симулираном месечном циклусу претраживања, свих 14 веб форума је индексирано, а њихов прикупљени садржај снимљен је у базу података.

7.4.1 Евалуација дупликата

Циљ ове евалуације је био да се открије колико је дупликата систем сакупио у сваком циклусу претраживања. Уместо да се посматрају само дупликати URL-ова, посматрају се стране са дуплираним садржајем. За садржај, поред URL-ова дискусија са индекс стране, такође се подразумевају и постови на дискусијама. Разлог је дистрибуција садржаја на више страна повезаних URL-овима за пагинацију. Будући да генерисање новог садржаја може „гурнути“ постојећи садржај на следеће у низу сукцесивне стране, посматрање само URL-а стране која садржи тај садржај било би погрешно, јер страна са истим URL-ом у различитим циклусима претраживања може имати различит садржај.

Да би се показале потешкоће у инкременталном претраживању, у даљем излагању ће предложени систем за претраживање бити упоређен са специјализованим претраживачима форума налик FoCUS-у, Vigi4Med-у и претраживачима представљеним у [35] (ASSD) - Слика 7.5. Иако CrimeBot [36] подржава инкрементално претраживање, аутори овог решења нису јавно публиковали технички опис и детаље потребне за репликацију и процену. Систем предложен у [37] је превише прилагођен vBulletin технологији. У овом раду је изабрана FoCUS архитектура пре него iRobot, јер је показала боље перформансе [3] у претраживању форума. Имплементација претраживача сличних FoCUS-у и ASSD-у изведена је као што је описано у [3] и [34]. Vigi4Med имплементација је доступна на GitHub репозиторијуму (https://github.com/bissana/Vigi4Med-Scraper), и за потребе ове евалуације по једна конфигурациона датотека је написана за сваки форум представљен у Табели 7.8. Да би

упоређивање било поштено, свим претраживачима је додата могућност препознавања и тумачења *sitemap* протокола. Ово је омогућило претраживачима да разликују старе, нове и ажуриране URL адресе на веб форумима, где је датотека *sitemap* правилно одржавана. За сваки месец дат је проценат прикупљеног новог садржаја у поређењу са свим раније сакупљеним садржајима из претходних месеци који су садржани у бази података.



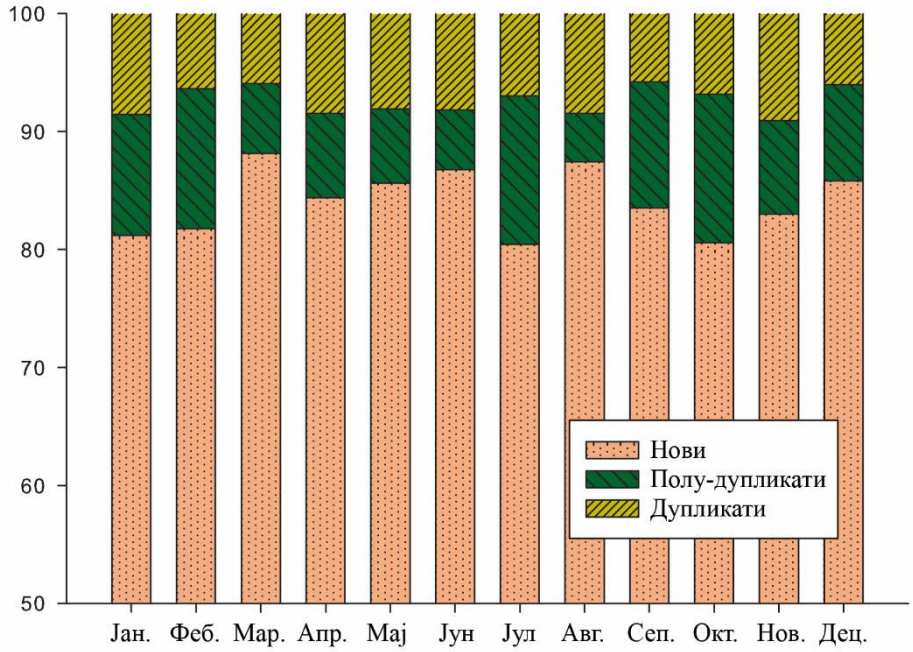
Слика 7.5 – Однос новог садржаја прикупљеног по месецима 2016 године за FoCUS, SInFo, Vigi4Med и ASSD претраживаче

У децембру 2015, са инцијалним претраживањем, проценат за све претраживаче је био 100%, јер је прикупљан искључиво нови садржај. Након иницијалног претраживања, у јануару 2016. проценат новог садржаја драстично опада за све претраживаче, осим за SInFo. Овакав резултат је био очекиван јер су остали претраживачи прикупили готово све што је већ било сакупљено у инцијалном претраживању. Проценат новог садржаја је наставио да опада за свако узастопно претраживање, са минималним повећањем у јуну и септембру. Ово повећање је углавном последица велике количине новог садржаја који се појавио током тих месеци, посебно на веб форумима generation-nt.com и forums.macrumors.com које правилно одржавају своју *sitemap* датотеку за део URL-ова који се тичу секције форума. Системи за претраживање Vigi4Med и ASSD показали су сличне резултате у односу новог и старог садржаја, јер оба система базирају своје путање кроз форум на XPath упитима. Разлика је у чињеници да ASSD претраживач генерише мање прецизне XPath упите на основу датих семантичких правила, док се за Vigi4Med ти исти упити ручно пишу и контролишу. Поред тога, оба система за претраживање су прикупила више новог садржаја од претраживача налик FoCUS-у, где је ово последица FoCUS-овог аутоматског генерисања регуларних израза за сваки веб форум засебно, што доводи до мање прецизности, али готово без било какве интервенције корисника.

SInFo је са друге стране показао високу тачност и низак ниво дупликата у сваком новом циклусу претраживања. Активност форума и дистрибуција новог садржаја по странама утичу на циљање најновијег садржаја, али проценат потпуно новог садржаја који је прикупљен у сваком циклусу претраживања је био стабилан, достижући максималних 88% и где ни једном није пао испод 80%. Већина дупликата које је SInFo прикупио налазили су се на странама са старим садржајем, које је требало посетити само да би се пронашли навигациони URL-ови, или да би се одредило када да се заустави претраживање тренутне дискусије или индекса. Неке

стране су такође имале мешавину старог и новог садржаја на дискусионој или индекс страни. Током евалуације ови полу-дупликати су рачунати као комплетни дупликати.

Да би се разликовале различите врсте дупликата које је прикупио претраживач SInFo, за сваки месец израчунат је проценат новог садржаја, полу-дупликата и дупликата, при чему се под садржајем мисли на садржај једне комплетне стране. То значи да се садржај сматра комплетно новим, само на страни на којој не постоји никакав стари садржај. Полу-дупликатима се означава садржај који се налази на странама на којима постоји мешавина новог и старог садржаја из претходних претраживања. На индекс страни се то може догодити са било којом врстом редоследа сортирања. Када се сортира према датуму последње активности, на истој страни може се појавити мешавина нових и старих, али ажурираних URL-ова дискусија. У случају редоследа сортирања према датуму започињања дискусије, страна на којој престају нови URL-ови дискусија и почињу стари сматра се полу-дупликатом. На странама дискусија, слични сценарији се примећују и са постовима, тј. страна која садржи и старе и нове постове сматра се страном са полу-дупликатима, без обзира на редослед сортирања на тој дискусији. Будући да су стране са полу-дупликатима недељива целина приликом претраживања, претраживач свакако мора да их посети да би пронашао нови садржај. Стране са потпуним дупликатима приказују се одвојено (Слика 7.6).



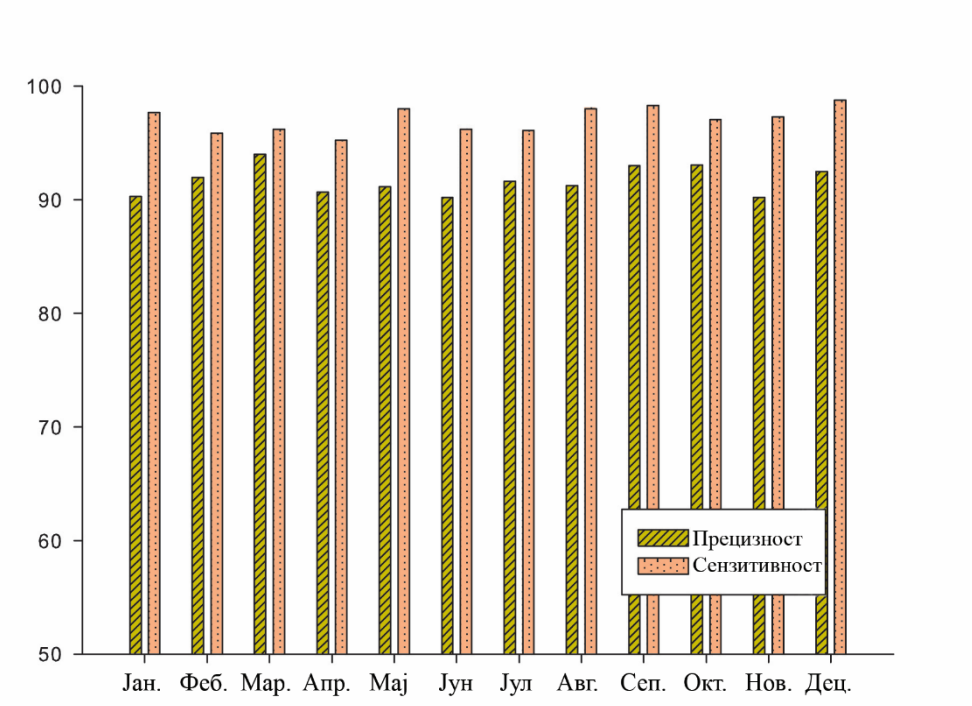
Слика 7.6 – Однос нових, полу-дупликата и дупликата откривених по месецима за време претраживања целе 2016 године

Под комплетним дуплим садржајем сматра се онај садржај на страни на којој се не могу пронаћи нови URL-ови дискусија или постови. Страна са комплетно дуплим садржајем може се преузети грешком или због сувишног али неопходног захтева. Сувишан али неопходан захтев значи да претраживач преузима индекс или дискусиону страну да би затим пронашао стари садржај, што је сигнал претраживачу да треба да се заустави. У неким случајевима се стари садржај може дистрибуирати преко целе стране без икаквог новог садржаја, јер претходна страна није садржала никакав стари садржај. Преузимање ових страна ја неопходно да би претраживач одредио када треба да се заустави. Поред тога, када се прикупљају постови на дискусији са растућим сортирањем, а ни један URL последње активности дискусије није

доступан на индекс страни, претраживач мора да посети прву страну да би пронашао само URL последње активности те дискусије, иако је та страна потпуни дупликат - Слика 6.21в. Резултати су приказани на Слици 7.6. За сваки месец представљен је укупни проценат новог садржаја, полу-дупликата и дупликата за свих 14 форума из скупа података.

Слика 7.6 показује задовољавајуће резултате у погледу малог броја дупликата. Полу-дупликати се сматрају неопходним и строго зависе од делимичне дистрибуције постова и URL-ова дискусија на њиховим странама. Већина полу-дупликата појавила се у октобру и јулу и достигла је 12,5%, када су се чешће појављивале стране са делимично помешаним старим и новим садржајем. Иако је стари садржај прескочен на мешовитим странама, претраживач је и даље морао да преузме читаву страну, чиме се троши време и оптерећује пропусни опсег.

Са друге стране, већина дупликата појавила се у новембру и достигла је 9%. То је последица дистрибуције новог садржаја на странама које су нарочито биле доминантне на iiyi.com и chinadaily.com.cn, где је систем за претраживање морао да преузме барем још једну додатну страну да би стопирао претраживање и где су те додатне стране биле комплетни дупликати.



Слика 7.7 – Прецизност и сензитивност по месецима за период од једне године инкременталног претраживања

Укупни дупликати у једногодишњем циклусу претраживања износе 7,4%, док су полу-дупликати заузели 8,54% целокупног садржаја. SInFo претраживач је прикупио 84% потпуно новог садржаја са додатних 8,54% садржаја за које је било потребно филтрирање и одвајање старих и нових постова. Ово је генерисало укупно 92,6% корисног садржаја који није прикупљен у претходним претраживањима.

7.4.2 Процена прецизности и сензитивности.

Генерално за веб сајтове постоје три битне метрике за инкрементално претраживање, а то су свежина, покривеност и старост података. Свежина се односи на податке у бази који су већ индексирани, а старост на податак који се појавио на сајту а још није индексиран. С обзиром да већ индексиран и раније преузет садржај форума нема потребу за освежавањем, и

будући да је фокус овог рада на инкременталном претраживању новонасталог садржаја а не на процени периода претраживања, свежина и старост нису узимани у даље разматрање. У овом случају покривеност се мери сензитивношћу, док прецизност представља ефикасност предложеног претраживача. Пошто су форуми првобитно комплетно били претражени и пресликани у базу података, сав садржај који треба циљати у инкременталном претраживању је већ познат, па се прецизност и сензитивност могу правилно израчунати. Прецизност и сензитивност претраживача се рачунају за сваки месец посебно, слично као и код евалуације дупликата и полу-дупликата. У претходном тесту нису разматране стране које имају на себи садржај који није релевантан за претраживање, тј. погрешне стране без постова и URL-ова дискусија. У следећој евалуацији се узимају у обзир ове погрешне стране и дефинише се прецизност на следећи начин:

$$precision = \frac{C_{new} + C_{semiduplicate}}{C_{all}} 100\% \tag{7.5}$$

где C_{all} представља сав сакупљени садржај укључујући нов, полу-дупликате, дупликате и стране са грешком, и где $C_{semiduplicate}$ и C_{new} означавају број страна полу-дупликата и новог садржаја покупљеног приликом претраживања. Сензитивност је дефинисана на сличан начин:

$$recall = \frac{C_{new} + C_{semiduplicates}}{I_{allrelevant}} 100\% \tag{7.6}$$

где је $I_{allrelevant}$ укупан број страна са целокупним релевантним садржајем, генерисаног на веб страни од датума последњег претраживања, који треба покупити у тренутном циклусу инкременталног претраживања. Резултати су приказани на Слици 7.7. Прецизност је мања од сензитивности у сваком месецу, пошто ова мера укључује стране које не садрже URL адресе дискусија нити постове, па самим тим нису ни релевантне за претраживање. У сличној литератури радови такође евалуирају Ф-меру [102], поред прецизности и сензитивности. Ф-мера је хармонијска средина прецизности и сензитивности. Ова мера је такође евалуирана у истаживању. Табела 7.10 показује задовољавајућу прецизност и сензитивност за нови садржај током једногодишњег инкременталног претраживања.

7.4.3 Искоришћеност протока.

Искоришћењем протока се мери способност претраживача да оптимизује и најбоље искористи пропусни опсег, што је веома битан фактор приликом претраживања. Ова мера се дефинише као:

$$Bu = \frac{B_{new} + B_{semiduplicate}}{B_{all}} 100\% \tag{7.7}$$

где је B_{all} цена протока која је дефинисана као укупан претражени садржај у временској јединици, док су B_{new} и $B_{semiduplicate}$ цене протока само новог и полу-дуплираног садржаја респективно, у поређењу са постојећим садржајем у бази података. Најбољи случај би био да приликом претраживања претраживач сакупља само веб стране које искључиво садрже само нови и полу-дуплицирани садржај, при чему би у том случају искоришћеност протока била 100%.

Резултати се могу видети у Табели 7.11, где је додатно приказана и искоришћеност протока FoCUS претраживача.

ТАБЕЛА 7.10
ПРОСЕЧНА ПРЕЦИЗНОСТ, СЕНЗИТИВНОСТ И F-МЕРА

Прецизност	Сензитивност	F-мера
91.6%	97%	94.22%

Као што је и очекивано, искоришћење пропусног опсега директно зависи од прецизности система за претраживање. Како претраживач понекад преузима стране које садрже дупликат или ирелевантан садржај, искоришћеност протока опада. Што је нижа прецизност, више ће се интернет протока непотребно утрошити на пренос података приликом претраживања Искоришћеност протока FoCUS претраживача је пропорционална великом броју дупликата који се појављују док се инкрементално претражује. Са друге стране, SInFo је максимално искористио интернет проток јер је оптимизовао број неопходних URL захтева за преузимање најновијег садржаја.

7.4.4 Компарација функционалности.

Табела 7.12 приказује поређење основних функционалности међу поменутим претраживачима форума. Карактеристике претраживача су класификоване према:

ТАБЕЛА 7.11
ИСКОРИШЋЕНОСТ ПРОТОКА НА ИНКРЕМЕНТАЛНОМ ПРЕТРАЖИВАЊУ ПЕРИОДА ОД ЈЕДНЕ ГОДИНЕ

Месец	Јануар 2016	Фебруар 2016	Март 2016	April 2016	Мај 2016	Јун 2016	Јул 2016	Август 2016	Септембар 2016	Октобар 2016	Новембар 2016	Децембар 2016
SInFo	88.82%	92.44%	94%	91.22%	91.87%	91.7%	90.98%	90.09%	92.7%	92.72%	90.81%	91.5%
FoCUS	20.13%	19.1%	20.9%	18.02%	13.8%	13.91%	13.93%	13.4%	12.71%	8.8%	8.27%	7.01%

Интервенција корисника: представља комплексност коришћења форумског претраживача. Vigi4Med захтева већу интервенцију корисника с обзиром на то да је потребно ручно написати XPath упите појединачно за сваки форум. CrimeBot захтева средњу до високу интервенцију корисника, јер се састоји од модула који су одговорни за сваку појединачну технологију форума. Претраживање нове појединачне или генеричке форумске технологије захтева писање и додавање нових модула у систем. Сличан концепт односи се на решење бр. 7. За решење бр. 6, семантичка правила у облику регуларних израза морају се прегледати и

ажурирати ако нису у складу са тренутном технологијом форума. Претраживачи као што су бр. 1, 2 или. 3 захтевају релативно ниске интервенције корисника.

ТАБЕЛА 7.12
КОМПАРАЦИЈА ОСНОВНИХ ФУНКЦИОНАЛНОСТИ ЈАВНО ДОСТУПНИХ ИНКРЕМЕНТАЛНИХ ПРЕТРАЖИВАЧА ВЕБ ФОРУМА

Форум	Интервенција корисника	Аутоматизација	Инкрементално претраживање:	Прилагодљиво инкрементално претраживање	Екстракција података	Начин рада
1. SInFo	Низак	Да	Да	Да	Да	RegEx изрази
2. FoCUS	Низак	Да	Не	Не	Не	RegEx изрази
3. iRobot	Низак	Да	Не	Не	Не	*Traversal path lookup*
4. Vigi4Med	Висок	Не	Не	Не	Да	XPath упити
5. CrimeBot	Средњи-висок	Не	Да	Полу	Да	XPath упити
6. Систем из [35]	Средњи	Полу	Не	Не	Да	RegEx семантичка правила/XPath
7. Систем из [37]	Средњи-висок	Не	Да	Полу	Да	Python скрипта

Аутоматизација: Дефинише способност претраживача да аутоматски проналази путање форума или детектује регионе са подацима на новим форумским технологијама које раније нису виђене. Претраживач бр. 6, је полу-аутоматизован и зависи од квалитета написаних семантичких правила. Са семантичким правилима доброг квалитета, овај систем за претраживање може бити прилагодљив и постићи висок ниво аутоматизације кроз различите врсте форумских технологија.

Инкрементално претраживање: Способност претраживача да инкрементално индексира форум циљајући само нови садржај, генерисан након последњег циклуса претраживања.

Прилагодљиво инкрементално претраживање за различите типове изгледа форума: прилагођавање претраживача различитим приказима форума и сортирања редоследа док се инкрементално претражује веб форум. Претраживачи бр. 5 и бр. 7 су полу-прилагодљиви, јер њихова способност да се прилагоде новим или другачијим стиловима редоследа сортирања увелико зависи од унутрашње строго предефинисане имплементације претраживања коју већ садрже.

Екстракција података: Способност система за претраживање да прикупи садржаје са постова веб форума.

Начин рада: Кратак опис интерне методе рада претраживача.

8 Закључак

У овом раду представљени су методи и технике на којима је изграђен прототип за ефикасно инкрементално претраживање форума - SInFo. Главна идеја SInFo претраживача је избегавање дупликата у сваком новом циклусу претраживања на форуму фокусирањем на генерички приступ који се прилагођава технологији форума и минимизирање пута до најновијег садржаја генерисаног након претходног датума претраживања. SInFo се састоји од две фазе: прве, почетне фазе у којој се детектује тип сортирања на индексним и дискусионим странама; и друге фазе, где користећи расположиве навигационе опције форума између страна и поштујући формат УРЛ-а, претраживач циља најновији садржај.

Без обзира на дистрибуцију новог генерисаног садржаја у наредним циклусима поновног претраживања, SInFo је постигао обећавајуће резултате. Изводећи експерименте на 14 веб форума конципираних од најпопуларнијих форумских технологија, заједно са репрезентативним индивидуалним технологијама, SInFo је значајно смањио дупликате у поређењу са имплементацијом FoCUS-а. На основу 100 насумично изабраних страна индекса и дискусија, показано је да предложени методи SInFo претраживача могу разликовати тип сортирања и репрезентацију садржаја с великом тачношћу, што је обавезно за касније претраживање форума током циљања искључиво ново-генерисаног садржаја. Укупне прецизности одређивања типа сортирања на индексним и дискусионим странама су 98,4% и 96,4% респективно, што подразумева робусност и високу адаптивност на разноликост презентација између форумских технологија. У тестовима приказаним у овом раду, SInFo је прикупио више од 92% корисних страна са садржајем које нису прикупљене у претходним претраживањима.

Даље, представљени су модели машинског учења за препознавање, екстракцију и нормализацију датума на стандардизован формат лако препознатљив претраживачу. Прецизна детекција и тумачење датума и времена је омогућила SInFo претраживачу исправно утврђивање редоследа сортирања на странама као и одабир метода неопходних за ефикасно циљање новонасталог садржаја. Посматрајући тестни и тренинг сет, модел за конверзију датума је постигао тачност од 99%, док NER модел има F1-меру од 99% на оба сета. Додатно, предложени NER модел је био за 4 и више пута ефикаснији у смислу детекције ентитета датума од постојећих, јавно доступних и већ тренираних *off-the-shelf* модела.

Као што је и очекивано, ефикасно циљање само најновијег садржаја је омогућило да SInFo претраживач максимално искористи пропусни опсег и оптимизује број преузимања потребних за дохватање страна које имају нов садржај. Такође је показано да без обзира на стране које немају нови садржај, али их претраживач ипак мора преузети, прецизност и сензитивност су били више него задовољавајући, са просечном прецизношћу од 91,6% и сензитивношћу од 97%. Поред наведених резултата, овај рад је допринео свеобухватним и опсежним истраживањима која су рађена на великом броју форумских технологија ради категоризације свих главних врста форумских технологија.

Будући рад ће се више усмерити на архитектуру базе података која може прихватити индекс и дискусионе стране као један објекат, заједно с другим објектима попут аутора, постова и датума који се могу појавити на страни. У овом истраживању, пројектовање адекватне базе података са приступом на нивоу објекта за време инкременталног претраживања се показало као нетривијалан задатак. Такође, планира се извођење детаљнијих и ширих експеримената укључивањем више софтверских пакета форума, а као амбициознији

циљ и имплементација интелигентног препознавања карактеристика нових форумских технологија како би се континуирано унапређивала употреба SInFo претраживача.

Литература

Референце које су коришћене у оквиру ове дисертације су наведене према редоследу појављивања у самом раду. За сваку референцу су наведене све битне информације, као и интернет страница уколико је доступна. Формат навођења референци одговара IEEE упутству предложеном у оквиру [број].

[1] M. Morzy, “Internet forums: what knowledge can be mined from online discussions,” in *Knowledge discovery practices and emerging applications of data mining: Trends and new domains*, IGI Global, 2011, pp. 315–336.

[2] R. Cai, J.-M. Yang, W. Lai, Y. Wang, and L. Zhang, “iRobot: an intelligent crawler for web forums,” in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 447–456.

[3] J. Jiang, X. Song, N. Yu, and C.-Y. Lin, “FoCUS: Learning to Crawl Web Forums,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1293–1306, Jun. 2013.

[4] J.-M. Yang, R. Cai, C. Wang, H. Huang, L. Zhang, and W.-Y. Ma, “Incorporating site-level knowledge for incremental crawling of web forums,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’09*, 2009, p. 1375.

[5] M. Pavkovic and J. Protic, “The use of an intelligent forum crawler for data retrieval from e-learning portals,” in *6th International Conference on Education and New Learning Technologies (EDULEARN)*, 2014, pp. 2441–2449.

[6] Y. Wang, J.-M. Yang, W. Lai, R. Cai, L. Zhang, and W.-Y. Ma, “Exploring traversal strategy for web forum crawling,” in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’08*, 2008, p. 459.

[7] “RFC 1738 - Uniform Resource Locators (URL),” *Network Working Group*, 1994. [Online]. Available: https://www.ietf.org/rfc/rfc1738.txt.

[8] “Forum Matrix,” 2017. [Online]. Available: http://www.forummatrix.org/index.php.

[9] “Wappalyzer,” 2017. [Online]. Available: https://www.wappalyzer.com/categories/message-boards.

[10] J. Kluge, F. Kargl, and M. Weber, “The Effects of the Ajax Technology on Web Application Usability.,” in *WEBIST (2)*, 2007, pp. 289–294.

[11] W. Morris, “9 Best Forum Software Picks to Build an Online Community in 2020,” 2020. [Online]. Available: https://www.hostinger.com/tutorials/best-forum-software.

[12] D. Hardt, “The OAuth 2.0 authorization framework,” RFC 6749, October, 2012.

[13] D. Recordon and D. Reed, “OpenID 2.0: a platform for user-centric identity management,” in *Proceedings of the second ACM workshop on Digital identity management*, 2006, pp. 11–16.

[14] D. Décary-Hétu and J. Aldridge, “Sifting through the net: Monitoring of online offenders by researchers,” *Eur. Rev. Organised Crime*, pp. 122–141, 2015.

[15] “phpBB,” 2020. [Online]. Available: https://www.phpbb.com/.

[16] K. R. Lakhani and E. Von Hippel, “How open source software works:‘free’ user-to-user assistance,” in *Produktentwicklung mit virtuellen Communities*, Springer, 2004, pp. 303–339.

[17] “myBB,” 2020. [Online]. Available: https://mybb.com/.

[18] “WordPress,” 2020. [Online]. Available: https://wordpress.com/.

[19] S. McKeever, “Understanding Web content management systems: evolution, lifecycle and market,” *Ind. Manag. data Syst.*, 2003.

[20] “Joomla!,” 2020. [Online]. Available: https://www.joomla.org/.

[21] J. Grappone and G. Couzin, *Search Engine Optimization (SEO): An Hour a Day*. John Wiley & Sons, 2011.

[22] “Drupal,” 2020. [Online]. Available: https://www.drupal.org/.

[23] “Vanilla,” 2020. [Online]. Available: https://vanillaforums.com/en/.

[24] “SMF - Simple Machines Forum,” 2020. [Online]. Available: https://www.simplemachines.org/.

[25] B. E. Brewington and G. Cybenko, “How dynamic is the Web?,” *Comput. Networks*, vol. 33, no. 1–6, pp. 257–276, Jun. 2000.

[26] B. E. Brewington and G. Cybenko, “Keeping up with the changing Web,” *Computer (Long. Beach. Calif).*, vol. 33, no. 5, pp. 52–58, May 2000.

[27] D. E. Holmes and L. C. Jain, Eds., *Data Mining: Foundations and Intelligent Paradigms*, 1st ed., vol. 25, no. 25. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[28] A. Goswami and A. Kumar, “A survey of event detection techniques in online social networks,” *Soc. Netw. Anal. Min.*, vol. 6, no. 1, p. 107, Dec. 2016.

[29] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

[30] M. Pavkovic and J. Protic, “Intelligent crawler for web forums based on improved regular expressions,” in *2013 21st Telecommunications Forum Telfor (TELFOR)*, 2013, pp. 817–820.

[31] M. Pavković and J. Protić, “The parallel crawler for indexing forum pages on the Internet,” in *2014 21st Conference on Electrical, Electronic and Computing Engineering (ETRAN)*, 2014.

[32] B. Audeh, M. Beigbeder, A. Zimmermann, P. Jaillon, and C. Bousquet, “Vigi4Med Scraper: A Framework for Web Forum Structured Data Extraction and Semantic Representation,” *PLoS One*, vol. 12, no. 1, pp. 1–18, Jan. 2017.

[33] L. Ora and S. R. Ralph, “Resource Description Framework (RDF) Model and Syntax Specification,” *W3C Recommendation*, 1999. [Online]. Available: https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

[34] J. Robie, D. Chamberlin, M. Dyck, and S. John, “XML Path Language (XPath) 3.0,” *World Wide Web Consortium*, 2014. [Online]. Available: http://www.w3.org/TR/2014/REC-xpath-30-20140408/.

[35] U. Baskaran and K. Ramanujam, “Automated scraping of structured data records from health discussion forums using semantic analysis,” *Informatics Med. Unlocked*, vol. 10, pp. 149–158, 2018.

[36] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, “CrimeBB,” in *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW ’18*, 2018, pp. 1845–1854.

[37] R. Williams, S. Samtani, M. Patton, and H. Chen, “Incremental Hacker Forum Exploit Collection and Classification for Proactive Cyber Threat Intelligence: An Exploratory Study,” in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2018, pp. 94–99.

[38] A. Baravalle, M. S. Lopez, and S. W. Lee, “Mining the Dark Web: Drugs and Fake Ids,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 2016, pp. 350–356.

[39] M. Henzinger, “Finding near-duplicate web pages: a large-scale evaluation of algorithms,” in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 284–291.

[40] M. Theobald, J. Siddharth, and A. Paepcke, “SpotSigs: robust and efficient near duplicate detection in large web collections,” in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 563–570.

[41] G. S. Manku, A. Jain, and A. Das Sarma, “Detecting near-duplicates for web crawling,” in *Proceedings of the 16th international conference on World Wide Web - WWW ’07*, 2007, p. 141.

[42] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, “Syntactic clustering of the Web,” *Comput. Networks ISDN Syst.*, vol. 29, no. 8–13, pp. 1157–1166, Sep. 1997.

[43] A. Bookstein, V. A. Kulyukin, and T. Raita, “Generalized hamming distance,” *Inf. Retr. Boston.*, vol. 5, no. 4, pp. 353–375, 2002.

[44] “The Sitemaps Protocol,” *Sitemaps XML format*, 2017. [Online]. Available: https://www.sitemaps.org/protocol.html.

[45] V. N. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[46] H. S. Koppula, K. P. Leela, A. Agarwal, K. P. Chitrapura, S. Garg, and A. Sasturkar, “Learning URL patterns for webpage de-duplication,” in *Proceedings of the third ACM international conference on Web search and data mining - WSDM ’10*, 2010, p. 381.

[47] X. Song, J. Liu, Y. Cao, C.-Y. Lin, and H.-W. Hon, “Automatic extraction of web data records containing user-generated content,” in *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM ’10*, 2010, p. 39.

[48] Yanhong Zhai and Bing Liu, “Structured Data Extraction from the Web Based on Partial Tree Alignment,” *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 12, pp. 1614–1628, Dec. 2006.

[49] W. Liu, H. Yan, and J. Xiao, “Automatically extracting user reviews from forum sites,” *Comput. Math. with Appl.*, vol. 62, no. 7, pp. 2779–2792, Oct. 2011.

[50] W. Liu, X. Meng, and W. Meng, “Vision-based web data records extraction,” in *Proc. 9th international workshop on the web and databases*, 2006, pp. 20–25.

[51] K. Simon and G. Lausen, “ViPER: augmenting automatic information extraction with visual perceptions,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005, pp. 381–388.

[52] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma, “Learning block importance models for web pages,” in *Proceedings of the 13th international conference on World Wide Web*, 2004, pp. 203–211.

[53] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, “Fully automatic wrapper generation for search engines,” in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 66–75.

[54] Y. Zhai and B. Liu, “Web data extraction based on partial tree alignment,” in *Proceedings of the 14th international conference on World Wide Web - WWW ’05*, 2005, p. 76.

[55] B. Liu, R. Grossman, and Y. Zhai, “Mining data records in web pages,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 601–606.

[56] J. Jansson and A. Lingas, “A fast algorithm for optimal alignment between similar ordered trees,” in *Annual Symposium on Combinatorial Pattern Matching*, 2001, pp. 232–240.

[57] R. Baeza-Yates, C. Castillo, M. Marin, and A. Rodriguez, “Crawling a country: better strategies than breadth-first for web page ordering,” in *Special interest tracks and posters of the 14th international conference on World Wide Web*, 2005, pp. 864–872.

[58] “IBM Date and Time Formats,” 2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSLVMB_sub/statistics_reference_project_ddita/spss/base/syn_date_and_time_date_time_formats.html.

[59] “Talend Calendar Types,” 2020. [Online]. Available: https://help.talend.com/reader/3zI67zZ9kaoTVCjNoXuEyw/Oipp3Pt9~TholtcfXMW8~w.

[60] M. Pavkovic and J. Protic, “SInFo–Structure-Driven Incremental Forum Crawler That Optimizes User-Generated Content Retrieval,” *IEEE Access*, vol. 7, pp. 126941–126961, 2019.

[61] “Datefinder,” 2020. [Online]. Available: https://github.com/akoumjian/datefinder.

[62] “SUtime,” 2020. [Online]. Available: https://nlp.stanford.edu/software/sutime.html.

[63] “Dateparser,” 2020. [Online]. Available: https://dateparser.io/.

[64] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Lingvisticae Investig.*, vol. 30, no. 1, pp. 3–26, 2007.

[65] R. Sharnagat, “Named entity recognition: A literature survey,” *Cent. Indian Lang. Technol.*, 2014.

[66] J. Li, A. Sun, J. Han, and C. Li, “A survey on deep learning for named entity recognition,” *IEEE Trans. Knowl. Data Eng.*, 2020.

[67] P.-H. Li, T.-J. Fu, and W.-Y. Ma, “Why Attention? Analyze BiLSTM Deficiency and Its Remedies in the Case of NER.”

[68] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[69] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv Prepr. arXiv1409.0473*, 2014.

[70] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” 1999.

[71] Y. LeCun, “Generalization and network design strategies,” *Connect. Perspect.*, vol. 19, pp. 143–155, 1989.

[72] C. Sutton and A. McCallum, “An introduction to conditional random fields,” *Found. Trends® Mach. Learn.*, vol. 4, no. 4, pp. 267–373, 2012.

[73] J. P. C. Chiu and E. Nichols, “Named entity recognition with bidirectional LSTM-CNNs,” *Trans. Assoc. Comput. Linguist.*, vol. 4, pp. 357–370, 2016.

[74] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, “Supervised learning of universal sentence representations from natural language inference data,” *arXiv Prepr.*

arXiv1705.02364, 2017.

[75] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[76] “Keras,” 2020. [Online]. Available: https://keras.io/.

[77] Y. Bengio, “Learning deep architectures for AI,” *Found. trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[78] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[79] “GloVe website,” 2020. [Online]. Available: https://nlp.stanford.edu/projects/glove/.

[80] “GloVe multilanguage soruces,” 2020. [Online]. Available: http://www.cs.cmu.edu/~afm/projects/multilingual_embeddings.html.

[81] D. Mueller, N. Andrews, and M. Dredze, “Sources of Transfer in Multilingual Named Entity Recognition,” *arXiv Prepr. arXiv2005.00847*, 2020.

[82] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.

[83] “Amazon Mechanical Turk,” 2020. [Online]. Available: https://www.mturk.com/.

[84] “Faker software package,” 2020. [Online]. Available: https://faker.readthedocs.io/en/master/.

[85] “nVidia Deep Learning Institute,” 2020. [Online]. Available: https://www.nvidia.com/en-us/deep-learning-ai/education/.

[86] “deeplearning.ai,” 2020. [Online]. Available: https://www.deeplearning.ai/deep-learning-specialization/.

[87] J. Brownlee, “Why One-Hot Encode Data in Machine Learning?,” 2020. [Online]. Available: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.

[88] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[89] “nVidia Tesla K80,” 2020. [Online]. Available: https://www.nvidia.com/en-gb/data-center/tesla-k80/.

[90] “AWS - Amazon Web Services,” 2020. [Online]. Available: https://aws.amazon.com/.

[91] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv Prepr. arXiv1412.6980*, 2014.

[92] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[93] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[94] K. You, M. Long, J. Wang, and M. I. Jordan, “How Does Learning Rate Decay Help Modern Neural Networks?,” 2019.

[95] R. Grishman and B. M. Sundheim, “Message understanding conference-6: A brief history,” in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[96] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel, “The Automatic Content Extraction (ACE) Program-Tasks, Data, and

Evaluation.," in *Lrec*, 2004, vol. 2, p. 1.

[97] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís, "Named entity recognition: fallacies, challenges and opportunities," *Comput. Stand. Interfaces*, vol. 35, no. 5, pp. 482–489, 2013.

[98] M. L. Patawar and M. A. Potey, "Approaches to named entity recognition: a survey," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 3, no. 12, pp. 12201–12208, 2015.

[99] E. F. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," *arXiv Prepr. cs/0306050*, 2003.

[100] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes," in *Joint Conference on EMNLP and CoNLL-Shared Task*, 2012, pp. 1–40.

[101] J. Cho and H. Garcia-Molina, "The Evolution of the Web and Implications for an Incremental Crawler," in *Proceedings of the 26th International Conference on Very Large Data Bases*, 2000, pp. 200–209.

[102] N. Chinchor and B. Sundheim, "MUC-5 evaluation metrics," in *Proceedings of the 5th conference on Message understanding - MUC5 '93*, 1993, p. 69.

Списак скраћеница

Списак скраћеница који се налазе у раду по редоследу појављивања.

Q&A – *Question and Answer*
FoCUS – *Forum Crawler Under Supervision*
AJAX – *Asynchronous JavaScript and XML*
NNTP – *Network News Transfer Protocol*
SSN – *Single Sign-On*
CMS – *Content Management System*
SEO – *Search Engine Optimization*
SMF – *Simple Machines Forum*
RDF – *Resource Description Framework*
SVM – *Support Vector Machine*
HTML – *HyperText Markup Language*
DOM – *Document Object Model*
GSRG – *Global Similarity Relationship Graph*
SInFo – *Structure-driven incremental forum crawler*
REPL – *RegEx Pattern Learning*
FC – *Forum Crawl*
IPSD – *Index Page Sort Detection*
TSD – *Thread Sort Detection*
RS – *Recrawl Scheduler*
PC – *Post Collect*
TD – *Thread Discovery*
LSTM – *Long Short Term Memory*
GloVe – *Global Vectors for Word Representation*
NER – *Named Entity Recognition*
IE – *Information Extraction*
BiLSTM – *Bidirectional LSTM*
CRF – *Conditional Random Fields*
ResNets – *Residual skip connections*
AMT – *Amazon Mechanical Turk*

CRD – *Creation Date*

LAD – *Last activity date*

ASC – *Ascending*

DSC – *Descending*

GPU – *Graphic Processing Unit*

AWS – *Amazon Web Services*

TP – *True Positives*

FP – *False Positives*

FN – *False Negatives*

TN – *True Negatives*

СПИСАК СЛИКА

СПИСАК ТАБЕЛА

Биографија аутора

Милош Павковић је рођен 14. новембра 1982. у Београду. Основну школу "Душан Јерковић" и техничку школу "Михајло Пупин" је завршио у Инђији. У октобру 2004. године уписао је основне студије на Електротехничком факултету, Одсек за софтверско инжењерство. Дипломирао је 2009. године, са просечном оценом 9, а дипломски рад је одбранио са оценом 10. Током студија, у оквиру "*Best Student Recognition*" догађаја, изабран је од стране IBM компаније за учешће у "*Internship*" програму, где је радио на пројекту "*A comparison of portlet container in IBM WebSphere Application Server and IBM WebSphere Portal*". Током пројекта боравио је у научном-истраживачком центру IBM-а у Ници, Француска.

Након дипломирања уписао је мастер студије на Електротехничком факултету 2009. године, на модулу Софтверско инжењерство. Положио је све испите предвиђене планом и програмом са оценом 9.83 и одбранио мастер рад са темом "Софтверски системи за примену Бајесових мрежа у медицини" са оценом 10.

На Електротехничком факултету је 2011. године уписао докторске студије на модулу Софтверско инжењерство код ментора проф. др Јелице Протић и положио све предмете са просечном оценом 10. У свом истраживачком раду показао је посебно интересовање за интелигентне системе претраживања на интернету, паралелизацију претраживања веб форума, аутоматску екстракцију података и примену машинског учења на аутоматизацију генерисања регуларних израза. Конципирао је и развио специјализовани претраживач веб форума, који кластеризацијом веб страна и применом машинског учења аутоматски прати скелетне линкове веб форума на оптималан начин и врши екстракцију кориснички генерисаног садржаја. Резултате тестова и истраживања је објавио на две међународне и три домаће конференције.

За време докторских студија, 10. октобра 2013. године изабран је у звање асистента за ужу научну област Рачунарске комуникације на Природно-математичком факултету Универзитета у Крагујевцу, и поново изабран у исто звање 10. октобра 2016. године. Као асистент на Природно-математичком факултету ангажован је на вежбама из предмета: Архитектура рачунара 1, Рачунарске мреже и мрежне комуникације, Оперативни системи 1. Додатно је ангажован од октобра 2017. године као асистент на Факултету инжењерских наука Универзитета у Крагујевцу, где држи предмет Рачунарске основе интернета на модулу за Софтверско инжењерство.

Од 1. новембра 2010. године па до данас је ангажован од стране фирме *Effyis Inc*. (сада *SocialGist Inc*.), са седиштем у Детроиту, Сједињене Америчке Државе, на конципирању и изради специјализованих претраживача веба.

У септембру 2006. године је основао фирму PulsArt. Током рада за ову фирму учествовао је у пројектовању и имплементацији локалних рачунарских мрежа, пројектовању инфраструктуре за паметне зграде, информационих система, веб портала, CMS веб базираних система и веб сајтова.

ИЗЈАВА О АУТОРСТВУ

Име и презиме аутора: **Милош Павковић**

Број индекса: **5002/2011**

Изјављујем

да је докторска дисертација под насловом

Побољшање перформанси прикупљања кориснички генерисаних садржаја на Вебу применом адаптивних интелигентних метода

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

У Београду, 28.09.2020. _____

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора: **Милош Павковић**

Број индекса: **5002/2011**

Студијски програм: **Електротехника и рачунарство**

Наслов рада: **Побољшање перформанси прикупљања кориснички генерисаних садржаја на Вебу применом адаптивних интелигентних метода**

Ментори: **проф. др Јелица Протић**

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањена у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, 28.09.2020 _____

Изјава о ауторству

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Побољшање перформанси прикупљања кориснички генерисаних садржаја на Вебу применом адаптивних интелигентних метода

која је моје ауторско дело.

Дисертацију са свим прилозима предао сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)
4. **Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)**
5. Ауторство – без прерада (CC BY-ND)
6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.

Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, 28.09.2020

1. **Ауторство.** Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.