УНИВЕРЗИТЕТ У НИШУ
ЕЛЕКТРОНСКИ ФАКУЛТЕТ

Ђорђе К. Петровић

Анализа структуре колекције правних докумената на основу њихове повезаности преко одређених језичких израза

ДОКТОРСКА ДИСЕРТАЦИЈА

Ниш, 2020.

UNIVERSITY OF NIŠ
FACULTY OF ELECTRICAL ENGINEERING

Đorđe K. Petrović

Analysis of the structure of collections of legal documents, based on their connection through certain linguistic forms

DOCTORAL DISSERTATION

Niš, 2020.

Подаци о докторској дисертацији

Ментор:	Др Сузана Стојковић, ванредни професор Универзитет у Нишу, Електронски факултет
Наслов:	Анализа структуре колекције правних докумената на основу њихове повезаности преко одређених језичких израза
Резиме:	Предмет истраживања ове дисертације је статистички приступ за обраду природног језика и екстракција језичких израза (ен. *linguistic forms*), са циљем проналажења језичких израза са највећом фрекфенцијом коришћења у правним документима. Овако добијени подаци су даље коришћени за истраживање информација у правним документима, за анализу референци или веза у правним документима, за анализу путања између повезаних правних докумената, као и за одређивање мере значаја неког правног документа са становишта веза између њих. Али, поред ове примене, употреба добијених података може да буде веома разнолика. Методологија и информације добијене на овај начин су добра основа и могу да се користе као улазни подаци за бројне друге анализе.
Научна област:	Електротехничко и рачунарско инжењерство (Рачунарство и информатика)
Научна дисциплина:	Анализа текста, Процесирање природних језика, Извлачење информација
Кључне речи:	истраживање језичких израза, истраживање правних докумената, истраживање текста, извлачење информација, обрада природног језика, анализа линкова
УДК:	004.8:(81'42+34)
CERIF класификација:	P170 Рачунарсто, нумеричка анализа, системи, контрола P176 Вештачка интелигенција
Тип лиценце Креативне заједнице:	**CC BY-NC-ND**

Data on Doctoral Dissertation

Doctoral Supervisor:	PhD Suzana Stojković, Associate professor University of Niš, Faculty of Electronic Engineering
Title:	Analysis of the structure of collections of legal documents, based on their connection through certain linguistic forms
Abstract:	This document employs a statistical approach in exploring language and extracting linguistic forms there contained, so as to identify the linguistic forms which are most frequently used in legal documents. Thus retrieved data can be used to research information, analyze references and links, trace pathways between correlating legal documents and establish the relevance of legal documents on the grounds of their mutual correlation. The retrieved data can further be utilized in various other manners. The methodology of this research and thus attained information form a good basis and act as input data for numerous further analyses.
Scientific Field:	Electrical and Computer Engineering (Computer Science)
Scientific Discipline:	Text mining, Natural language processing, Information extraction
Key Words:	linguistic forms mining, legal documents mining, text mining, information extraction, natural language processing, link analysis
UDC:	004.8:(81'42+34)
CERIF Classification:	P170 Computer science, numerical analysis, systems, control; P176 Artificial intelligence
Creative Commons License Type:	**CC BY-NC-ND**

Посвећено

Михаилу, Оīњену и Неди

са жељом да у свему буду бољи од своī оца.

Са искреним задовољством пишем текст који се налази на овој страници. То значи да се завршава мој посао око припреме ове дисертације, па сам помислио да је ово прилика да изразим своју захвалност онима који су ме подржали током целокупног процеса који је довео до овога. Велику захвалност дугујем

- Професорки *Милени Станковић*, на свему што сам од ње научио и што ми је дала подршку током истраживачког процеса који је довео до овог документа. Посебно желим да изразим захвалност за консултације, за које је она одвојила своје време, а које су за мене биле веома драгоцене;
- Професорки *Сузани Стојковић*, на одличној сарадњи, саветима и помоћи око израде овог документа;
- Професорима *Леониду Стоименову*, *Драгану Јанковићу* и *Мирјани Ивановић*, на томе што су својим ауторитетом подржали овај документ и мене;
- *Електронском факултету* у Нишу, на томе што сам имао прилику, част и задовољство да студирам на том факултету. Намера ми је да својим будућим радом макар мало допринесем угледу овог факултета;
- *Високој пословној школи струковних студија* из Ваљева, на подршци и на томе што имам могућност да се усавршавам уз свакодневни рад. Своје знање и искуство сам одавно ставио у службу и на корист овој Школи;
- Напослетку овог списка, а у ствари на првом месту у мом животу, мојој деци, *Михаилу*, *Огњену* и *Неди*, супрузи *Бојани*, родитељима *Мири* и *Крсти*, брату *Жарку*, за све што ми пружају. Ово је и њихов успех, а не само мој.

Садржај

1 Увод

У овом поглављу се налазе уводна разматрања, мотив за теоретско и практично истраживање које се налази у овом документу, методологија истраживања, као и преглед садржаја по поглављима.

Користе се разни термини који се односе на употребу различитих типова алгоритама за проналажење и извлачење корисних информација из текстова, као што су истраживање текста (ен. *text mining*), анализа текста (ен. *text analytics*) или машинско учење из текста (ен. *machine learning from text*). Поред тога, обрада природног језика се често третира ка засебно научно поље, мада је блиско повезано са овом облашћу. Како овај рад обухвата примену алгоритама за извлачење корисних информација из текста правних докумената, а на основу одређених језичких израза, тако и рад може да се сврста у претходно наведене области.

Процес проналажења и извлачења корисних информација из збирке текстуалних докумената захтева више различитих корака. Општи процес истраживања текстова се састоји од следећих корака [1]:

- Дефинисање проблема,
- Прикупљање потребних података,
- Дефинисање својстава,
- Анализа података и
- Тумачење резултата.

У наставку ће бити дате уводне напомене које се односе на дефинисање проблема, док ће остали кораци према претходној листи, бити дати касније у тексту.

1.1 Дефинисање проблема

Тема овог рада је „Анализа структуре колекције правних докумената на основу њихове повезаности преко одређених језичких израза“. Да би таква анализа могла да буде урађена, потребно је да претходно буде дефинисан модел структуре колекције правних докумената. Да би такав модел могао да буде дефинисан, потребно је да се примене технике за истраживање текста и да се пронађу и издвоје они језички изрази који се користе за дефинисање веза са другим правним документима. Такође, потребно је пронаћи начин за аутоматско или машинско откривање веза или референци у текстовима.

Било ком поступку истраживања текста, претходи поступак припреме текста за даље истраживање. Како је предмет истраживања колекција правних докумената на српском језику, посебна пажња треба да буде посвећена особеностима и изазовима који се односе на истраживање и рад са текстом на српском језику. Према информацијама којима располажемо, за текстове који су на српском језику не постоје јавно доступни алати за припрему текстова, који су у стандардној употреби. На пример, постоји неколико имплементација процеса за извлачење морфолошког корена речи, али још увек ни једна од њих не даје максималну тачност, нити је у општој употреби. У складу са тим, примена различитих метода за препроцесирање текстова на српском језику, може касније да произведе разлике у добијеним резултатима. Због тога је потребно да се процени утицај различитих алата за припрему текстова на српком језику, који ће се у даљем истраживању користити. Процена утицаја алата за припрему текста и цео поступак припреме ће бити спроведен на примеру израчунавања сличности између статута акредитованих високошколских установа у Републици Србији. Статути су правни документи. Они уређују исте теме, али за различите установе. Креирани су и усвојени од стране различитих установа и као такви су погодни за истраживање сличности између текстова, уз употребу одговарајућих методологија и алата.

Проблеми, сложени по редоследу по коме треба да буду решавани, су:

- процена утицаја различитих алата за припрему текстова на српком језику, на даље истраживање,
- проналажење и издвајање оних језичких израза који се користе за дефинисање веза са другим правним документима,
- проналажење начина за аутоматско или машинско откривање веза или референци у правним текстовима,
- дефинисање модела структуре колекције правних докумената и
- анализа структуре колекције правних докумената.

Мотив за теоретско и практично истраживање је предлагање решења за претходно набројане проблеме.

1.2 Информације у правном домену

Информације у правном домену се често чувају као текст у релативно неструктуираном облику [2]. На пример, статути, пресуде и коментари се чувају као слободни текстуални документи. Структура таквих текстуалних докумената може да

буде уређена на различите начине. На примеру националног законодавства Републике Србије, постоји документ под називом "Јединствена методолошка правила за израду прописа" [3], које је 2010. године донео Законодавни одбор Народне скупштине Републике Србије. Овај документ одређује правила која би требало да се примењују при изради закона, одлука, пословника и других општих аката које доноси Народна скупштина Републике Србије. Овај документ уређује све што је потребно са правног аспекта, међутим, у њему није дефинисано ништа са аспекта аутоматског управљања комплексним информацијама које се налазе у оваквим правним документима. Осим тога, а на примеру националног законодавства Републике Србије, постоје закони и други прописи који су донети пре појаве Јединствених методолошких правила за израду прописа [3], и као такви нису нужно направљени у складу са овим правилима. Све ово указује на то да постоји разноликост у структури и садржају ових прописа.

Као предлог за превазилажење оваквих појава, постоји иницијатива да се стандардизују метаподаци о правним текстовима [4]. Визија предлагача је била да постоји унапређени скуп тагова метаподатака и унапред дефинисаних правних термина, како би се омогућила аутоматска размена и повезивање правних докумената на Интернету и за потребе шире друштвене заједнице. Ипак, до прихватања и примене оваквих стандарда, сада су у употреби правни документи који нису уређени на овај начин.

Откривање знања преко аутоматске анализе слободног текста је поље истраживања које је еволуирало од истраживања у области претраживање информација (ен. *information retrieval research*) и назива се истраживање текстова (ен. *text mining*) [2]. Аутоматско преузимање информација из велике збирке докумената је била једна од првих примена рачунарске науке на законе и то је задатак који је и данас присутан [2]. Два главна питања која се тичу података у овој области су [5]:

- побољшање техника и метода за руковање комплексношћу знања у овој области
- и проналажење погодних начина за чување и преузимање информација.

На неки начин, методологија која је описана у овом раду покушава да да одговор и свој допринос на оба претходна питања.

1.3 Методологија истраживања

Методологија која ће овде бити описана се састоји од примене неких техника и алгоритама из области истраживања текстова, у које између осталог спада извлачење

информација из текстова закона и других прописа, са циљем практичне примене аутоматског управљања са комплексним информацијама које се налазе у правним документима. Биће описан поступак проналажења језичких израза са највећом фрекфенцијом коришћења у текстовима закона који се примењују на територији Републике Србије, а затим ће посебна пажња бити посвећена језичким изразима који се користе за референцирање или повезивање. Подаци добијени на овај начин могу да се користе за извлачење информација у правним документима, за анализу референци или веза у правним документима, за анализу путања између повезаних правних докумената, за одређивање мере значаја неког правног документа са становишта веза између њих, као и за многе друге анализе. Резултати овог рада посебну примену могу да имају када је потребно спровести промене у законима и другим прописима. Током циклуса еволуције неког закона, он може да се мења више пута [6]. Промене у неком пропису имају утицај на све повезане прописе и области којима се они баве. Успостављањем система веза међу правним документима, законима и другим прописима, може да се постигне ефикасно управљање променама у тим документима, имајући у виду сву међузависност између њих. На тај начин је направљен покушај да се да одговор на прво питање које спада у аутоматско преузимање информација из велике збирке докумената [5], а то је побољшање техника и метода за руковање комплексношћу знања у овој области.

Осим тога, са становишта техника и метода за руковање комплексним правним документима, у поступку за трансформацију, тачније за реструктуирање података, описан је начин који је коришћен за складиштење и чување правних докумената. Имајући у виду да је у законима основна класификациона јединица **Члан закона**, биће описана и примењена сегментација текстова закона на поједине чланове закона, на начин да сваки члан буде сачуван као један запис у бази података. Ово је метод који је погодан за чување и преузимање информација из текстова закона и других прописа, а који олакшава даљу анализу и руковање комплексим правним документима. На тај начин је направљен покушај да се да одговор и на друго питање која спада у аутоматско преузимање информација из велике збирке докумената [5], а које се бави проналажењем погодних начина за чување и преузимање информација.

Последњих година модели неуронских мрежа постижу значајне резултате у различитим истраживачким областима, као што су у рачунарски вид (ен. *Computer vision*), пропознавање говора (ен. *speech recognition*), обрада природног језика (ен. *Natural Language Processing*) и другим областима, за које је тешко дефинисати како их

треба решити. У овом документу ће бити описано прављење модела и примена неуронских мрежа, као могућег решења за потребе машинског предвиђања постојања веза или референци у необележеним подацима, као што су текстови нових закона.

Поред свега наведеног, у овом документу ће бити описана и примена техника и алата који спадају у теорију графова, на скупу текстуалних података и скупу веза које су откривене у тим подацима, најпре са циљем анализе путања између повезаних докумената, али и за одређивање мере значаја неког од докумената са становишта веза између њих.

Информације добијене истраживањем језичких израза у правним документима, методологије које су овде описане, алати и добијени резултати, могу да буду одлична основа и улазни подаци за многе даље анализе.

1.4 Преглед садржаја по поглављима

У поглављу са насловом „**Преглед релевантних истраживања**“ су цитирани други радови на ову исту тему, са коментарима о томе у чему је овај документ исти, односно у чему потврђује те друге радове, а по чему се овај документ разликује од тих других радова.

У поглављу „**Машинско учење из текстова**“ је дат општи теоретски опис, приказ основних појмова, технологија, алгоритама и апликација, које се односе на ово истраживање. У оквиру овог поглавља, у потпоглављу са насловом „**Извлачење информација из текстова**“ је дат детаљни теоретски опис и приказ актуелних технологија и алгоритама који се користе у овој ужој области. Део потпоглавља, са насловом „Истраживање текстова који се налазе на вебу“ је објављен на конференцији „Телфор“ [7].

Сваки поступак машинског истраживања текстова има више фаза, а једна од обавезних фаза је припрема текстова за истраживање. У поглављу „**Припрема текстова на српском језику, за потребе даљег истраживања**“ је део који је посвећен припреми текстова за даље истраживање са посебним освртом на рад са правним документима који су на српском језику. Алгоритми који се баве текстовима, могу да се поделе на оне алгоритме који су независни од језика на коме је написан текст који је предмет истраживања и на оне који су специфични за поједине језике. Како се у овој фази налазе алгоритми који су специфични за језик на коме је текст написан, то је овој материји дата посебна пажња, са практичним примерима примене појединих алгоритама. Текстуални

документи, који су коришћени у истраживању су правни документи, тачније статути високошколских установа у Републици Србији. Статути уређују исте теме, али за различите установе, креирани су и усвојени од стране различитих установа и као такви су погодни за истраживање сличности између текстова тих докумената, уз употребу одговарајућих методологија и алата. У овом поглављу је описан оригинални истраживачки рад са примерима примене појединих алгоритама за припрему текстова. Велики део истраживања приказаних у овом поглављу су описани у раду који је објављен у часопису „*Facta Universitatis, Series: Mathematics and Informatics*“ [8].

У поглављу са насловом „**Истраживање језичких израза у правним документима**“ је описан оригинални истраживачки рад. Велики део овог поглавља је објављен у часопису „*Computer Science and Information Systems*“ [9]. У оквиру овог поглавља, у прва три потпоглавља је описан поступак прикупљања, припреме и трансформације података. У потпоглављу са насловом „Истраживање језичких израза“ је описано опште истраживање коришћених језичких израза у текстовима српских закона. У потпоглављу са насловом „Језички изрази за анализу веза у правним документима“ је посебна пажња посвећена овим језичким изразима.

Текст је један од најраспрострањенијих секвенцијалних података и као такав је погодан за примену модела дубоког машинског учења, при чему се дубоко машинско учење примењује на препознавање образаца кроз обраду природног језика. У поглављу са насловом „**Примена неуронских мрежа за предвиђање језичких израза за повезивање у текстовима закона**“ ће бити објашњени принципи и дати примери обучавања различитих типова неуронских мрежа, за потребе класификовања текстова и за потребе проналажења или предвиђања постојања језичких израза за повезивање, у текстовима нових закона и прописа. Садржај овог поглавља је подељен у два дела. Уопштен опис машинског учења из текстуалних података је дат у првом делу, а у другом делу је показана примена неуронских мрежа за потребе учења и предвиђања језичких израза за повезивање у текстовима закона. Део потпоглавља са насловом „Матрица за уградњу“ је објављен на конференцији „*2019 International Conference on Artificial Intelligence: Applications and Innovations, IC-AIAI 2019*“ [10].

Скуп текстуалних података, који су предмет овог рада, као и скуп веза које су откривене, су погодни да се на тај модел примене технике и алати који спадају у теорију графова. У поглављу „**Примена теорије графова на правне документе и везе између њих**“, у потпоглављу „Анализа путања између повезаних закона“ ће бити описана

анализа чији је циљ да се пронађу тзв. „елементарне путање“ унутар посматраног модела података на основу откривених веза између њих, као и проналажење циклуса. У наредном потпоглављу ће бити урађено проналажење правних докумената „веће важности“ са становишта откривених веза између њих. Велики део овог поглавља је објављен у часопису „*Computer Science and Information Systems*“ [9].

У наставку се налази „Дискусија и будући рад“, „Закључак“, „Литература“, „Прилози“, „Биографија аутора“ и „Изјаве аутора“.

2 Преглед релевантних истраживања

У овом поглављу ће бити дат преглед књига и истраживања која се баве истим или сличним областима као ово истраживање. Према ужим областима цитирана истраживања могу да се уопштено групишу на следећи начин:

- Истраживања која се уопштено баве машинским учењем, истраживањем текстова, неуронским мрежама и употребом програмских језика;
- Истраживања која се баве специфичностима рада са текстова који су на српском језику или на неком другом језику и
- Истраживања која се баве анализом правних текстова.

Неке од последњих књига из области Машинског учења из текстова су „*Text Mining with Machine Learning: Principles and Techniques*“ [1], „*Machine Learning for Text*“ [11] и друге књиге, у којима се описују актуелни алгоритми и апликације у овој области.

Особеностима истраживања српског језика су се у свом истраживању бавили Витас и остали аутори [12], а закључак до кога су дошли Кајан, Пљасковић и Црнишин [13] је да је машинска обрада текстова на српском језику прави изазов. Постоји и неколико радова на теме које су повезане са методама и алатима за припрему српског језика за даље истраживање, што је предмет и овог рада.

У свом раду Кешељ и Шипка [14] су представили општи суфиксни метод за конструкцију стемера (ен. *stemmer*) и лематизера (ен. *lemmatizer*) за језике са богатом флексијом и оскудним ресурсима, као што је српски језик. Евалуација, коју су урадили на веродостојим подацима је дала тачност њиховог метода од 79%. Стемер за уклањање суфикса у текстовима на српском језику је представљен у истоименом раду, чији је аутор Милошевић [15]. Овај стемер је пружио подстицај за даљи рад на пољу истраживања текстова на српском језику. Ефектом поступка морфолошке нормализације (ен. *Morphological Normalization*) и уградње речи (ен. *Word Embeddings*) на класификацију осећања (ен. *Sentiment Classification*) у документима на српском језику су се бавили аутори Батановић и Николић [16]. Они су оцењивали утицај лематизера и стемера на класификаторе, који су обучени и процењени на скупу података о српском прегледу филмова. Открили су да је употреба стемера боља од употребе лематизера, у условима у којима су вршили истраживање, како у погледу тачности класификације, тако и у погледу ефикасности нормализације. Ипак, за разлику од резултата које су они добили, а по којима се стемер који је направљен за хрватски језик [17] показао као најбољи

кандидат у коришћењу н-грамских функција вишег реда, у истраживању које је спроведено и које ће овде бити описано, а за потребе одређивања сличности између дугачких текстова, добијени су супротни резултати од њихових. Сви претходно наведени аутори су дали свој значајан допринос даљем истраживању у области машинске обраде текстова на српском језику.

Осим радова у којима су описана истраживања текстова који су на српском језику, свакако постоје и радови који су се бавили особеностима истраживања текстова на другим језицима. У експерименту који је изведен на корпусу докумената о политици, а који су на чешком језику [18], аутори су истраживали утицај техника припреме текста за потребе откривања плагијата. У свом истраживању, они нису користили стеминг, већ лематизацију, као један од поступака за припрему текстова за даље истраживање. У раду [19], који се бавио категоризацијом текстова који су на арапском језику је спроведено истраживање утицаја, како се методе припреме текстова односе на перформансе три алгоритма машинског учења, *Naive Bayesian*, *DMNBtext* и *C4.5*. За разлику од поменутих радова, у овом раду је између осталог коришћено четири различита алата за стеминг и урађена је процена њиховог утицаја на израчунавање сличности између текстова који су на српском језику.

Постоји неколико истраживања, која се баве истраживањем "језичких израза", фраза, "секвенци састављених од речи" и слично. У књизи „*Phrase Mining from Massive Text and Its Applications*" [20], аутори се баве истраживањем аутоматске идентификације висококвалитетних фраза из бројних докумената. Они су предложили методологије које су засноване на фразама променљиве дужине, а такође су представили и апликације које омогућавају истраживање фраза. Иако су у поменутом истраживању аутори користили израз „phrases", у овом документу је коришћен израз „Језички изрази" (ен. *Linguistic forms*), јер је закључено да је тај израз адекватнији за опис правних термина (или израза) који су предмет овог рада, што је касније и објашњено у тексту.

Такође, постоји неколико радова која се баве истраживањем фрекфенција појављивања језичкиих израза у документима, затим откривањем кључних језичких израза, откривањем "правог квалитета" језичких израза и слично. У радовима [21] и [22], аутори су се бавили истраживањем аутоматске класификације докумената. Они су се фокусирали на проблем извлачења кључних фраза из збирке текстова, како би их користили као атрибуте за класификацију. Они траже секвенце речи (кључне фразе) које ће користити као својства за правила класификације, а не за извлачење правила

удруживања. Аутори [21] су закључили да избор кључних фраза не треба да се базира на оним кандидатима за фразе које су честе (у оквиру целе збирке текстова), већ на основу избора кључних фраза које постоје у неколико тесктова у колекцији, али су прилично чести унутар ових текстова. Осим тога, и аутори [20] тврде да сирова фрекфенција (ен. *raw frequency*) из података има тенденцију да створи погрешну процену квалитета и они покушавају да исправе убедљиву сирову фрекфенцију како би се открио прави квалитет фразе, испитивањем контекста њеног помињања. Циљ овог рада није класификација докумената. Један од корака у овом раду је проналажење језичких израза са највећом фрекфенцијом коришћења у посматраним текстовима, а затим и провера да ли тако добијени подаци могу да се користе за неке даље анализе. У наставку овог рада, предмет анализе је употреба језичких израза за анализу веза између правних докумената. Али, поред ове примене, употреба података који су добијени може да буде веома разнолика. Свакако, резултати добијени овим истраживањем могу да се у неком будућем раду користе као улазни подаци за даљу анализу квалитета језичких израза као што је то описано у [20].

Постоји и више радова који се баве анализом текстова правних докумената, анализом референци у правним документима, као и више радова који су се бавили националним правним документима појединих држава.

У раду [23], аутори су предложили приступ машинског учења за потребе идентификације реторичких улога у правним документима. То су постигли екстракцијом и модификованим рангирањем реченица са придодатим значењем њихових конкретних улога у тексту. Дотакли су се и језичких израза који се често користе као показатељи заједничких реторичких улога реченица. За разлику од поменутог рада, предмет овог рада нису целе реченице и њихова улога у тексту, већ су предмет рада језички изрази који су коришћени у текстовима правних докумената и у наставку је представљена методологија како је то урађено.

Као што је написано у књизи [24], правни документи су препуни имплицитних и експлицитних референци, а ово истраживање је то и потврдило. Постоји више радова у којима се описују поступци анализе линкова у правним документима. У свом раду [25], аутори су описали софтверски оквир (ен. *software framework*) за детекцију и решавање референци у националним и ЕУ законодавствима, судској пракси, парламентарним документима и службеним гласилима. У свом раду [26], аутори су описали примену алгоритма за анализу линкова (*Page Rank*) за потребе проналажења информација у

правним документима. Затим, у свом раду [27], аутор је описао анализу цитата из канадске судске праксе (ен. *Canadian Case Law*) при чему су анализирали постојеће цитате са одговарајућег веб-базираног система који су користили.

Постоји још неколико истраживања која су се бавила анализом линкова између правних докумената, као што су [28], [29], [30]. Углавном, у оваквим истраживањима је анализа линкова рађена на основу података који су доступни у *XML* формату и који већ у себи имају *XML* ознаке које се користе за означавање веза између докумената. У свом раду [28] аутори истражују цитатну мрежу кодирања које се користи у Сједињеним америчким државама (ен. *source of Federal law*) преко испитивања дистрибуције усмереног степена мреже (ен. *directed degree distributions of the network*). За потребе поменутог истраживања, аутори су добили *XML* снимак кода, где су цитати експлицитно кодирани унутар ових *XML* докумената на нивоу секција. Истраживањем мреже правних докумената (ен. *Legislation Networks*) који се примењују на подручју Новог Зеланда су се бавили аутори [29] у свом истраживању. Као и у претходном примеру, аутори су користили *XML* формат са веб сајта Владе Новог Зеланда, са везама између докумената. За разлику од наведених истраживања, за потребе овог истраживања су као извор података коришћени текстуални документи без било каквих унапред припремљених цитата или линкова и на том скупу докумената ће бити показан поступак за проналажење цитата или референци у тексту.

У свом раду [31] аутори су представили приступ за извлачење машински читљивог семантичког приказа законодавства, из неструктуираног формата докумената. Њихов метод изражава структуру правних докумената у облику скупа синтаксних правила тј., специфичног језика за правне документе, и проценили су овај приступ на скупу грчких правних докумената. Метод који је описан у [31] подразумева да сви правни документи буду конвертовани у датотеке са обичним текстом (ен. *plain text files*) и затим су радили конверзију наслеђених и правних докумената са обичним текстом у стандардни *XML* формат и тако су стварали *XML* датотеке које су компатибилне са *Akoma Ntoso* алгоритмом. *Akoma Ntoso* је шема за моделирање парламентарних, законодавних и судских докумената [32]. У овом раду, предмет истраживања није унутрашња подела прописа на блокове са информацијама, нити подела на шире класификационе јединице. За потребе овог рада, прописи су дељени на чланове закона. Како је члан закона јединствени логички ентитет, то је у овом раду спроведена сегментација текстова закона у појединачне чланове закона.

Аутори [30] су у свом раду користили јавно доступне податке из немачког законодавства и анализирали су законе са становишта случајности различитих типове веза у законским документима. У овом истраживању је потврђено постојање ових типова референци или веза на примеру скупа прописа који су предмет анализе, а који се односе на законодавне документе који се примењују у Републици Србији.

У свом раду [6], аутори су се бавили проблемом постојања различитих верзија закона, који се чувају у систему за контролу ревизија и како омогућити корисницима да посете актуелну верзију сваког закона, да прегледају историју ревизија и да прате промене између различитих ревизија, а затим како да се аутоматски примене измене и да се оне објаве у систему за контролу ревизија. У овом раду, предмет истраживања нису различите ревизије закона, већ се посматра веза између закона и указује на могући утицај измена неког од закона на скуп других важећих закона који су повезани са њим.

Постоје и истраживања која су се бавила обрадом правних текстова на српском језику. Предмет истраживања представљеног у [33] се односи на специфична језичка правила у законодавним текстовима на српском језику која се могу изразити помоћу лингвистичких метода подржаних рачунарима. У овом раду су детаљно описане фразе за референцирање (ен. *referencing phrases*) и структура правног акта, при чему су коришћене методе обраде природног језика (ен. *natural language prosecсing*) које су засноване на језичким правилима (ен. *language rules*). Према [34], постоје две стратегије: приступ заснован на правилима (ен. *Rule-based approach*) и статистички приступ (ен. *Statistical approach*). У истраживању које је предмет ове дисертације се користи статистички приступ за обраду природног језика, који није био предмет поменутог истраживања. Статистички метод су користили и аутори [35] у свом раду уз образложење да је српски језик високо флексибилан језик са врло ограниченим електронским лингвистичким ресурсима.

За истраживања која су овде описана и спроведена, програмски код у језику *Python* (https://www.python.org/) је писан применом платформе *Anaconda* (https://www.anaconda.com/) и употребом окружења *Jupyter*. Платформа *Anaconda* је коришћена јер пружа погодно окружење и намењена је за научна истраживања из области науке о подацима, бесплатна је за коришћење, то је дистрибуција отвореног кода и подржава рад са програмским језицима *Python* и *R*. Ова платформа омогућава прикупљање података, управљање окружењем, употребу моћних алата отвореног кода и друго. Као саставни део платформе *Anaconda*, коришћено је окружење *Jupyter Notebook*,

које пружа идеално окружење за анализу података и приказ и анализу резултата у реалном времену.

Применом програмског језика *Python* је такође извршена обука неуронских мрежа по узору на експеримент у коме су за потребе класификовањања текстова коришћене неуронске мреже [36]. Уз мање измене у односу на наведени експеримент и на примеру скупа података који је предмет овог рада, обучаване су следеће неуронске мреже: Рекурентна неуронска мрежа (ен. *Recurrent Neural Network, RNN*), Конволуциона неуронска мрежа (ен. *Convolutional Neural Network, CNN*), Хијерархијска мрежа са уграђеним механизмом пажње (ен. *Hierarchical Attention Network, HAN*).

3 Машинско учење из текстова

У овом поглављу ће бити дат општи теоретски опис, приказ основних појмова, технологија и алгоритама, како би боље биле објашњене методе и алати који су коришћени у истраживању које ће касније бити описано.

Извлачење корисних информација из текстова, уз употребу различитих типова статистичких алгоритама се назива истраживање текстова (ен. *text mining*), анализа текстова (ен. *text analytics*) или машинско учење из текстова (ен. *machine learning from text*). Актуелни алгоритми и апликације, који се баве истраживањем текстова, могу да се разврстају у неколико група:

- Припрема текстова (ен. *Text Preparation*) и израчунавање сличности (ен. *Similarity Computation*)
- Основне апликације за истраживање текстова, као што су Декомпозиција матрице (ен. *Matrix Factorization*), Моделирање тема (ен. *Topic Modeling*), Кластеризација (ен. *Text Clustering*), Класификација текстова (ен. *Text Classification*) и Спојено истраживање текстова са хетерогеним подацима (ен. *Joint Text Mining with Heterogeneous Data*)
- Претраживање и рангирање информација (ен. *Information retrieval and ranking*)
- Истраживање текстова усмерено на рад са секвенцама и природним језиком, у шта спада Моделирање секвенци текста и дубоко учење (ен. *Text Sequence Modeling and Deep Learning*), Сумирање текстова (ен. *Text Summarization*), Извлачење информација (ен. *Information Extraction*), Истраживање мишљења и анализа осећања (ен. *Opinion Mining and Sentiment Analysis*) и Сегментација текстова и откривање догађаја (ен. *Text Segmentation and Event Detection*)

Неки од набројаних алгоритама ће бити описани и примењени за потребе овог рада.

3.1 Репрезентације текстова

Приликом рада са текстовима, потребно је знати како користити и како комбиновати технике за руковање текстом, чија дужина може да се креће у распону од неколико појединачних речи, преко докумената, до целокупних колекција докумената. Поред тога, апликације и алгоритми, да би могли да буду примењени, захтевају да текст

буде претворен у одговарајући облик или репрезентацију. У апликацијама које се баве истраживањем текстова, постоје две репрезентације текстова:

- Текст као прост скуп или „врећа“ речи (ен. *Text as a bag-of-words*)
- Текст као скуп секвенци (ен. *Text as a set of sequences*)

Текст као врећа речи је најчешће коришћена репрезентација за истраживање текстова. Овај приступ је заснован на претпоставци да су речи међусобно независне и да редослед речи у тексту може да се занемари. Иако ни једна од ових претпоставки није реална, овај приступ се показао као врло једноставан и врло ефектан и као такав се користи у пракси. У овом случају, сваки документ се претвара у вектор учесталости појављивања речи у датом документу, односно у ретко поседнуту вишедимензионалну репрезентацију, која се затим користи за потребе истраживања. За многе апликације, као што су класификација, моделирање система и системи за препоручивање, овај тип репрезентације је довољан.

Текст као скуп секвенци је репрезентација текста у којој се поједине реченице у документу издвајају као низови или секвенце. Дакле, овде је важан редослед речи, иако је редослед често локализован унутар граница реченице или пасуса. Често се документ третира као скуп независних и мањих јединица (нпр. реченица или пасуса). Овај приступ користе апликације које захтевају већу семантичку интерпретацију садржаја документа.

У овом раду су коришћене обе репрезентације текста. У поглављу 4, које се бави припремом текстова и израчунавање сличности између текстова, је коришћена репрезентација текста као „вреће речи“, а у поглављу 5, које се бави истраживањем језичких израза у правним документима, и у поглављу 6, које се бави применом неуронских мрежа, је коришћена репрезентација текста као скупа секвенци.

3.2 Секвенцијално језичко моделирање

Иако је векторско представљање текста корисно за решавање многих проблема, постоје апликације у којима је веома важан секвенцијални приказ текста. Конкретно, свака апликација која захтева семантичко разумевање текста, изискује третирање текста као секвенце, а не као „вреће речи“ (ен. *bag-of-words*). У истраживању које је описано у поглављу 5.4, а које се бави истраживањем језичких израза у правним документима, је коришћена репрезентација текста као скуп секвенци. Осим тога, у поглављу 6 је описана примена неуронских мрежа за учење из секвенцијалних података. Два су основна разлога када је оваква репрезентација текста посебно корисна [11]:

1. **Разлози који се односе на податке** - Када је основни документ велики, репрезентација текста као „вреће речи“ садржи довољно информација у облику фрекфенција речи, за примену машинског учења. У неким поставкама, дужине текстуалних јединица су мале. На пример, текстови који одговарају микро блоговима или твитовима су релативно кратки. У таквим случајевима једноставно нема довољно информација у репрезентацији текста у облику „вреће речи“, да би се направили смислени закључци. Да би се највише извукло из ограничених података, за кратке текстове се користи секвенцијално језичко моделирање.
2. **Разлози који се односе на примену** – Многе апликације, као што су сумирање текстова, извлачење информација, истраживање мишљења и слично, захтевају семантички увид у текстове. Семантичко разумевање се може добити само третирањем реченица као секвенци.

Редослед речи изражава семантику која се не може извести из репрезентације текста у облику „вреће речи“. На пример, посматрајмо следеће реченице:

Ветар је одувао снег.

Снег је одувао ветар.

Јасно је да су ове две реченице различите. Ако се има у виду да је уобичајени редослед речи у српском језику субјекат-предикат-објекат, онда је друга реченица необична, али са становишта „вреће речи“ ове речнице су идентичне.

Постоје два општа приступа за рад са језичким структурама из реченица:

1. **Методе које су специфичне за језик** – Ове технике у процесу учења користе облике речи и друге језичке облике, што захтева подршку од стране лингвиста одређеног језика. Пример методе која је специфична за језик је употреба скупа граматичких правила, која су специфична за дати језик. Кључна ствар је да је коришћење знања из домена лингвистике, уз употребу правила, од суштинског значаја за функционисање таквих система машинског учења.
2. **Методе које не зависе од језика** – Ове технике креирају језички модел искључиво коришћењем статистичке анализе секвенцијалног редоследа речи. Статистички модел језика је дистрибуција вероватноће по секвенцама речи, и модел учи статистичку вероватноћу да реч следи секвенцу речи у реченици. Ови модели могу да се користе у раду са произвољним језицима и апликацијама јер се основне презентације језика уче на начин вођен подацима, без значајног знања из домена.

Људски језик је довољно нетачан и сложен, у смислу варијација у употреби, да није једноставно декодирати семантичку интерпретацију реченице, која се темељи искључиво на граматичким правилима. Веома често, наше разумевање реченица се заснива на коришћењу нашег интуитивног животног искуства у учењу и семантичког значења реченица из примера који се не могу буквално кодирати.

Проблем, који је уско повезан са статистичким моделирањем језика, је проблем кодирања свих информација у секвенцама текста. Овим проблемом се бави метода машинског учења и метода за истраживање текстова, под називом Инжењерство својстава (ен. *feature engineering*).

3.3 Истраживање текстова који се налазе на вебу

Текстуални садржаји, који су предмет истраживања које је описано у овом раду, су прикупљени са различитих локација на вебу. Адресе локација и поступак прикупљања су детаљно описани у наставку у поглављима 4.1 и 5.1.

Истраживање текстова који се налази на веб страницама се значајно разликује од истраживања текстова који се налазе у текстуалним документима. Уместо обичног текста, веб странице користе ХТМЛ за означавање и приказ свог садржаја. На тај начин се осим текстуалних информација, чувају и информације о форматирању, о другим мултимедијалним садржајима, линковима и слично. Ова комбинација својстава, које пружа ХТМЛ, обезбеђује и могућности и изазове за алгоритме који се баве истраживањем текстова [34]. Линкови, дефинисана ХТМЛ структура и атрибути за означавање стилова, представљају најкорисније ствари у истраживању текстова на вебу.

Цитатна структура веба игра веома важну улогу у истраживању веба. Долазни и одлазни линкови неке странице су више него корисни за проналажење садржаја на неку тему. Поред тога, структура ХТМЛ документа такође може да пружи квалитетне информације алгоритмима за истраживање текстова. ХТМЛ код омогућава означавање делова садржаја на основу њихове функционалности, другим речима, могуће је користити елементе који недвосмислено описују делове садржаја. Употреба одговарајућих елемената и прецизно дефинисање значења садржаја на вебу, а у циљу омогућавања његовог индиректног или директног машинског истраживања су основне идеје Семантичког веба [37].

Осим тога, прави „благослов“ за проналажење и извлачење информација из текстова са веб страница је постојање описа стилова [38]. Скоро свака веб страница

садржи описе стилова, помоћу којих се прави разлика између ХТМЛ елемената, који иначе имају исту ознаку, а да би ти елементи били другачије обликовани или стилизовани. То значи да неке ознаке могу да изгледају овако `<span class="green"></span>` или овако `<span class="red"></span>`. Приликом проналажења и извлачења информација из текстова са веб странице, лако је на основу класе раздвојити две различите ознаке из претходног примера, а затим извршити издвајање само текстова који су обојени зеленом бојом, без текстова који су обојени црвеном бојом или без неких других текстова [7]. Каскадни описи стилова се ослањају на ове атрибуте за идентификацију стилова и готово је гарантовано да ће ове класе и атрибути бити на бројним модерним веб сајтовима [38].

На веб страницама, поред главног садржаја постоји и већа количина неважних информација, као на пример рекламе. Приликом истраживања текстова на веб страницама, посебан изазов је уклањање таквог неважног садржаја.

3.4 Припрема текстова

Текстуални садржаји су креирани од стране људи и углавном су у неструктуираном облику. Да би се текст, као неструктуирани формат, претворио у структуирани и вишедимензиони облик, неопходно је да се **припреми** или **препроцесира** (ен. *Text preprocessing*). Овај процес је повезан за лингвистичким знањем из домена појединих језика. У овом документу ће више пажње бити посвећено припреми текстова који су на **српском језику**.

Уобичајене методе за претходну обраду текстова су:

- Извлачење текстуалних садржаја,
- Рашчлањивање или токенизација,
- Препроцесирање токена и
- Нормализација

Приликом припреме текстова за даље истраживање, потребно је наћи решење за разне изазове. На пример, у поступку аутоматског прикупљања текстова са Интернета, потребно је прикупити текст који се налази у документима на вебу, а затим га припремити за даље истраживање. У оваквим случајевима, необрађен садржај може да садржи елементе као што су ХТМЛ тагови, грешке у писању, двосмислене речи итд. Осим тога, документи, као што су веб странице, могу да садрже више блокова, од којих већину могу да чине рекламе или други садржај који тематски није повезан са основним

документом. Овакви садржаји такође треба да буду уклоњени правилном припремом текстова за даље истраживање. Из прикупљеног материјала је, дакле, потребно **извући текстуални садржај** који ће бити предмет даљег истраживања.

Након „чишћења“ и извлачења текстова, приступа се његовом **рашчлањивању**, односно претварању низа карактера у низ речи или **токена**. Свако помињање неке речи у документу се третира као посебан токен, при чему се појам „токен“ односи на низ знакова који се као недељива јединица третира у даљој обради. Овај процес се такође назива и **токенизација.**

Нису све речи једнако важне у аналитичким задацима. **Речи без веће садржајне вредности** (тзв. „стоп-речи“) представљају прилично екстреман случај веома честих речи, на једном крају спектра, које треба уклонити из разматрања. Један од поступака за **препроцесирање токена** подразумева уклањање речи без веће садржајне вредности, које обично нису дискриминативне за већину апликација за истраживање текстова, а додају само велику количину „шума“. Уобичајени предлози, везници, заменице и слично се сматрају речима без веће садржајне вредности. Често су доступни речници ових речи, за поједине језике. Осим поступка уклањања речи без веће садржајне вредности, у **препроцесирање токена** спадају и **свођење речи на основу, уклањање знакова интерпункције, свођење текстова на исту величину слова** итд.

Након препроцесирања токена, праве се репрезентације текстова у векторском простору која је ретко-поседнута мултидимензионална репрезентација која садржи фрекфенцију појављивања појединих речи. Након извлачења фрекфенције термина из колекције, ради се **нормализација**, тако да врло чести појмови добијају мање пондере или тежинске факторе. Овај тип нормализације се назива нормализацијом инверзне фрекфенције документа (ен. *inverse document frequency normalization*).

У поглављу са насловом „**Припрема текстова на српском језику за потребе даљег истраживања**“ ће детаљно бити описан поступак припреме текстова, са посебним освртом на рад са текстовима који су на српском језику.

3.5 Израчунавање сличности између текстова

Многе методе за истраживање текстова и претраживање информација, захтевају израчунавање сличности између парова докумената. Ово израчунавање је веома осетљиво на начин репрезентације документа. На пример, када се користи бинарна репрезентација, *Jaccard*-ов коефициент је ефективан начин израчунавања сличности, а

са друге стране, косинусна сличност је прикладна за случајеве у којима се располаже са фрекфенцијом термина [11]. Управо из разлога што се израчунавање сличности осетљиво на начин репрезентације текстова, овај поступак је коришћен за потребе процене утицаја различитих алата за припрему текстова на српском језику, на даље истраживање. У поглављу 4, са насловом „Припрема текстова на српском језику, за потребе даљег истраживања“ ће бити описано и примењено израчунавање сличности између правних текстова.

3.6 Сегментација текстова

Дугачки докуметни обично садрже више логичких или тематских целина. Откривање граница између целина истог текстуалног документа се назива **сегментација текста**. Циљ сегментације текста је да се документ подели у повезане логичке или тематске целине.

У ненадгледаној сегментацији текста, траже се само промене теме или границе између сегмената. У надгледаној сегментацији, траже се специфични типови сегмената, на пример сегмент политике или сегмент спорта у новинским чланцима.

Сегментација текстова може да се подели на језичку и тематску. Језичка сегментација одговара сегментацији у речи, реченице или пасусе, и често се заснива на интерпункцијским знацима и појавама које су специфичне за поједине језике. Са друге стране, тематска сегментација се заснива на семантичком садржају. У свим случајевима, јединица сегментације је мањи део документа.

Текстови закона, који су предмет истраживања које је описао у поглављу 5, су по правилу у интегралном облику. Унутрашња подела текстова закона има за циљ груписање материје, ради њеног систематизовања и лакше примене. Према тој подели, између осталог је дефинисано да је Члан основна класификациона и логичка јединица закона [3]. У поступку трансформације података, које је у наставку описано у потпоглављу 5.3, примењена је сегментација текстова на поједине чланове закона.

3.7 Извлачење информација из текстова

Проблеми, као што су проналажење и издвајање одређених језичких израза у правним документима и дефинисање модела структуре колекције правних докумената су у суштини проблеми извлачења информација из текстова. Аутори [34], термин “Извлачење информација” управо и описују као извлачење информација и релација

између њих, из збирке докумената. Ово је кључан корак у претварању неструктуираног текста у структуирану репрезентацију. Осим тога, многе друге примене истраживања текстова, као што су истраживање мишљења (ен. *opinion mining*) и откривање догађаја (ен. *event detection*), користе технике извлачења информација.

У својој најосновнијој форми, текст који се истражује је дефинисан као низ токена, при чему текст није означен својствима ових токена. Циљ извлачења информација је да се открију корисне информације из посматраног низа токена. Мало конкретније, израз „Извлачење информација“ (из текстова) се у [11] описује као:

1. Препознавање именованог ентитета у текстовима (Токени у тексту могу да се односе на именоване ентитете, као што су локације, особе, организације и сл.);
2. Извлачење веза - генерално следи после препознавања именованог ентитета и односи се на проналажење односа између различитих именованих ентитета.

3.7.1 Методе за препознавање именованог ентитета у текстовима

Именовани ентитет у тексту је низ речи које одговарају одређеном ентитету у стварном свету (тј. ентитету са именом). Препознавање именованих ентитета може да се користи за доменски специфичне послове, као што су извлачење правно релевантних информација из правних докумената или уговора, извлачење рачуноводствених информација из фактура, пореских, банковних и других докумената итд. Задаци попут описаних, захтевају релативно висок ниво разумевања језика и познавање домена који је предмет истраживања. Изазове за успешну примену ових метода у раду са правним документима представљају сложеност тих докумената и ниска толеранција на грешке, посебно када се ради са текстовима закона и других прописа.

Циљ препознавања ентитета у текстовима је да се закључи да ли је нека одређена реч или нека одређена група речи део неког именованог ентитета. Опредељење да се посматрана реч или група речи означи као именовани ентитет, зависи од својстава те речи или групе речи. Најчешће коришћени начини за одређивање својстава укључују проверу да ли се посматране речи појављују у одговарајућим листама или лексиконима, проверу облика речи, граматичких својства и контекста у коме се налазе речи или групе речи. За успешно препознавање и извлачење ентитета, важно је како се комбинују различите врсте провера у циљу тачнијег идентификовања именованих ентитета. Према [11], постоје две основне класе метода за препознавања именованих ентитета у текстовима:

- Прва класа метода, која се назива **методе засноване на правилима** (ен. *Rule-based methods*), користи збирку условних правила која се примењују на текст, како би се идентификовали могући ентитети. Збирке правила се обично праве коришћењем комбинације аутоматизованих и ручних подешавања ради извлачења информација.
- Друга класа метода, која се назива **статистичким методама учења**, користи класификационе моделе да би се вршило предвиђање да ли реч или група речи одговара именованом ентитету. Овај приступ користи скривене Марковљеве моделе, Марковљеве моделе максималне ентропије и Условне случајне области.

Методе за препознавање ентитета у тексту, засноване на правилима

Методе за препознавање ентитета у тексту, које су засноване на правилима, користе колекцију правила. Методе засноване на правилима функционишу тако што се сваки токен у тексту претвара у скуп својстава. Ова својства се у комбинацији са контекстом користе за извлачење ентитета. На пример, једно очигледно својство неког токена може да буде да ли је реч написана малим или великим почетним словом. Ово својство дакле помаже у дефинисању различитих образаца или правила. Процес издвајања својстава је важан аспект инжењерства својстава (ен. *feature engineering*) у методама заснованим на правилима. Скуп правила, која су пронађена у подацима, је следећег облика:

Контекстуални шаблон ⇒ *Акција*

Контекстуални шаблон на левој страни правила је комбинација услова који одговарају својствима која су придружена токенима. Према томе, ако низ знакова у тексту одговара неком обрасцу, каже се да је ту пронађено правило. Акција на десној страни може да одговара означавању те секвенце, као именованог ентитета. Генерално, може да одговара почетку уметања ентитетске ознаке на одређену позицију, крај ознаке ентитета или вишеструке ознаке. Најједноставнији и најчешћи случај је онај у коме је десна страна правила истовремено и ознака ентитета.

Природа леве стране правила може да варира у зависности од специфичног скупа правила, која се примењују. Типична својства, која су повезана са сваким токеном су:

- сама појава токена, као његово основно својство,
- својства проистекла из правописних правила,
- својства проистекла из граматичких правила,

- својства проистекла из речника и лексикона, који се користе за одређивање да ли токен припада неком одређеном типу именованог ентитета и
- својства проистекла из контекста у коме је нека реч или група речи употребљена.

Основно својство токена је његова појава у облику стринга. У неким случајевима, ово основно својство може да буде довољно информативно за извлачење ентитета.

Правописна правила, као што је употреба великих и малих слова, знаци интерпункције или избор специфичних правописних правила, могу да помогну у добијању својстава токена. На пример, ако се велико слово налази у средини реченице, онда то обично означава одговарајућу именицу.

Технике које се користе за рашчлањивање реченица на језичке изразе, фразе или комаде реченица, се називају технике плитког или површног рашчлањивања (ен. *Shallow parsing*) [34]. На тај начин се добијају информације о врстама речи, односно **граматичка својства** речи. Нека граматичка својства могу да одговарају низу који се састоји од више токена.

Бројни речници и лексикони су често прва локација која се узима у обзир приликом препознавања и извлачења ентитета из текстова. Лексикон је листа речи које су груписане по категоријама [34]. Они могу да садрже спискове познатих ентитета или могу да садрже спискове „помоћних речи“, као што су нпр. титуле или обележја компанија. Поред тога, речници могу чак и да идентификују да ли се токен појављује као део одређеног имена. У складу са тим, лексикони могу да се користе на два начина:

- Први је директна идентификација именованих ентитета. Ово је посебно корисно за категорије речи, које готово увек одговарају именованом ентитету, попут имена држава.
- Друга употреба лексикона је да се дефинише скуп помоћних речи, које могу да се користе за идентификовање именованих ентитета. Уобичајени лексикони помоћних речи укључују изразе који се користе приликом ословљавања (као што су господин, госпођа, госпођица и слично), титуле (академик, професор, доктор, наставник и сл.), идентификаторе улица (улица, ул., булевар,...), обележја компанија (Д.О.О., А.Д., К.Д. и сл.). „Помоћне речи“ могу да пруже информацију да би у њиховом окружењу требало да се налази именовани ентитет.

Међутим, у одређеним контекстима може да дође до грешака. На пример, реч „Дунав“, може да се односи на реку, али и на истоимену осигуравајућу компанију. Изазов

у препознавању именованих ентитета у текстовима, а на основу лексикона, је да се се одреди када би друга својства требало да имају већу важност од својстава која су додељена уз помоћ лексикона [34]. Дакле, за већину апликација, употреба само лексикона није довољна за препознавање именованих ентитета из текста и потребни су и други начини за одређивање својстава. Лексикони могу да пруже снажне доказе да је одређени језички израз именовани ентитет, али не помажу у анализи када се израз не налази у лексикону, када се израз користи на неуобичајен или на непознат начин или када се користи на више начина.

Претходно набројани приступи користе граматичка и језичка својства углавном у комбинацији са контекстом појединих речи или група речи, да би се идентификовали именовани ентитети. **Контекст** неке речи или групе речи укључује својства речи пре или после посматране речи или групе речи. У неким методама заснованим на правилима, текст се секвенцијално означава у фазама. У таквим случајевима, ознаке у ранијим фазама се користе као својства у условима за правила у каснијим фазама. Као додатак структуираних образаца који одговарају токену, лева страна неког правила може опционо да садржи шаблоне који одговарају контексту који претходи или који следи после ентитета. Примери два могућа правила су:“

(Токен=“Проф.“, Правопис=прво велико слово) ⇒ Лично име

(Правопис=прво велико слово, Токен =“ д.о.о.”) ⇒ Назив организације

У претходним примерима, прво правило одговара редоследу од два токена, при чему је први „Проф.“, а други почиње великим словом. Друго правило одговара низу од два токена од којих први почиње великим словом, а други је скраћеница „д.о.о.“. Доступни су многи корисни речници титула и назива компанија за конструкцију оваквих правила. Према томе, могу да се појаве облици као што су „класе из речника“ да би се описали овакви токени. Алтернативни скуп правила би могао да буде:

(Класа из речника=Титуле, Правопис=прво велико слово) ⇒ Лично име

(Правопис=прво велико слово, Класа из речника=Суфикс компаније) ⇒ Име организације

Треба напоменути да регуларни израз на левој страни правила може да буде прилично сложен и да постоје многе алтернативе. Имајући у виду велики број начина на које може да се направи правило подударања за исти израз, постоје значајни изазови ефикасности у прављењу система за препознавање ентитета у тексту, који је заснован на правилима.

Лако је видети да постоје сличности у системима који се користе за извлачење информација из текстова и који су засновани на правилима са онима који се користе за класификацију. Главна разлика је у томе што је структура правила често сложенија у извлачењу информација.

Статистичке методе учења за препознавање ентитета у текстовима

Статистички приступи препознавању ентитета се третирају као проблем означавања низа, са циљем да се пронађе највероватнији низ ознака, на основу улазне секвенце токена. У статистичке технике машинског учења спадају „Скривени Марковљеви модели“ (ен. *Hidden Markov Models*), „Марковљеви модели максималне ентропије“ (ен. *Maximum Entropy Markov Models*) и „Условна насумична поља“ (ен. *Conditional Random Fields*).

Претпоставка која стоји иза модела секвенци, као што су ***Скривени Марковљеви модели***, је да је текст изворно генерисан у паровима речи и њихових ознака, али да су ознаке изгубљене или „сакривене“ [34]. Дакле, извлачење ознака може да се посматра као покушај опоравка низа оригиналних ознака, које су повезане са текстом. Скривени Марковљеви модели према [11], пролазе кроз низ скривених стања, свако стање производи токен у датој секвенци текста и стања су зависна једна од других. Скривени Марковљеви модели покушавају да моделирају све могуће секвенце токена и њихових ознака. Међутим, у просечном корпусу, само ће мали подскуп од свих могућих секвенци да буде посматран, што значи да Скривени Марковљеви модели троше време на већи број секвенци које неће бити посматране.

За разлику од Скривених Марковљевих модела, који генеришу секвенце користећи прелазе између стања, **Марковљеви модели максималне ентропије** директно моделирају вероватноћу означавања на основу стања [39]. Код ових модела, свако изворно стање има експоненцијални модел који узима посматрана својства као улаз и даје расподелу по могућим наредним стањима. Ови експоненцијални модели се тренирају одговарајућом методом скалирања у оквиру максималне ентропије.

Условна насумична поља (ен. *Conditional Random Fields*) [40] су статистички модел за сегментирање и означавање секвенцијалних података, који нуди неколико предности у односу на Скривене Марковљеве моделе, а такође избегава ограничење Марковљених модела максималне ентропије. Основна разлика је што Марковљеви модели максималне ентропије користе експоненцијалне моделе по стању за условне

вероватноће следећих стања, с'обзиром на тренутно стање, док модел Условна насумична поља има јединствен експоненцијални модел за заједничку вероватноћу целог низа ознака, с'обзиром на посматрани низ. Због тога пондери или тежински фактори различитих својстава у различитим стањима могу да се међусобно размењују. Модел „Условна насумична поља“ (ен. *Conditional Random Fields*) је дискриминишућа алтернатива Скривеним Марковљевим моделима, осмишљена тако да моделира условну вероватноћу секвенце ознака, избегавајући при том процену пуне вероватноће. Од статистичких модела, Условна насумична поља су међу најуспешнијима за извлачење информација.

3.7.2 Извлачење веза из текстова

Извлачење веза из текстова се обавља на основу извлачења ентитета. Другим речима, када се из текста извуку ентитети, онда и везе између њих могу да буду истражене. Проблем извлачења веза из текстова је дефинисан на следећи начин: Ако је дат фиксни скуп релација R, циљ је да се идентификују све појаве ових веза, где су ентитети већ означени, али недостају везе између њих. У поставкама надгледаног машинског учења, постоји корпус за обуку, у коме су идентификовани и ентитети и односи између појединих појава ентитета. Ако постоји такав корпус, на тестном документу, који је без икаквих ознака, прво се издвајају ентитети у тестном документу, а затим и везе између ентитета. За сваки пар ентитета, који се помиње у реченици, задатак је да се утврди да ли постоји веза између њих из скупа релација R. Такође, претпоставља се да скуп R садржи посебан тип веза који се назива „Null“, а који се примењује у случајевима када се пар ентитета појављује у истој реченици, али не постоји веза између њих.

Извлачење веза као класификација

Проблем извлачења веза може да се представи као проблем класификације. Ако се извлаче везе само између појава ентитета у оквиру реченице, онда се могу издвојити парови појава ентитета у истој реченици, и у подацима за обуку и у подацима за тестирање. Дакле, кључ је креирање једне инстанце података за сваки пар ентитета унутар реченице. Таква инстанца је такође означена као тип везе за реченице у документима за обуку, а није означена у документима за тестирање.

Информације које су потребне за доношење закључака о постојању веза између ентитета су сакривене у вокабулару и граматичкој структури реченице у којој се пар ентитета појављује. На пример, дата је следећа реченица за тестирање, у којој су ентитети обележени, али везе нису обележене:

„Електронски факултет се налази у Нишу“.

У претходној реченици су „Електронски факултет“ и „Ниш“ два именована ентитета, а може се закључити да се један ентитет типа „институција“ налази у/на ентитету типа „локација“. Подаци за обуку се користе тако да би се сазнала чињеница да језички израз „налази у“ пружа корисне назнаке за следећу везу:

НалазиУ(*Електронски факултет*, *Ниш*)

Другим речима, потребно је издвојити изразе из различитих региона реченице (нпр. токени између пара ентитета) да би се донели закључци о вези између ентитета. Процес машинског учења за издвајање веза може да буде имплементиран као издвајање одговарајућих језичких израза из реченице која садржи пар појава ентитета.

За парове ентитета, који се помињу у истој реченици, а за које у подацима за обуку веза између њих није означена, креира се негативна инстанца за обуку и користи се ознака „Null“. Пример реченице у којој су означена три ентитета:

„*Универзитет може, на предлог факултета, да додели звање професора емеритуса*“.

У претходној реченици се налазе три ентитета, два ту типа „институција“, а један је типа „звање“. Из једне овакве реченице, за сваки пар ентитета могу да се извуку три инстанце за обуку. Прве две инстанце су:

ДоделиЗвање(Универзитет, Професор емеритус)

НаПредлог(Факултет, Универзитет)

Трећа инстанца за обуку може да буде означена и као „Null“, на пример између ентитета „Факултет“ и „Професор емеритус“. Такав пример током обуке може да буде користан као негативна инстанца типа везе, у односу на типове веза које су предмет интересовања.

У примерима, када се истражују везе између ентитета типа „особа“, у подацима за обуку везе могу да се дефинишу као нпр. „брат“, „сестра“, „жена“, „муж“, како би се на одговарајући начин обележиле инстанце веза за обуку. Међутим, овде се може приметити да у српском језику, типови веза између особа зависе од смера везе између

тих особа: „брат“-„сестра“, „муж“-„жена“, итд., што значи да се између три ентитета типа „особа“, може извући чак шест инстанци за обуку.

Алтернативни приступ за извлачење веза из текстова је да се користе „функције сличности кернела“ (ен. *kernel similarity functions*), које дефинишу сличности између парова инстанци (нпр. парова обележених веза). Кернел методе су индиректан начин реализовања „инжењерства својстава“ (ен. *feature engineering*) и о њима ће бити више речи у наставку текста. Међутим, у случају експлицитног инжењерства својстава (ен. *explicit feature engineering*), једна предност је да може да се користи широка палета метода за класификацију. У ствари, најраније технике су биле методе засноване на правилима (ен. *rule-based methods*), које су специјализовани типови класификатора. У наставку ће бити речи о ова два различита начина извођења инжењерства својстава.

Предвиђање веза из текстова помоћу експлицитног инжењерства својстава

Својства могу да буду извучена из ентитета или изван њега. Својства која су извучена из ентитета се називају **ентитетским својствима** (ен. *entity features*). Међутим, својства који су извучена из региона реченице, који окружује ентитете, или оног који се налази између два ентитета, су корисна за извођење закључака. Таква својства се називају **контекстним својствима** (ен. *contextual features*).

Ентитетска својства и контекстна својства се користе на мало другачији начин током процеса инжењерства својстава. Међутим, у оба случаја слична својства су извучена из појединих токена, који се не разликују много од оних који се користе у извлачењу ентитета. Ова својства (која су повезана са појединим токенима) су [11]:

1. основни облици речи (ен. *surface tokens*),
2. ознаке врсте речи (ен. *parts-of-speech tags*),
3. својства која су извучена из синтаксног стабла реченице (ен. *constituency-based parse-tree structure*).

Основни облици речи: ако се посматра реченица *„Електронски факултет се налази у Нишу“*, у овој реченици реч „налази“ и језички израз „налази у“ дају корисне информације о вези између два ентитета, једног типа „институција“ и другог типа „локација“. У многим случајевима, подаци за обуку могу да садрже довољан број таквих појављивања ових случајева који много говоре о везама између ентитета.

Ознаке врста речи: постоји неколико врста речи, које у реченицама могу да се појављују, као нпр. именице, глаголи или друге врсте речи. Постоје и случајеви када иста

реч може да се појави или као именица или као глагол. Примери таквих реченица су: „Мој тата је купио радио“ и „Мој тата је радио у предузећу“. У претходној реченици, чињеница је да се реч „радио“ користи као глагол и као таква је корисна за доношење закључака о вези између ентитета „особа“ и ентитета „институција“. Међутим, иста реч може да се употреби и као именица, па приликом доношења закључака о везама између ентитета треба бити обазрив и посматрати шири контекст.

Својства која су извучена из структуре синтаксног стабла: у многим случајевима структура реченица може да буде компликована, што може да проузрокује праве изазове приликом доношења закључака. На пример, реченица може да садржи више од два именована ентитета и да постоји нека двосмисленост у одлучивању који парови ентитета су ближе повезани, или на који начин треба користити индиције које су извучене из реченице. У таквим случајевима, веома је корисна структура синтаксног стабла. Као пример, посматраћемо следећу реченицу: „*Универзитет може, на предлог факултета, да додели звање професора емеритуса*“. У овој реченици, језички израз „*факултета*“ се налази близу израза „*звање професора емеритуса*“ и лако је да алгоритам машинског учења користи токен „*да додели*“, па да направи погрешну претпоставку. Међутим, у синтаксном стаблу“ ће читав језики израз „*на предлог факултета*“ бити у потпуно другом подстаблу. Међутим, синтаксна стабла су прилично „скупа“ за изградњу. Зато се користе поједностављени структурални прикази, који се називају графови зависности (ен. *dependency graphs*) [11]. Овај тип извлачења својстава спада у општи приступ метода заснованих на графу.

Понекад постоје ограничења да се поједини облици речи користе за извлачење веза између ентитета. Често је много корисније за извлачење веза када се користи комбинација својстава из реченице.

Предвиђање веза из текстова помоћу имплицитног инжењерства својстава – Метода кернела

Појам „Кернел“ се односи на начин на који се трансформишу подаци у вишу димензију. При препознавању образаца у подацима, пожељно је да алгоритми ефикасно анализирају и класификују податке. Ако је граница између различитих класа превише закривљена, алгоритму ће требати много времена да конвергира, а чак и када конвергира, можда граница није оптимална.

Због тога је потребно инстанце трансформисати у вишу димензију, тако да их је након трансформације лако раздвојити. На пример, инстанце које се налазе у две димензије се пројектују у три димензије, на начин да лако може да се нацрта једноставна раван за раздвајање скупа инстанци. Инстанцама су додељене различите вредности за трећу димензију, а њихова пројекција у две димензије даје почетни скуп података. То се назива „**трик кернела**“ [41]. Трик кернела је математички алат који може да се примени на било који алгоритам који искључиво зависи од скаларног производа два вектора. Да бисмо израчунали скаларне производе два вектора у вишој димензији, заправо није потребно да се подаци пројектују у вишу димензију. Може да се искористи кернел функција да се директно израчуна скаларни производ, користећи векторе ниже димензије.

Кључна поента овог приступа је стварање одговарајуће дефиниције функције сличности између пара реченица и парова ентитета унутар њих. У конкретном случају извлачења информација, ентитетски аргументи унутар реченица морају да се користе у израчунавању сличности, како би се осигурало да је функција сличности довољно дискриминишућа у односу на везу која се истражује. Међутим, за остале апликације које се баве обрадом природног језика, а које не користе ентитете, могу се конструисати мање модификације ових функција сличности.

3.8 Класификација текстова

Класификација текстова (ен. *Text classification*) је подела текстова у унапред дефинисане групе или класе, које се идентификују преко њихових ознака. Типичан пример су апликације за класификацију е-порука у две групе, непожељне (или спам) и оне које нису непожељне. Код класификовања, скуп података за обуку је унапред припремљен, са примерима текстова који припадају одређеним класама. Затим се врши тестирање на необележеном скупу текстова, за који се жели да ти текстови буду класификовани у неку од унапред дефинисаних класа. Процес обуке модела на основу података за обуку, а затим примена тог модела на податке за тестирање се назива **генерализација**. Основни принцип је да се искуства из (специфичних) примера за обуку, са познатим ознакама, „генерализују“ на произвољним подацима за тестирање, са непознатим ознакама. На пример, модел би могао да сазна да је реч „посланик“ повезана са ознаком „скупштина“ и могао би да искористи ову чињеницу да документима за тестирање, који садрже ову реч, додели ознаку „скупштина“.

И класификовање и кластеризација врше раздвајање података у групе. Међутим, применом класификовања је раздвајање високо контролисано, са унапред осмишљеном представом груписања, које је дефинисано преко података за обуку. Подаци за обуку пружају смернице алгоритму и то је разлог због кога се класификација назива надгледано машинско учење.

У истраживању које је у овом раду описано у поглављу 6, а које се бави прављењем модела за предвиђање постојања језичких израза за повезивање у текстовима нових закона и других прописа, у суштини је вршена класификација текстова употребом неуронских мрежа.

4 Припрема текстова на српском језику за потребе даљег истраживања

У овом поглављу ће бити објашњена важност и утицај припреме текстова на даље истраживање и детаљније ће бити описан читав овај поступак по фазама. Поред тога, овде ће бити указано на могуће проблеме који могу да настану услед непажљиве примене неких од алата за припрему текстова, а за потребе машинског учења, при чему је посебна пажња посвећена особеностима и изазовима који се односе на припрему и рад са текстовима који су на српском језику.

Текстуални документи, који су коришћени у истраживању су статути високошколских установа у Републици Србији. Према Члану 56, Закона о високом образовању [42], Статут је основни општи акт високошколске установе којим се уређује организација установе, начин рада, управљање и руковођење, као и друга питања од значаја за обављање делатности и рад високошколске установе, у складу са законом. Поред тога, овај Закон у Члану 6 дефинише да аутономија универзитета и других високошколских установа подразумева, између осталог, и право на доношење статута. То значи да су све високошколске установе у Републици Србији обавезне да имају овај пропис и да садржај тог прописа самостално уређују. Другим речима, статути уређују исте теме, али за различите установе, креирани су и усвојени од стране различитих установа и као такви су погодни за истраживање сличности између текстова тих докумената, уз употребу одговарајућих методологија и алата.

Због сличности са српским језиком, имплементација која се развијена за хрватски језик, такође може да буде корисна [17]. Циљ истраживања које ће бити описано у овом поглављу је анализа метода и алата за припрему текстова на српском језику, на примеру колекције текстуалних докумената. Тачније, биће урађена анализа утицаја методологија и алата за припрему текстова за потребе даљег истраживања, на смањење вектора за мултидимензионално представљање посматраних текстова и на израчунавање сличности између посматраних докумената. Поступак који је примењен се састоји од следећих фаза:

- Прикупљање текстуалних докумената за потребе овог рада
- Припрема текстова – препроцесирање
- Представљање текстова у векторском облику
- Анализа утицаја методологија и алата за припрему текстова на величину вектора за мултидимензионално представљање текстова

- Израчунавање сличности између посматраних текстова
- Анализа утицаја методологија и алата за припрему текстова, на израчунавање сличности између текстова

4.1 Прикупљање докумената

Према Правилнику о стандардима и поступку за акредитацију високошколских установа [43], прописан је посебан стандард који се односи на „Јавност у раду“. Овај стандард предвиђа да високошколска установа треба да има своју веб презентацију на Интернету и да ту објављује различите информације и документа из свог делокруга. Захваљујући томе, процес увида у статуте високошколских установа у Србији је олакшан и за потребе овог истраживања, прикупљени су ови статути са Интернета.

Национално тело за акредитацију и проверу квалитета у високом образовању Србије – НАТ (некада: Комисија за акредитацију и проверу квалитета – КАПК), у свом „Водичу за студенте“ редовно ажурира информације о акредитованим студијским програмима у Републици Србији [44]. Осим информација о акредитованим студијским програмима, у овом водичу се налазе и основне информације о акредитованим високошколским установама. Из тог документа су прикупљене информације о свим акредитованим високошколским установама, адресама њихових веб презентација и са тих адреса је током пролећа 2018. године прикупљено 180 текстова статута. Треба напоменути да је број високошколских установа у Србији, у тренутку када су прикупљани подаци био 222. Разлика у броју установа и у броју прикупљених текстова статута је настала из следећих разлога: постоји одређени број акредитованих високошколских установа које су без својства правног лица и као такве немају своје статуте, а неколико установа није објавило статуте на својим веб сајтовима. Осим тога, у појединачним случајевима текстови статута који су објављени на Интернету су били у ПДФ формату у коме је активирана заштита приступа (ен. *access-right protected PDF documents*) и као такви нису могли да буду коришћени у овом истраживању.

4.2 Поступак припреме текстова за потребе даљег истраживања – препроцесирање текстова на српском језику

Ефикасно истраживање текстова (ен. *text mining*) се заснива на методама за припрему текстова (за препроцесирање текстова). Заправо, истраживање текстова је толико зависно од различитих техника за припрему текстова, да се може рећи да је у

одређеном степену и дефинисано овим детаљним припремама [45]. Скуп карактера неког текста се по правилу састоји од израза коју су састављени од различитих појмова из речника, при чему се под речником мисли на базу која садржи скуп свих речи које се користе у неком језику. Ови изрази се најчешће праве уз употребу вишеструких временских и других граматичких промена речи. Осим тога, у текстовима се често могу пронаћи речи истог облика, али различитог значења (хомоними), речи различитог облика али истог или сличног значења (синоними), речи које су написане великим словом, итд. Све овакве појаве треба да се обраде у процесу претварања текста у основне појмове или термине, којима се затим одређује фрекфенција појављивања у тексту. Због тога се и каже да пре-процесирање текста (ен. *pre-processing*) има за циљ стварање „ретко поседнуте“ (ен. *sparse*) вишедимензионалне репрезентације текста [11]. Појам „ретко поседнута“ се односи на димензионалност вектора, зато што се у текстовима појављује далеко мањи број речи и израза, него што је број појављивања речи и израза у речнику.

У већини поступака за препроцесирање текстова се користи поступак нормализације [34]. Нормализације текста који је на српском језику, због особености овог језика [12], представља прави изазов [13].

За потребе овог рада, поступак припреме текстова је спроведен у следећим фазама:

- Извлачење сировог текста из докумената (ен. *Raw Text Extraction*)
- Уклањање цртица и осталих знакова интерпункције
- Токенизација – подела текста на саставне делове
- Претварање текста у одговарајућу величину слова
- Избацивање речи без веће садржајне вредности (ен. *Stop-words*)
- Извлачење морфолошког корена речи - *Stemming*

4.2.1 Извлачење сировог текста

Статути који су прикупљани су били у облику текстуалних докумената. Први корак у припреми текстуалних докумената за даљу анализу је претварање сировог текста (ен. *raw text*) у низ карактера (ен. *character sequence*). Текстуална репрезентација неког језика је низ карактера, али се веома често текст појављује у бинарним форматима, као што су *Microsoft Word* документи или докуметни у ПДФ формату (*Portable Document Format, PDF*). Како није прописан формат у коме се статути високошколских установа у Републици Србији објављују, тако је овим установама остављено да саме донесу одлуку

о томе. На основу докумената који су прикупљени за потребе овог рада, 94.4% статута је објављено у ПДФ формату, а 5.6% статута је објављено у *.DOC или *.DOCX формату. То значи да је потребно да се прикупљени документи, који су скуп бајтова, претворе у низ карактера.

За потербе претварања бинарних докумената у текст, према [11] фактори који могу на то да утичу су:

1. Поједини текстуални документи могу да буду представљени одређеном врстом кодирања, у зависности од формата докумената, као што су *Microsoft Word* документи, документи у *PDF* формату или *ZIP* датотеке;
2. Језик документа дефинише свој скуп знакова за кодирање.

Поједини системи за кодирање су веома осетљиви на то који скуп карактера се користи и не могу сви системи за кодирање да се подједнако добро користе за манипулацију са свим скуповима карактера. За документ који је написан на неком од језика, нпр. на српском језику, се користи различит скуп карактера од документа који је написан на неком другом језику, нпр. на енглеском језику. За енглески језик, као и за многе друге европске језике, се користи латински скуп карактера, који се скраћено означава као *ASCII* скуп карактера. *Unicode Consortium* је направио стандардни скуп карактера, назван ***Unicode***, у коме је сваки карактер представљен преко јединственог идентификатора [11]. Чак, готово сви симболи који су нам познати из различитих језика (укључујући математичке симболе и многе древне карактере) могу да буду представљени као *Unicode*. Ово је и разлог због кога је *Unicode* постао уобичајени стандард за представљање свих језика. ***UTF-8*** се често користи као уобичајен на многим системима, а посебно је погодан и за *ASCII* скуп карактера. Међутим, иако је могуће користити *UTF-8* за кодирање практично било ког језика (што је и доминантан стандард), многи језици су представљени и другим кодовима, Врста кода који је употребљен зависи од језика, од онога ко је направио документ и од платформе на којој се документ налази. У многим случајевима, мета-подаци документа садрже корисне информације о врсти његовог кодирања, без потребе да се то закључује испитивањем садржаја документа. Скуп докумената који су предмет овог рада, писани су на српском језику, што значи да се је за креирање ових текстова употребљен скуп карактера *UTF-8*.

За потребе овог рада, текстови су за даљу анализу припремљени тако што су претворени у „обичан текст“ (ен. *plain text*). На тај начин су занемарена било каква кодирања, форматирања и уређивања текста. Добијен је скуп карактера у формату *.TXT,

при чему је текст једног документа сачуван у једној датотеци уз употребу *UTF-8* скупа карактера.

4.2.2 Цртице и остали знаци интерпункције

У поступку препроцесирања текстова, треба посебну пажњу посветити раду са знацима интерпункције, а посебно цртицама и њиховој улози. У неким случајевима ови знаци дефинишу границе између речи, а у неким случајевима су саставни део израза [11]. Осим тога, треба предвидети и могућност да су приликом прелома текста коришћене цртице када је реч дељена на слогове, при чему је део речи у једном реду текста, други део речи у другом делу текста (ен. *Hyphens*). У овом случају би разбијање довело до промене семантичког значења.

У зависности од врсте истраживања и потребе да се води рачуна о семантичком значењу, изрази који су састављени од више речи и који у себи садрже цртице или друге интерпункцијске знаке, могу, али и не морају, да буду подељени у засебне речи за потребе даљег истраживања текста. Како предмет овог рада није семантичко значење, нити су посматрани текстови дељени на слогове, у потпупку припреме текстова су уклоњени сви интерпункцијски знаци, што значи да су и сви изрази који су били састављени од више речи и који су у себи садржавали овакве знаке, подељени у засебне речи. На тај начин је број карактера у текстовима, у просеку смањен за 4.2% (просечан број карактера у текстовима је био 110692.7, а након уклањања знакова интерпункције, просечан број карактера је био 106051.5).

Разлика између улазног текста и текста са уклоњеним знацима интерпункције је дат у следећој табели:

Табела 1 Припрема текста - Уклањање знакова интерпункције, пример

Улазни текст:
С Т А Т У Т ЕЛЕКТРОНСКОГ ФАКУЛТЕТА У НИШУ I ОПШТЕ ОДРЕДБЕ Предмет уређивања Члан 1. Овим Статутом уређује се делатност Електронског факултета у Нишу (у даљем тексту: Факултет), његова организација, управљање, начин финансирања и друга питања од значаја за рад Факултета, у складу са Законом о високом образовању (у даљем тексту: Закон) и Статутом Универзитета у Нишу.
Излазни текст:
С Т А Т У Т ЕЛЕКТРОНСКОГ ФАКУЛТЕТА У НИШУ I ОПШТЕ ОДРЕДБЕ Предмет уређивања Члан 1 Овим Статутом уређује се делатност Електронског факултета у Нишу у даљем тексту Факултет његова организација управљање начин финансирања и

друга питања од значаја за рад Факултета у складу са Законом о високом образовању у даљем тексту Закон и Статутом Универзитета у Нишу

4.2.3 Токенизација

Пре било какве даље обраде текста, континуални скуп карактера је потребно да се подели у саставне делове од којих се текст састоји. Документи се могу поделити у поглавља, одељке, пасусе, реченице, речи, па чак и слогове и слова. Приступ који се најчешће користи у системима за истраживање текста је токенизација [45]. „**Токен**“ је непрекидни низ карактера који има семантичко значење. Токени могу да се понављају и није извршена никаква њихова додатна обрада (као што је свођење речи на основни облик) [11]. Процес којим се текст дели на саставне јединице или токене, од којих већина одговара речима из језика на коме је написан посматрани текст, се назива „**токенизација**“ [45]. Решавање изазовних проблема из перспективе одлучивања где су границе речи у неком тексту је саставни део овог процеса [11]. Не постоји јединствени начин за најбоље обављање токенизације. Када текст читају људи, они „обављају токенизацију“ прецизно и без много размишљања, али се испоставља да је овај задатак много двосмисленији када треба да га обради рачунарски програм. То опет значи да различити програми за токенизацију стварају нешто другачију сегментацију и због тога као главно правило које треба да се користи је да се исте апликације за токенизацију користе доследно, на свим посматраним текстовима у неком истраживању [11]. За потребе токенизације у овом раду, употребљена је библиотека отвореног кода „*Natural Language Toolkit (NLTK)*“ [46], са циљем да када се токени извуку из збирке докумената, да они даље могу да се трансформишу, да би се на крају добили основни појмови са одређеним фрекфенцијама појављивања.

Разлика између улазног текста и текста након токенизације је дат у следећој табели:

Табела 2 Припрема текста - Токенизација, пример

Улазни текст:
С Т А Т У Т ЕЛЕКТРОНСКОГ ФАКУЛТЕТА У НИШУ I ОПШТЕ ОДРЕДБЕ Предмет уређивања Члан 1 Овим Статутом уређује се делатност Електронског факултета у Нишу у даљем тексту Факултет његова организација управљање начин финансирања и друга питања од значаја за рад Факултета у складу са Законом о високом образовању у даљем тексту Закон и Статутом Универзитета у Нишу
Излазни текст:

```
['С', 'Т', 'А', 'Т', 'У', 'Т', 'ЕЛЕКТРОНСКОГ', 'ФАКУЛТЕТА',
'У', 'НИШУ', 'I', 'ОПШТЕ', 'ОДРЕДБЕ', 'Предмет', 'уређивања',
'Члан', '1', 'Овим', 'Статутом', 'уређује', 'се',
'делатност', 'Електронског', 'факултета', 'у', 'Нишу', 'у',
'даљем', 'тексту', 'Факултет', 'његова', 'организација',
'управљање', 'начин', 'финансирања', 'и', 'друга', 'питања',
'од', 'значаја', 'за', 'рад', 'Факултета', 'у', 'складу',
'са', 'Законом', 'о', 'високом', 'образовању', 'у', 'даљем',
'тексту', 'Закон', 'и', 'Статутом', 'Универзитета', 'у',
'Нишу']
```

4.2.4 Претварање текста у одговарајућу величину слова

Употреба великих и малих слова често дефинише семантичку интерпретацију, што је релевантно за послове истраживања текста. Велика слова у тексту се користе из различитих разлога, као што су започињање реченице, као део наслова или због тога што се користе властите именице. Цео процес претварања карактера неког текста у одговарајућу величину се назива „*truecasing*“ [47]. Међутим, постоје ограничења у примени поступка промене величине слова у тексту. Једно од ограничења је двосмисленост која може да настане у случајевима када различита величина слова даје различито значење истим речима, а постоје и други примери. Ипак, у многим случајевима је могуће користити овај поступак. Иако поступак није савршен, једноставност његове примене омогућава ефикасност у припреми и обради текстова. У овом раду употребљен је алгоритам за нормализацију величине слова у тексту (ен. *case normalization*) који претвара цео текст у мала слова, како би се избегао проблем различиотог тумачења истог текста, уколико је написан различитом комбинацијом великих и малих слова.

Разлика између улазног текста и текста који је претворен у мала слова је дат у следећој табели:

Табела 3 Припрема текста - Претварање текста у одговарајућу величину слова, пример

Улазни текст:
['С', 'Т', 'А', 'Т', 'У', 'Т', 'ЕЛЕКТРОНСКОГ', 'ФАКУЛТЕТА', 'У', 'НИШУ', 'I', 'ОПШТЕ', 'ОДРЕДБЕ', 'Предмет', 'уређивања', 'Члан', '1', 'Овим', 'Статутом', 'уређује', 'се', 'делатност', 'Електронског', 'факултета', 'у', 'Нишу', 'у', 'даљем', 'тексту', 'Факултет', 'његова', 'организација', 'управљање', 'начин', 'финансирања', 'и', 'друга', 'питања', 'од', 'значаја', 'за', 'рад', 'Факултета', 'у', 'складу', 'са', 'Законом', 'о', 'високом', 'образовању', 'у', 'даљем', 'тексту', 'Закон', 'и', 'Статутом', 'Универзитета', 'у', 'Нишу']

Излазни текст:

```
['с', 'т', 'а', 'т', 'у', 'т', 'електронског', 'факултета',
'у', 'нишу', 'і', 'опште', 'одредбе', 'предмет', 'уређивања',
'члан', '1', 'овим', 'статутом', 'уређује', 'се',
'делатност', 'електронског', 'факултета', 'у', 'нишу', 'у',
'даљем', 'тексту', 'факултет', 'његова', 'организација',
'управљање', 'начин', 'финансирања', 'и', 'друга', 'питања',
'од', 'значаја', 'за', 'рад', 'факултета', 'у', 'складу',
'са', 'законом', 'о', 'високом', 'образовању', 'у', 'даљем',
'тексту', 'закон', 'и', 'статутом', 'универзитета', 'у',
'нишу']
```

4.2.5 Избацивање речи без веће садржајне вредности

Речи без веће садржајне вредности (стоп-речи, ен. *stop-words*) су уобичајене речи у сваком језику које не доприносе различитости садржаја. На пример, у задатку класификовања чланака према областима то су речи које се са приближно истом фрекфенцијом појављују нпр. у чланцима о спорту и нпр. у чланцима о политици. Због тога је логично да се уклоне такве речи које лоше утичу на разликовање између текстова. Уобичајено се користе следеће стратегије за уклањање оваквих речи.

1. Сви предлози, прилози и везници су „стоп-речи“. Понекад се и заменице сматрају „стоп-речима“;
2. Доступни су и користе се речници „стоп-речи“ за поједине језике;
3. Могу да се идентификују чести токени у било којој одређеној локацији и може се поставити граница на некој фрекфенцији, да би се уклониле „стоп-речи“.

Уклањање „стоп-речи“ је чврста варијанта мекшег приступа са пондерисањем честих речи приликом инверзне нормализације фрекфенције документа (ен. *inverse document frequency normalization*). У неким случајевима постоји губитак информација повезаних са чврстим уклањањем „стоп-речи“. Због тога, многи системи за претраживање (ен. *search*) и рударење (ен. *mining*) не уклањају „стоп-речи“, већ се једноставно ослањају на приступ смањења тежине честих речи [11].

За потребе овог рада, направљена је сопствена колекција која се састоји од 288 речи без веће садржајне вредности написане ћириличним писмом на српском језику и те речи су уклањане из посматраних текстова за потребе даљег истраживања.

Разлика између улазног текста и текста из кога су избачене речи без веће садржајне вредности је дат у следећој табели:

Табела 4 Припрема текста - Избацивање речи без веће садржајне вредности, пример

Улазни текст:
['с', 'т', 'а', 'т', 'у', 'т', 'електронског', 'факултета', 'у', 'нишу', 'і', 'опште', 'одредбе', 'предмет', 'уређивања', 'члан', '1', 'овим', 'статутом', 'уређује', 'се', 'делатност', 'електронског', 'факултета', 'у', 'нишу', 'у', 'даљем', 'тексту', 'факултет', 'његова', 'организација', 'управљање', 'начин', 'финансирања', 'и', 'друга', 'питања', 'од', 'значаја', 'за', 'рад', 'факултета', 'у', 'складу', 'са', 'законом', 'о', 'високом', 'образовању', 'у', 'даљем', 'тексту', 'закон', 'и', 'статутом', 'универзитета', 'у', 'нишу']
Излазни текст:
['т', 'т', 'т', 'електронског', 'факултета', 'нишу', 'і', 'опште', 'одредбе', 'предмет', 'уређивања', 'члан', '1', 'овим', 'статутом', 'уређује', 'делатност', 'електронског', 'факултета', 'нишу', 'даљем', 'тексту', 'факултет', 'организација', 'управљање', 'начин', 'финансирања', 'друга', 'питања', 'значаја', 'рад', 'факултета', 'складу', 'законом', 'високом', 'образовању', 'даљем', 'тексту', 'закон', 'статутом', 'универзитета', 'нишу']

Уколико има потребе, из текста могу да се избаце и речи које су краће од неког унапред задатог броја слова. Разлика између улазног текста и текста из кога су уклоњене све речи које су краће од 5 слова је дат у следећој табели:

Табела 5 Припрема текста - Избацивање речи које су краће од 5 слова, пример

Улазни текст:
['т', 'т', 'т', 'електронског', 'факултета', 'нишу', 'і', 'опште', 'одредбе', 'предмет', 'уређивања', 'члан', '1', 'овим', 'статутом', 'уређује', 'делатност', 'електронског', 'факултета', 'нишу', 'даљем', 'тексту', 'факултет', 'организација', 'управљање', 'начин', 'финансирања', 'друга', 'питања', 'значаја', 'рад', 'факултета', 'складу', 'законом', 'високом', 'образовању', 'даљем', 'тексту', 'закон', 'статутом', 'универзитета', 'нишу']
Излазни текст:
['електронског', 'факултета', 'опште', 'одредбе', 'предмет', 'уређивања', 'статутом', 'уређује', 'делатност', 'електронског', 'факултета', 'даљем', 'тексту', 'факултет', 'организација', 'управљање', 'начин', 'финансирања', 'друга', 'питања', 'значаја', 'факултета', 'складу', 'законом', 'високом', 'образовању', 'даљем', 'тексту', 'закон', 'статутом', 'универзитета']

4.2.6 Консолидација заснована на коришћењу

Појам „консолидације засноване на коришћењу“ (ен. *Usage-Based Consolidation*) је поступак који је сасвим сличан *stemming*-у, осим што је једноставнији процес и примењује се током токенизације из употребу „lookup“ табела [11]. Основна идеја је да се мале варијације истог токена често односе на исту реч. Ове варијације најчешће настају у зависности од писца, аутора текста, географског подручја на коме је текст настао, употребе наречја и слично. У свим таквим случајевима је важно да ове варијације у тексту буду консолидоване у један термин. На пример, може у пракси да се направи табела или друга структура података о свим могућим варијацијама токена са њиховим стандардизованим облицима. Имајућу у виду да је предмет овог рада збирка докумената правно-формалне природе, то значи да је текст који је коришћен у посматраним документима формалан, без малопре набројаних варијација и због тога није било потребе да се на ову збирку докумената примењује консолидација заснована на коришћењу.

4.2.7 Морфолошка нормализација

Морфолофија је део науке о језику, која проучава врсте речи и облике тих речи. У српском језику постоји 10 врста речи [48], које се деле на:

- Променљиве врсте речи (имају деклинацију и конјугацију), у које спадају
    - Именице,
    - Заменице,
    - Придеви,
    - Глаголи и
    - Бројеви.
- Непроменљиве врсте речи (не мењају свој облик), у које спадају
    - Прилози,
    - Предлози,
    - Везници,
    - Узвици и
    - Речце.

Деклинација је промена речи по падежима, а речи које се у српском језику мењају по падежима су именице, заменице, придеви и бројеви. Глаголи су врста речи које имају *конјугацију*, односно, мењају се по лицу, броју, времену, роду...

Дакле, морфологија језика подразумева да речи могу да имају више облика. Иако немају потпуно исто значење, облици речи су семантички веома блиски са осталим облицима речи у тексту [49]. Да би се ојачао овај семантички однос између различитих облика речи, може да се примени морфолошка нормализација. **Морфолошка нормализација** је спајање различитих морфолошких варијација појма у исти базни облик чија је улога да смање величину вокабулара и на тај начин смање и поседнутост вектора текстуалих података, што класификаторима олакшава тачан модел утицаја сваке речи или израза [16]. Два уобичајена поступка нормализације су „Стеминг“ (ен. *Stemming*) и „Лематизација“ (ен. *Lemmatization*).

„*Stemming*“ је процес уједињавања сродних речи, које имају исти корен [11], и у било коме посупку истраживања текста, „*stemming*“ је процес коме треба посветити посебну пажњу. То је процес чија примена искључиво зависи од језика на коме су написани текстови који су предмет истраживања и самим тим утицај овог процеса на даље истраживање зависи од језика. На пример, текстуални документ може да садржи исту реч у једнини или у множини, у различитим временима или неке друге варијације те речи. У таквим случајевима има смисла да се све те варијације неке речи уједине у једну реч, јер промене речи не мењају њено семантичко значење са тачке гледишта истаживања података.

Уобичајено, „*stemming*“ се односи на процес извлачења морфолошког корена речи, а различите хеуристике се користе за постизање тог циља. Уобичајене технике су [11]:

1. Полуаутоматске прегледне табеле - табеле које су унапред креиране на полуаутоматски начин, уз различите хеуристике;
2. Уклањање суфикса - подразумева постојање ускладиштених правила за проналажење основног облика речи. Оваква правила могу да уклањају и префиксе, мада се уобичајено користи за суфиксе;
3. „Лематизација“ – софистицирани присуп који користи информацију о врсти речи, како би одредио основни облик речи. Правила нормализације зависе од врсте речи и веома су специфична за сваки језик.

Понекад се сматра да се лематизација разликује од процеса „*stemming*“, јер она превазилази једноставна правила уклањања и користи морфолошке корене речи. Овакав приступ даје верзију из речника за посматрану реч, и позната је као „лема“. Лематизатору

је за обављање задатка потребан велики речник и знање специфично за сваки језик, у поређењу са другим алатима за „*stemming*“.

Класичан алгоритам за „*stemming*“ за текстове на енглеском језику је Портеров алгоритам [50]. Најновија верзија Портеровог алгоритма се назива и „*Snowball*“, односно „грудва снега“. За текстове на српском језику постоји неколико објављених имплементација алгоритма за „*stemming*“. У свом истраживању, аутори [51] су користили следеће имплементације:

- Стемер и лематизатор за језике са богатом флексијом и оскудним ресурсима, заснован за обухватању суфикса [14];
- Стемер за српски језик [15];
- Стемер за хрватски језик [17].

Према истраживању од стране аутора [14], тачност стемера за српски језик који су они тестирали је нешто мало мања од 80%. Треба напоменути да је последњи наведени стемер направљен за употребу за текстове на хрватском језику, а због сличности српског и хрватског језика, могуће је уз одређене услове овај стемер користити и за текстове на српском језику. Примена овог стемера на текстове на српском језику је чак добила одличне оцене у истраживању које је објављено у [52].

За претходно набројане стемере, очекује се да улазни текст буде форматиран употребом карактера *UTF-8*, при чему ће и излазни текст бити добијен у овом скупу карактера. Пошто се у српском језику користе два писма, ћирилица и латиница, улазни текстови за српске стемере могу да буду на оба ова писма, али као излазни текст, ови стемери производе текстове у латиничном писму.

Стемери за српски језик интерно користе тзв. „*dual1*“ кодни систем (ен. *dual1 coding system*), у коме су дозвољена само латинична слова, без дијакриктичких знакова (ен. *diacritical marks*). Да би се добили текстови у овом кодном систему, сва ћирилична слова се прво преведу у еквивалент латичничног писма, а након тога се сва слова са дијакриктичким ознакама замењују са одговарајућим словима без ових ознака (нпр. Č/č се мења са Cx/cx, Ć/ć са Cy/cy, Dž/dž са Dx/dx, Đ/đ са Dy/dy, Ž/ž са Zx/zx, Š/š са Sx/sx, Lj/lj са Ly/ly и Nj/nj са Ny/ny).

У овом раду, за потребе извлачења морфолошког корена речи – *Stemming*, употребљена је јавно доступна имплементација колекције Стемера за српски и хрватски језик, чији је аутор Вук Батановић [53]. Испоручена библиотека, која је написана у

програмском језику *Java*, између осталог омогућава процесуирање садржаја текстуалних датотека из командне линије, употребом следеће команде:

```
java -jar SCStemmers.jar StemmerID InputFile OutputFile
```

где је *StemmerID* број одговарајућег алгоритма:

1. „Кешељ & Шипка – Greedy“ [14], у даљем тексту „Стемер1“
2. „Кешељ & Шипка – Optimal“ [14], у даљем тексту „Стемер2“
3. „Милошевић“ [15], у даљем тексту „Стемер3“
4. „Ljubešić & Pandžić“ [17], у даљем тексту „Стемер4“

Разлика између улазног текста и текста након примене четири претходна алгоритма је дат у следећој табели:

Табела 6 Припрема текста - Stemming, примери

Улазни текст:
['електронског', 'факултета', 'опште', 'одредбе', 'предмет', 'уређивања', 'статутом', 'уређује', 'делатност', 'електронског', 'факултета', 'даљем', 'тексту', 'факултет', 'организација', 'управљање', 'начин', 'финансирања', 'друга', 'питања', 'значаја', 'факултета', 'складу', 'законом', 'високом', 'образовању', 'даљем', 'тексту', 'закон', 'статутом', 'универзитета']
Излазни текст након примене алгоритма „Стемер1“:
['elektronsk', 'fakultet', 'opš', 'odredb', 'predm', 'uređivanj', 'statut', 'uređ', 'delatnos', 'elektronsk', 'fakultet', 'da', 'tekst', 'fakul', 'organizacij', 'upravljanj', 'nači', 'finansiranj', 'dru', 'pitanj', 'znač', 'fakultet', 'sklad', 'zakon', 'viso', 'obrazovanj', 'da', 'tekst', 'zako', 'statut', 'univerzitet']
Излазни текст након примене алгоритма „Стемер2“:
['elektronsk', 'fakultet', 'opš', 'odred', 'predmet', 'uređivanj', 'statut', 'uređ', 'delatnos', 'elektronsk', 'fakultet', 'da', 'tekst', 'fakultet', 'organizacij', 'upravlja', 'način', 'finansiranj', 'dru', 'pitanj', 'znač', 'fakultet', 'sklad', 'zakon', 'viso', 'obrazovanj', 'da', 'tekst', 'zakon', 'statut', 'univerzitet']
Излазни текст након примене алгоритма „Стемер3“:
['elektron', 'fakult', 'opš', 'odredb', 'predm', 'uređivanj', 'statut', 'uređ', 'delatnost', 'elektron', 'fakult', 'dalj', 'tekst', 'fakult', 'organizacij', 'upravljanj', 'način', 'finansiranj', 'drug', 'pitanj', 'znač', 'fakult', 'sklad', 'zakon', 'visok', 'obrazovanj', 'dalj', 'tekst', 'zakon', 'statut', 'univerzit']
Излазни текст након примене алгоритма „Стемер4“:
['elektronsk', 'fakultet', 'opšt', 'odredb', 'predmet', 'uređivanj', 'statut', 'uređuj', 'delatnost', 'elektronsk', 'fakultet', 'dalj', 'tekst', 'fakultet', 'organizacij',

```
'upravljanj', 'način', 'finansiranj', 'drug', 'pitanj',
'značaj', 'fakultet', 'sklad', 'zakon', 'visok',
'obrazovanj', 'dalj', 'tekst', 'zakon', 'statut',
'univerzitet']
```

4.3 Представљање текстова у векторском простору

У већини апликација које се баве истраживањем, користи се векторски простор, као мултидимензионално представљање текста [11]. Овај векторски простор садржи једну димензију за сваку реч, а вредност димензије је увек позитивна, када је реч присутна у тексту. У супротном, вредност је нула. Позитивна вредност може да буде нормализована фрекфенција појављивања те речи или индикатор бинарне вредности 1. У неком посматраном документум број речи које тај документ садржи је мали подскуп речника неког језика. Није неуобичајено да збирке докумената, које чине неки речник, да имају знатно више од сто хиљада речи, док просечан број речи у неком документу може да буде неколико стотина.

Треба имати на уму да се процесом претварања документа у векторско представљање, губи информација о редоследу речи у документу. Стога се овај модел назива моделом „вреће речи“ (ен. *bag-of-words model*) [11]. Постоје два најчешће коришћена вишедимензионална представљања текстуалних података, који одговарају „бинарном моделу“ (некад се назива и „*Bernoulli*“ или „*boolen*“ модел) и „*tf-idf*“ моделу.

У неким апликацијама је довољно да се користи бинарно представљање „0-1“, које пружа информацију да ли се нека реч налази у документу или се не налази. Међутим, бинарним представљањем се губи много информација, јер ово представљање не садржи фрекфенције појављивања појединих речи у тексту, а такође се не врши нормализација за потребе одређивања важности речи у тексту [11]. Међутим, главне предности бинарног представљања су компактност и могућност коришћења апликација, попут оних које се баве присуством или одсуством одређених речи у документу.

Ипак, већина приказа текста не функционише са бинарним моделом и уместо тога се користе нормализоване фрекфенције појавивања термина. Овај модел се назива и „*tf-idf*“, где „*tf*“ означава фрекфенцију или број појављивања неког термина у документу, а „*idf*“ означава инверзну фрекфенцију, односно број докумената који садрже неки термин. Ако је $\bar{X} = (x_1 \ldots x_d)$, d-димензионална презентација текста неког документа, након фазе екстракције термина, тада x_i представља број појављивања неког термина у документу.

У следећој табели је приказано израчунавања фрекфенције појављивања термина у примеру текста:

Табела 7 Фрекфенција појављивања термина у тексту, пример

Улазни текст:
['електронског', 'факултета', 'опште', 'одредбе', 'предмет', 'уређивања', 'статутом', 'уређује', 'делатност', 'електронског', 'факултета', 'даљем', 'тексту', 'факултет', 'организација', 'управљање', 'начин', 'финансирања', 'друга', 'питања', 'значаја', 'факултета', 'складу', 'законом', 'високом', 'образовању', 'даљем', 'тексту', 'закон', 'статутом', 'универзитета']
Фрекфенција појављивања термина у тексту:
[('факултета', 3), ('електронског', 2), ('статутом', 2), ('даљем', 2), ('тексту', 2), ('опште', 1), ('одредбе', 1), ('предмет', 1), ('уређивања', 1), ('уређује', 1), ('делатност', 1), ('факултет', 1), ('организација', 1), ('управљање', 1), ('начин', 1), ('финансирања', 1), ('друга', 1), ('питања', 1), ('значаја', 1), ('складу', 1), ('законом', 1), ('високом', 1), ('образовању', 1), ('закон', 1), ('универзитета', 1)]

Пошто фрекфенције речи у дугачким документима могу понекад значајно да варирају, има смисла да се користе функције за пригушивање (ен. *dumping*) ових фрекфенција. Функција квадратног корена или логаритам, могу да се користе како би се смањио ефекат спама [11]. Инверзна фрекфенција документа id_i, неког i-тог термина је функција броја докумената у којима се овај термин појављује: $id_i=log(n/n_i)$ [11]. Фрекфенција појављивања термина се нормализује множењем броја појављивања термина са инверзном фрекфенцијом документа: $x_i=x_i*id_i$.

4.4 Анализа утицаја методологија и алата за припрему, на величину вектора за мултидимензионално представљање теста

Број карактера у посматраној колекцији докумената, пре и после уклањања знакова интерпункције је дат у следећој табели:

Табела 8 Приказ броја карактера у посматраној колекцији докумената

Текст ИД.	1	2	3	...	**Просечан број карактера у посматраним текстовима**
Сирови текст, без било какве припреме, број карактера у тексту:	37904	220283	160648	…	**110692.7**

Текст са уклоњеним знацима за интерпункцију, број карактера у тексту:	37010	201385	155420	…	**106051.5**

На основу информација, које су приказане у претходној табели, види се да је након уклањања знакова за интерпункцију, смањен просечан број карактера у тексу за око 4.2%.

Број токена у посматраној колекцији докумената, након уклањања знакова за интерпункцију, а према наредним фазама за припрему текста, је дат у следећој табели:

Табела 9 Приказ броја токена у посматраној колекцији докумената, према фазама за припрему текста

Текст ИД.	1	2	3	...	**Просечан број токена у посматраним текстовима**
Број токена у тексту:	5548	30039	22595	…	**15104.1**
Број јединствених токена у тексту:	881	2040	1867	…	**1620.0**
Број јединствених токена након претварања у мала слова:	761	1913	1653	...	**1439.1**
Број јединствених токена након уклањања речи без веће садржајне вредности:	734	1786	1622	...	**1414.1**

На основу података који су приказани, јасно се може видети да свака фаза на свој начин доприноси смањењу вектора за мултидимензионално представљање текстова, па ће захваљујући томе и даља обрада бити ефикаснија.

Следећа фаза у припреми текста је било извлачење морфолошког корена речи (ен. *Stemming*). У овој фази су за исту намену примењена четири различита алгоритма, који су, сваки на свој начин, допринели даљем смањењу вектора за мултидимензионално представљање текстова. У следећој табели су дати подаци о броју јединствених токена након примене сваког од ових алгоритама:

Табела 10 Број токена у посматраној колекцији докумената, након примене различитих алгоритама за извлачење морфолошког корена речи

Текст ИД.	1	2	3	...	**Просечан број токена у посматраним текстовима**
Број јединствених токена након примене аглоритма „Стемер1“:	428	945	858	…	**775.9**
Број јединствених токена након примене аглоритма „Стемер2“:	407	907	814	…	**736.3**

Број јединствених токена након примене аглоритма „Стемер3“:	458	970	892	...	**801.9**
Број јединствених токена након примене аглоритма „Стемер4“:	734	1786	1622	...	**1388.9**

4.5 Израчунавање сличности између текстова након примене различитих алгоритама за припрему текста

Многе апликације за истраживање мултидимензионалних података користе Еуклидско растојање (ен. *Euclidean distance*) како би измерили удаљеност између парова тачака. На први поглед изгледа да ово може да се примени за израчунавање удаљености између текстова, ако се текст посматра као посебан случај мултидимензионалних података. Међутим, Еуклидска удаљеност није добра у рачунању удаљености у мултидимензионалним приказима који су веома проређени, и у којима број нултих вредности значајно варира у различитим тачкама. А, управо ово се често јавља у случају текстова, због различите дужине различитих докумената. Еуклидско растојање ће доследно пријављивати веће вредности за удаљеност између парова докумената, чак и ако су велике фракције тих докумената заједничке [11]. Ово указује да је потребно да се користе функције које јако нормализују вредности за удаљеност (или за сличност) између текстова, за различите дужине текстова. Природно решење за овај проблем је коришћење косинуса угла између мултидимензионалних вектора, који представљају два документа [11]. То је зато што косинус између пара вектора не зависи од дужине вектора, већ само од угла између њих. Осим тога, косинусна сличност (ен. *cosine similarity*) је погодна за случајеве у којима је фрекфенција термина (ен. *term frequency*) дата експлицитно [11]. Применом косинусне сличности се добијају вредности у интервалу од 0 до 1. Пошто израчунавање косинуса угла између пара вектора зависи од фракција са дељеним речима у сваком од посматраних докумената, због тога ова функција у великој мери не зависи од дужине докумената.

Након урађене припреме текстова, како је то претходно објашњено, тако припремљени документи су претворени у векторски облик по „*tf-idf*“ моделу. У овом раду, као меру сличности између посматраних текстова, израчунат је косинус угла између пара вектора. На тај начин је добијена матрица сличности *S*, која садржи добијене податке о сличностима између парова посматраних докумената:

$$S = \left| s_{ij} \right| = \begin{vmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & \dots & \dots & \dots \\ \dots & \dots & \dots & s_{(n-1)n} \\ s_{n1} & \dots & s_{n((n-1)} & s_{nn} \end{vmatrix}$$

где је

S – матрица сличности између докумената

s_{ij} – сличност између текстова i и j, посматраног скупа докумената

Јасно је да је ово симетрична квадратна матрица, за коју важи да је

- s_{ii}=1, за свако i
- s_{ij}=s_{ji}, за свако i и за свако j

Како је у овом раду коришћено четири различита алгоритма за „*Stemming*“ за српски језик, добијене су четири матрице са вредностима за сличност између посматраних докумената:

- S_1 – матрица сличности између посматраних докумената након примене алгоритма „Стемер1“
- S_2 – матрица сличности између посматраних докумената након примене алгоритма „Стемер2“
- S_3 – матрица сличности између посматраних докумената након примене алгоритма „Стемер3“
- S_4 – матрица сличности између посматраних докумената након примене алгоритма „Стемер4“

Приказ добијених вредности ових матрица је:

$$S_1 = \left| s_{ij} \right|_1 = \begin{vmatrix} 1 & 0.606 & ... & 0.672 \\ 0.606 & 1 & ... & ... \\ ... & ... & 1 & 0.900 \\ 0.672 & ... & 0.900 & 1 \end{vmatrix}, \quad S_2 = \left| s_{ij} \right|_2 = \begin{vmatrix} 1 & 0.621 & ... & 0.658 \\ 0.621 & 1 & ... & ... \\ ... & ... & 1 & 0.907 \\ 0.658 & ... & 0.907 & 1 \end{vmatrix},$$

$$S_3 = \left| s_{ij} \right|_3 = \begin{vmatrix} 1 & 0.626 & ... & 0.652 \\ 0.626 & 1 & ... & ... \\ ... & ... & 1 & 0.909 \\ 0.652 & ... & 0.909 & 1 \end{vmatrix}, \quad S_4 = \left| s_{ij} \right|_4 = \begin{vmatrix} 1 & 0.565 & ... & 0.631 \\ 0.565 & 1 & ... & ... \\ ... & ... & 1 & 0.869 \\ 0.631 & ... & 0.869 & 1 \end{vmatrix}.$$

Као коначан резултат за сличност између два текста у посматраном скупу докумената, коришћена је средња вредност за сличност, која је добијена употребом набројаних алгоритама за „*Stemming*“:

$$\overline{s_{ij}} = \frac{\sum(s_{ij})}{n} = \frac{\sum(s_{ij})}{4}$$

где је:

$\overline{s_{ij}}$ – просечна вредност за сличност између два текста, узимајући у обзир све коришћене алгоритме „*Stemming*“;

n=4 – број различитих алгоритама који су коришћени у овом раду.

Приказ просечних вредности за сличност између посматраних текстова је дат у следећој матрици:

$$\overline{S} = \left|\overline{s_{ij}}\right| = \begin{vmatrix} 1 & 0.605 & \dots & 0.653 \\ 0.605 & 1 & \dots & \dots \\ \dots & \dots & 1 & 0.896 \\ 0.653 & \dots & 0.896 & 1 \end{vmatrix}$$

У следећој табели је дат приказ вредности за сличност између појединих текстова, након примене различитих алгоритама за извлачење морфолошког корена речи, као и израчунате просечне вредности за сличност између текстова:

Табела 11 Сличност између Текста(i) и Текста(j) након примене различитих алгоритама за извлачење морфолошког корена речи

Ознака текста		**Сличност између Текста(i) и Текста(j) након примене алгоритама**				**Просечна сличност**
i	***j***	**Стемер1**	**Стемер2**	**Стемер3**	**Стемер4**	
1	1	100.0%	100.0%	100.0%	100.0%	100.0%
1	2	60.6%	62.1%	62.6%	56.5%	60.5%
1	3	77.7%	77.7%	78.5%	74.3%	77.1%
1	4	78.3%	78.6%	79.1%	75.5%	77.9%
...	...	...	...	...	...	...

4.6 Анализа утицаја методологија и алата за припрему текстова на израчунавање сличности између њих

Овај део обухвата следеће анализе:

- Анализа утицаја целокупног поступка припреме текстова на израчунавање сличности између њих;
- Анализа утицаја различитих алгоритама и алата за извлачење морфолошког корена речи из текстова на српском језику, на израчунавање сличности између текстова.

4.6.1 Утицај припреме текстова на израчунавање сличности између њих

У следећој табели је дат приказ поређења израчунатих вредности за сличност између посматраних текстова, које су добијене након спроведеног целокупног поступка за припрему текстова, са вредностима које би се добиле да се сличност између текстова израчунавала без поступка за припрему текста:

Табела 12 Сличност између посматраних текстова након припреме и без припреме текстова

Ознака текста		**Сличност између Текста(i) и Текста(j)**	
i	***j***	**Након припреме текстова**	**Без припреме текстова**
1	1	100.0%	100.0%
1	2	60.5%	65.6%
1	3	77.1%	91.9%
1	4	77.9%	91.8%
...	...	...	...
Просечна Вредност:		71.3%	80.5%

Просечна сличност између свих посматраних текстова, која је израчуната узимајући у обзир све примењене алгоритме за „*Stemming*“ је $\overline{s_{ij}}$=71.3%. Просечна вредност за сличност између посматраних текстова у сировом облику, на које није примењена ни једна од техника за припрему или препроцесирање текста износи $\overline{s_{ij}}$=80.5%. На основу ових вредности, може да се закључи да поступци за припрему текстова утичу на смањење вредности за сличност, у односу на вредности које би се добиле да се израчунава сличност између текстова без њихове претходне припреме (без препроцесирања текста). Другим речима, разлике између текстова су уочљивије када се израчунавање сличности врши након препроцесирања текста.

4.6.2 Утицај алгоритама за извлачење морфолошког корена речи на израчунавање сличности између текстова

У следећој табели су дате просечне вредности за сличност између посматраних текстова након примене алгоритама за извлачење морфолошког корена речи, рачунато за различите текстове (индекси *i* и *j* су различити):

Табела 13 Просечне вредности за сличност између посматраних текстова након примене различитих алгоритама за извлачење морфолошког корена речи

Ознака алгоритма за извлачење морфолошког корена речи	**Стемер1**	**Стемер2**	**Стемер3**	**Стемер4**
Просечне вредности за сличност између текстова	73.2%	71.9%	73.2%	66.9%

На основу добијених података, може се закључити да су различити алгоритми за припрему текстова допринели томе да се добијени резултати за сличност између текстова разликују. Може се приметити да су са становишта просечних вредности, добијени слични резултати након имплементације алгоритма „Стемер1“ и алгоритма „Стемер3“.

Ако се посматра сличност између појединих парова текстова, са становишта резултата добијених применом различитих алгоритама за „*Stemming*“, потражени су документи за које се израчунате вредности за сличност према појединим алгоритмима највише разликују од просечних вредности. У следећој табели су за поједине примере дате израчунате вредности за сличност између појединих текстова, као и израчунате просечне вредности за сличност:

Табела 14 Поједини парови текстова за које се израчуната сличност највише разликује од просечних вредности

Ознака текста		**Сличност између Текста(i) и Текста(j) након примете алгоритма**				**Просечна сличност**
i	***j***	**Стемер1**	**Стемер2**	**Стемер3**	**Стемер4**	
...	...	...	...	...	...	...
16	116	55.0%	58.8%	**66.2% (разлика 20.4%)**	3.1%	45.8%
...	...	...	...	...	...	...
75	116	75.7%	**77.7% (разлика 18%)**	79.8%	5.7% **(разлика 54%)**	59.7%
...	...	...	...	...	...	...
193	196	**68.9% (разлика 16.6%)**	69.6%	68.1%	2.7	52.3%
...	...	...	...	...	...	...

Подаци из претходне табеле пружају следеће информације:

- Након примене алгоритма „Стемер1“, највеће одступање овог алгоритма у односу на просечне вредности је 16.6%. За документе са ознакама 193 и 196, у посматраном скупу докумената, употребом овог алгоритма је добијена сличност између ових докумената 68.9%, док је просечна вредност за сличност између овог пара докумената 52.3%, узимајући у обзир све алгоритме.
- Након примене алгоритма „Стемер2“, највеће одступање овог алгоритма у односу на просечне вредности је 18%. За документе са ознакама 75 и 116, у посматраном скупу докумената, употребом овог алгоритма је добијена сличност између ових

докумената од 77.7%, док је просечна вредност за сличност између овог пара докумената 59.7%, узимајући у обзир све алгоритме.

- Након примене алгоритма „Стемер3“, највеће одступање овог алгоритма, у односу на просечне вредности је 20.4%. За документе са ознакама 16 и 116, у посматраном скупу докумената, употребом овог алгоритма је добијена сличност између ових докумената од 66.2%, док је просечна вредност за сличност између овог пара докумената 45.8%, узимајући у обзир све алгоритме.
- Након примене алгоритма „Стемер4“, највеће одступање овог алгоритма у односу на просечне вредности је чак 54%. За документе са ознакама 75 и 116, у посматраном скупу докумената, употребом овог алгоритма је добијена сличност између ових докумената од 5.7%, док је просечна вредност за сличност између овог пара докумената 59.7%, узимајући у обзир све алгоритме.

Текстуални подаци, који су коришћени у овом раду, су јавно доступни, спадају у групу формалних, правних текстова, и написани су од стране различитих аутора. Као такви, спадају у „дуге“ текстове и по томе, предмет овог рада нису били кратки текстови попут коментара или твитова на Интернету. На основу наведеног, подаци који су коришћени за истраживање ни на који начин нису могли да утичу на резултате истраживања, односно, ово истраживање може да се понови на неком другом скупу података.

Технике, попут алгоритама за извлачење морфолошког корена речи, помажу у компензацији ретко поседнутих података (ен. *data sparseness*), а прекомерно агресиван стеминг може лако да погорша перформансе класификације [54].

5 Истраживање језичких израза у правним документима

У овом поглављу ће бити описан поступак истраживања и проналажења језичких израза са највећом фрекфенцијом коришћења у текстовима закона који се примењују на територији Републике Србије, а затим ће посебна пажња бити посвећена језичким изразима који се користе за референцирање или повезивање. Експеримент је спроведен на колекцији која се састоји од 1120 текстова закона и садржи следеће фазе:

- Прикупљање података (ен. *Data collection*)
- Припрема података – препроцесирање (ен. *Data preprocessing*)
- Трансформација података (ен. *Data transformation*)
- Истраживање језичких израза (ен. *Linguistic forms mining*)
- Анализа веза (ен. *Link analysis*) између правних докумената на основу пронађених језичких израза

5.1 Прикупљање података

У правне документе спадају закони, уредбе, правилници, одлуке, решења, статути, наредбе, упутства и остали документи из ове области. За потребе овог рада и експеримента који се спроведен, ограничили смо се само на законе који се примењују на територији Републике Србије. Што се тиче Републике Србије, посебним Законом о објављивању закона и других прописа и аката је у Члану 29 дефинисано да је да су свим корисницима Интернета доступни без накнаде текстови важећих прописа и других аката Републике Србије. У складу са тим, текстови ових закона су јавно доступни и налазе се на неколико јавних веб сајтова, као што су веб сајт Народне скупштине Републике Србије (http://www.parlament.gov.rs/akti/doneti-zakoni/doneti-zakoni.1033.html), веб сајт "Правно-информациони систем Републике Србије" (http://www.pravno-informacioni-sistem.rs/SlGlasnikPortal/reg/content) и други. Са ових адреса је могуће преузети интегралне текстове закона и за потребе овог рада је прикупљена збирка текстова важећих закона.

На веб адресама које омогућавају приступ овим текстовима, закони су подељени у различите категорије и подкатегорије, али то није био предмет интересовања за потребе овог рада.

5.2 Припрема података – препроцесирање

Као што је претходно описано, закони који су прикупљени за потребе овог рада су у облику текстуалних докумената. Језик који се користи у законима и прописима је доста формалан и своди се на употребу сличних језичких израза, који се понављају, тако да се намеће потреба да се ради истраживање са језичким изразима, а не са појединим речима. Према [45], фразе, изрази који се састоје од више речи или изрази који се састоје од више речи спојених цртицама, не представљају својства на нивоу појединачних речи. Дакле, у овом раду, предмет посматрања нису поједине речи, већ језички изрази који се састоје од више речи (ен. *multiword expressions*), односно, често понављани делови реченица истог облика.

Из тог разлога, у овој фази припреме тектуалних података, нису примењени сви алгоритми који се уобичајено користе и који су претходно описани. У овој фази, за потребе даље анализе, најпре су документи претворени у „обичан текст“ (ен. *plain text*). На тај начин су занемарена било каква претходна форматирања или уређивања текста. Добијен је скуп докумената у формату **.TXT*, при чему је текст једног закона сачуван у једном документу. Затим је коришћен алгоритам за нормализацију величине слова у тексту (ен. *case normalization*), који претвара цео текст у мала слова, како би се избегао проблем различитог тумачења истог текста, уколико је написан комбинацијом великих и малих слова.

5.3 Трансформација података

Према [2], трансформација података може да се уради на неки од следећих пет начина: Обједињавање вредности података (ен. *Aggregation of data values*), Нормализација вредности података (ен. *Normalisation of data values*), Смањење својстава (ен. *Feature reduction*), Смањење примера и реструктурирање (ен. *Example reduction and Restructuring*). За потребе овог рада је изабрана сегментација текста у природно повезане секције, зато што може да буде корисна са становишта проналажења информација и каснијег евентуалног повезивања информација. Осим тога, избор одговарајућих делова текста је користан када су документи дуги и када су само делови текста интересантни за кориснике [55].

Уобичајена унутрашња подела закона и других прописа се врши на следеће шире класификационе јединице [3]:

- део,

- глава,
- одељак и
- пододељак.

Део обухвата тематску целину прописа и најшира је класификациона јединица прописа. Део може да се дели на главе, којима тематске целине могу да се деле на функционалне или смисаоне целине. Глава може да буде подељена на одељке, а одељци се даље могу поделити на пододељке.

Члан закона је једна логична целина која садржи једну или више правних норми. Члан даље може да се дели на ставове, ставови на тачке, тачке на подтачке, а подтачке на алинеје. Осим тога, Члан је основна класификациона јединица у законима. Трансформација података је урађена као сегментација интегралних текстова закона, која је имала за циљ да један члан закона, као основна класификациона и логична целина, буде сачуван као један запис у бази података. Поступак сегментације текста неког закона је урађен у програмском језику *Python*, на следећи начин:

```
file=r'tekst_nekog_zakona.txt'
openFile= open(file, 'r', encoding='utf-8-sig')
clanovi = []
clanovi= openFile.read().split('\nČlan')
for i in range(1,len(clanovi)):
    clanovi[i]="Član"+clanovi[i]
```

Из претходног програмског кода се може видети:

- да је као сепаратор за сегментацију текстова, коришћен стринг „`Član`“, који се налази непосредно после ознаке за нови ред „`\n`“
- употребом команде `split` се овај сепаратор губи из добијених текстова, па је накнадно исти тај стринг враћен у текстове, за све сегменте осим првог, помоћу одговарајуће петље.

У примеру колекције која је предмет овог рада, применом описаног поступка сегментације текстова закона на поједине чланове закона, добијено је 59167 записа у бази података. За сваки од записа се такође складиште подаци о називу Закона коме тај Члан припада, као и одговарајући идентификатор тог Члана. Предмет овог рада није била унутрашња подела прописа на шире класификационе јединице, као што су део, глава, одељак и пододељак. Ово свакако може да буде предмет неког другог истраживања.

Према подели која је дата у [45], на претходно описан начин су добијени ентитети или основни градивни блокови текста од којих се састоје сви закони. У даљем тексту биће показано да ли је у поступку истраживања језичких израза у правним документима могуће издвојити још неки од основних елемената према претходно наведеној подели.

5.4 Истраживање језичких израза

Према [56], „Језички изрази“ (ен. *linguistic forms*) су

1. Јединице или обрасци (шаблони, мустре) језика, које се обично разматрају независно од његове придружене функције или вредности.
2. Карактеристике или облик таквих јединица или образаца језика. Такође, ове карактеристике се посматрају у односу на једну такву јединицу или образац језика.

У литератури постоје и други изрази са истим или сличним значењем, као нпр. "Кључне фразе", "секвенце речи", "реченице" и слично. Како реч "фразе" никако није адекватан за опис правних појмова или израза који су предмет овог рада, у складу са дефиницијом из речника [56], у овом раду се користи израз „језички изрази“ (ен. *Linguistic forms*), иако је и израз "секвенце речи" (ен. *sequences of words*), такође одговарајући. У контексту овог рада: **Језички израз** је честа реч (или основа речи), секвенца састављена од више речи (кључна фраза, ен. *key-phrase*) где су речи (или основе речи) у одређеном редоследу.

Технике које се уобичајено користе у поступку претраживања информација и истраживању текстова у правним документима и базама података су [2]:

- извлачење информација (ен. *information extraction*)
- сажимање или резимирање текста (ен. *text summarisation*)
- категоризација текста (ен. *text categorisation*)
- кластеризација текста (ен. *text clasterisation*)

Према овој подели, методологија која је коришћена у истраживању у овом документу, може да се сврста у извлачење (ен. *extraction*) информација. Централна карактеристика извлачења информација је попуњавање шаблона специфичног за неки домен [12], а у случају овог рада, то је правни домен.

Осим чланова закона, који се могу сматрати ентитетима како је то претходно описано, у даљем поступку екстракције се из текста чланова закона проналазе и издвајају

језички изрази, који се такође могу сматрати новим ентитетима. Према [34], постоје две стратегије за екстракцију ентитета:

- приступ заснован на дефинисању правила (ен. *rule-based*), при чему се дефинишу условна правила која се примењују у тексту да би се идентификовали могући ентитети;
- статистички приступ, који третира екстракцију ентитета као процес класификације секвенци.

У овом раду користи се статистички приступ екстракцији језичких израза са циљем проналажења језичких израза са највећом вероватноћом коришћења у посматраним текстовима. За ту намену, у програмском језику *Python* је направљен програм за проналажење свих језичких израза (делова реченица) у претходно описаној бази података. Један од параметара за извршавање овог програма је N - улазна променљива која се односи на број речи из којих се састоји језички израз. Укратко, примењен је следећи алгоритам:

```
for n in range(2, N):
    for record in databasecursor:
        find all n-word linguistic_forms in current record
        insert into table all n-word linguistic_forms
select and group n-word linguistic_forms, count of n-word
linguistic_forms
```

Према [20], у пракси се може поставити максимална улазна вредност за пронемљиву N да би се ограничио број речи појединих језичких израза (да би се ограничила дужина фразе). Чак и ако није дато експлицитно ограничење, дужина фразе је типично мала константа.

У пракси, најпре је покренуто извршавање програма за проналажење свих језичких израза који се састоје од 2 речи, затим који се састоје од 3 речи, итд. Као резултат извршавања програма је добијена табела у бази података са свим језичким изразима који се помињу у свим члановима свих посматраних закона, као и то колико пута се у неком члану неког закона помиње неки језички израз. У следећој табели је приказано колико различитих језичких израза је пронађено у бази података која је предмет овог рада:

Tabela 1 Укупан број пронађених језичких израза у посматраној бази података

Језички изрази који се састоје од...	Број пронађених различитих језичких израза
2 речи	1651855
3 речи	3702636
4 речи	5021611
5 речи	5754751
…	…

За потребе даље евалуације, пронађени језички изрази су груписани и извршено је пребројавање њиховог појављивања у различитим члановима закона. На основу ових информација је сада могуће добити тачне податке о томе који језички изрази су у којој мери коришћени за израду текстова посматраних закона. У следећој табели су дати неки од најчешће коришћених језичких израза у посматраној бази података и одговарајући проценат њиховог појављивања у члановима закона:

Табела 15 Неки од најчешће коришћених језичких израза у посматраној бази података и одговарајући проценат њиховог појављивања у члановима закона

Језички изрази који се састоје од...			
2 речи	**3 речи**	**4 речи**	**5 речи**
“овог члана 26.67%	“у складу са” 23.66%	“става 1 овог члана” 17.10%	„из става 1 овог члана“ 15.63%
“у складу” 24.58%	“1 овог члана” 18.45%	“из става 1 овог” 15.65%	“из става 2 овог члана” 4.19%
“складу са” 23.66%	“става 1 овог” 17.13%	“у складу са законом” 6.10%	“у складу са овим законом” 3.79%
“овог закона” 22.45%	“из става 1” 16.95%	“у складу са овим” 4.68%	“ступања на снагу овог закона” 3.41%
“из става” 21.67%	“у року од” 9.39%	“става 2 овог члана” 4.67%	“дана ступања на снагу овог” 2.51%
…	…	…	…

5.5 Анализа веза између правних докумената на основу пронађених језичких израза

У даљој анализи ће ће бити приказан значај и могуће примене добијених података. Као што је написано у књизи [24], правни документи су препуни имплицитних и експлицитних референци, а ово истраживање је то и потврдило.

Одмах се може приметити да се најчешће пронађени изрази користе за навођење или повезивање са истим или другим ставовима, члановима или законима. На основу

језичких израза са највећом фрекфенцијом појављивања, мануелно је вршено рударење података, да би били пронађени слични језички изрази, али са мањом фрекфенцијом појављивања у посматраном скупу језичких израза. За те потребе су коришћени *SQL* упити. Пример *SQL* упита, који је коришћен за проналажење језичких израза који се састоје од 3 речи, и који се користе за повезивање са другим законима, а који су слични са изразом као што је „у складу“ (најчешће коришћени језички израз у српским законима, дужине 2 речи, а који се користи за повезивање са другим законима) је:

```
SELECT * FROM tblExtractedLF
WHERE ((tblExtractedLF.LinguisticForms)
Like "*у складу*");
```

На овај начин је пронађен већи број језичких израза дужине 3 речи, који се користе за повезивање са другим законима, као што су: „у складу са“ (који је и најчешће коришћен кезички израз дужине 3 речи“, „складу са законом“, „складу са чланом“, „складу са одредбама“, „складу са прописима“, „складу са националним“, итд.

Исти поступак је коришћен за проналажење језичких израза дужине 4 речи, који се користе за повезивање са другим прописима, а који су слични са изразом као што је „у складу са“. Пронађени су језички изрази као што су „у складу са законом“ (који је најчешћи језички израз дужине 4 речи), „у складу са чланом“, „у складу са националним“, „у складу са посебним“, итд. Затим је овај поступак поновљен за проналажење језичких израза дужине 5 речи, који се користе за повезивање са другим законима.

Након тога је примењен поступак проналажења и издвајања језичких израза који се састоје од већег броја речи (више од 5 речи), који су слични са претходно пронађеним језичким изразима. То је, такође, урађено мануелно, употребом *SQL* упита над посматраним скупом прописа, а пример таквог упита је:

```
SELECT * FROM tblLaws
WHERE ((tblLaws.ArticleText)
Like "*у складу са*");
```

На тај начин су пронађени језички изрази попут „у складу са Уставом“, који се помиње у 82 члана закона, „у складу са Законом о пореском поступку и пореској администрацији“, који се помиње у 71 члану закона, и многе друге језичке изразе који се користе за повезивање. Укупан број језичких израза, који су на овај начин пронађени и издвојени је 2069. Број веза између прописа, а које су пронађене на основу овако

пронађених језичких израза је 38074. У наставку ће бити приказани само примери таквих језичких израза зато што је детаљна листа свих откривених израза превелика.

5.6 Језички изрази за анализу веза у правним документима

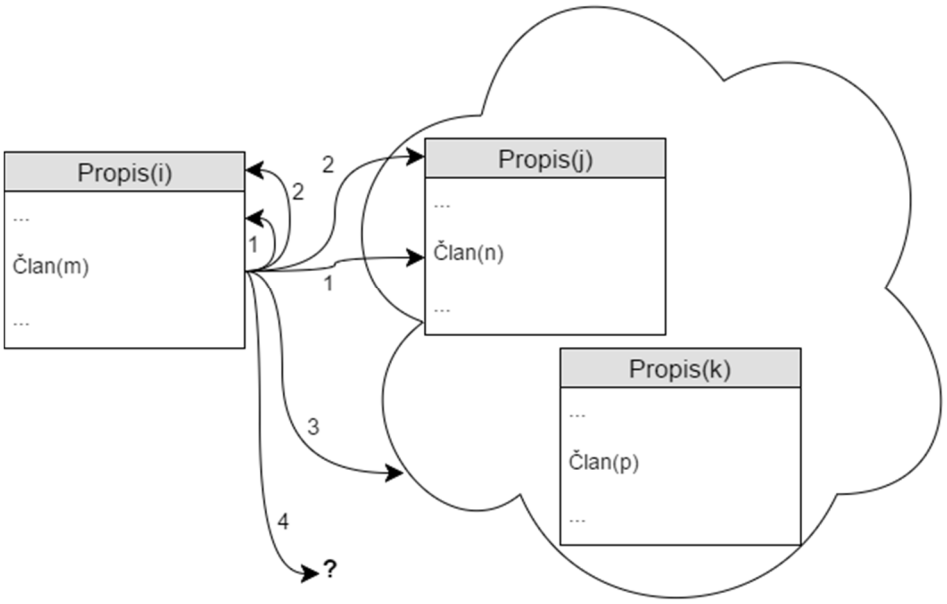
У свом истраживању *Waltl* и сарадници су представили следеће различите типове референци у правним документима [30]: Потпуно експлицитне референце (ен. *Full-explicit reference, FR*), Полуекспликитне референце (ен. *Semi-explicit Reference, SR*), Имплицитне референце (ен. *Implicit Reference, IR*) и Прећутне референце (ен. *Tacit Reference, TR*). Детаљнијом анализом језичких израза у анализираним документима, откривено је четири типа језичких израза који се користе за повезивање, као што је то приказано на Слици 1:

1. Језички изрази који се користе за упућивање на одређене одредбе истог или другог прописа, а који се користе да би се избегло понављање појединих одредаба. У одредби којом се упућује на други пропис, наведе се називи тог прописа и број службеног гласила у коме је пропис објављен, а ако се упућује на одређене одредбе тог прописа, наводе се и чланови у којима су те одредбе садржане [3]. Ови језички изрази су најпрецизнији за анализу линкова и они спадају у категорију Потпуно експлицитних референци (FR).
2. Језички изрази који се користе за позивање на неки одређени закон, а да се при томе не прецизира неки одређени члан закона. Ови језички изрази су такође корисни за примену анализе линкова, а ова група спада у Полуекспликитне референце (SR).
3. Језички изрази који се користе за упућивање на други прописе, навођењем уопштеног назива за одређену врсту прописа из области која се уређује, а да се при том не помиње тачан назив неког закона или прописа. Ова група језичких израза спада у категорију Имплицитних референци (IR). Овакви језички изрази се углавном употребљавају у ситуацијама када је потребно поштовати временски редослед доношења закона из одређене области и када је потребно поштовани хијерархију аката и техника примењивања. Неки од језичких израза, који спадају у ову групу су:
    - „у складу са законом којим се уређује %“ – Ови језички изрази могу да пруже информацију која може да се искористи за анализу линкова, али

нису довољно прецизни. Такви језички изрази могу да се искористе за „ручно“ референцирање на поједине законе.

- „којим се уређује област %“ – Ови језички изрази такође нису довољно прецизни. Такви језички изрази могу да се искористе за референцирање ма групу закона који припадају некој области.

4. „Пређутни“ језички изрази (TR) се користе уопштено за упућивање на примену закона (у множини), а да се при том не помиње ни један одређени закон. Ови језички изрази не пружају информацију која може да се искористи за анализу линкова. Такви језички изрази су:
    - „у складу са законом“, „у складу са законом и другим прописима“, без навођења неког одређеног закона, а при чему се уопштено мисли на усклађеност „са законима“;
    - „у складу са посебним прописима“;
    - „у складу са досадашњим прописима“;
    - „у складу са прописима који су важили“, „у складу са раније важећим прописима“;
    - „у складу са важећим прописима“, „у складу са важећом законском регулативом“;
    - „у складу са одговарајућим прописима“;
    - „у складу са општим прописима“



Слика 1 Типови језичких израза за повезивање

Најчешће коришћени језички изрази су језички изрази који се употребљавају за референцирање или за повезивање са истим или са неким другим ставовима, члановима или законима. Уколико су пронађени и издвојени, овакви језички изрази управо пружају могућност повезивања и детаљне анализе веза између закона и делова закона. Језички изрази који се користе за повезивање се даље могу поделити у следеће подгрупе:

- **Језички изрази који се користе за ауторефернцирање или аутоповезивање** – У српском језику, у ову групу спадају следећи језички изрази:
    - „овог члана“ – најчешће коришћен језички израз, који се помиње у 26.6% чланова различитих закона. Овај језички израз се користи за ауторефернцирање, односно за позивање у тексту на исти члан закона;
    - „овог закона“ – често коришћен језички израз, који се помиње у 22.45% чланова различитих закона. Овај језички израз се користи за ауторефернцирање, односно за позивање у тексту на исти закон;
    - „из става % овог члана“ – често коришћен језички израз, који се користи у 21.2% чланова различитих закона, за референцирање на неки став истог члана закона, при чему се ознака „%“ односи на ознаку или редни број одређеног става;
    - „у складу са овим“ – често коришћен језички израз, који се помиње у 4.68% чланова различитих закона. Овај језички израз се користи за опште

ауторефернцирање, односно за позивање у тексту на исти закон, члан, став и сл. На пример, језички израз „у складу са овим законом“ се помиње у 3.79% чланова различитих закона и користи се за референцирање на исти закон;

- „у складу са чл. % овог закона“ (1.29%), при чему у тексту може да се помиње један или више чланова закона;
- „на основу одредаба чл. % овог закона“ (мање од 1%);
- „у складу са одредбама овог члана“ (мање од 1%), „у складу са одредбама овог закона“ (мање од 1%).

- **Језички изрази који се користе за референцирање или повезивање са другим националним законима, члановима закона или ставовима** – у српском језику, у ову групу спадају следећи језички изрази:
  - „у складу са“ – често коришћен језички израз, који се помиње у 23.66% чланова различитих закона. Овај језички израз се користи за референцирање или позивање на неки став, члан или закон;
  - „из става“ – често коришћен језички израз који се помиње у 21.67% чланова различитих закона. Овај језички израз се користи за референцирање или позивање на неки одређени став, неког члана, неког закона;
  - „у складу са законом“ – често коришћен језички израз, који се помиње у 6.10% чланова различитих закона. Овај језички израз се користи за референцирање или позивање на неки одређени закон;
  - „у складу са Уставом“ (мање од 1%).
- **Језички изрази који се корсите за референцирање или позивање са међународним законима или међународним институцијама** (Напомена: предмет овог рада није био одређивање ка којим међународним законима постоје везе или референце, већ је само идентификовано постојање тих веза):
  - „Закон о потврђивању %“, „Закон о ратификацији %“; (мање од 1%)
  - „усаглашавање % са одговарајућим прописима Европске уније“, „између Републике Србије и Европске уније“, „закона ЕУ“, „законе ЕУ“, „у службеном листу Европске уније“, „надлежним органима Европске уније“, „Процес стабилизације и придруживања Европској унији“, „У складу са % прописима Европске уније“ (мање од 1%);

- „у складу са међународним обавезама“, „у складу са правилима потврђеним међународним споразумима“, „у складу са закљученим међународним уговором“, „у складу са општеприхваћеним правилима међународног права“ (мање од 1%),
- „у складу са Резолуцијом Савета безбедности Уједињених нација“, „у складу са Резолуцијом Савета безбедности УН“, „у складу са Резолуцијом 1244 Савета безбедности Уједињених нација“,... (мање од 1%).

- **Језички изрази који се користе за референцирање или повезивање са нижим правним актима** (Предмет овог рада није био анализа ових веза или референци):
  - „у складу са уговором“, „у складу са тим уговором“ – језички изрази који се помињу у мање од 1% чланова различитих закона;
  - „у складу са статутом“ (мање од 1%)
  - „у складу са посебним актима“, „у складу са актом“ (мање од 1%)
  - „у складу са упутствима“ (мање од 1%)

Приликом истраживања језичких израза и проналажења оних који могу да се користе као линкови, за потребе овог рада, структура линкова је направљена тако да се неки линк састоји из два дела:

- Идентификатор закона и
- Идентификатор члана закона.

Према овој структури, линк који повезује Члан(*m*) Прописа(*i*) са Чланом(*n*) Прописа(*j*), може да се представи на следећи начин:

```
Link(Source)=Target
Link(Пропис(i),Члан(m))=[Пропис(j),Члан(n)]
```

У случајевима када се језички изрази користе за позивање на неки одређени закон, а да се при том не прецизира одређени члан закона, у том случају се креирају линкови који се састоје од идентификатора закона, а уместо индентификатора члана закона се користи нула:

```
Link(Пропис(i),Члан(m))=[Пропис(j),0]
```

На овај начин је добијен модел који се састоји од скупа ентитета и скупа релација између њих, а већина операција које се спроводе над таквим скуповима се моделира као операције на графоцима [45].

У тексту неког Прописа(i) може више пута да се помиње неки Пропис(*j*), али у овом раду није узимано у обзир колико се пута Пропис(*j*) помиње у тексту Прописа(*i*).

За потребе овог рада, информација о постојању везе између два прописа је сасвим довољна, а информација о томе колико пута се у тексту помиње неки пропис је занемарена. На основу овога, у овом раду везама или линковима између прописа нису додељени тежински фактори, па самим тим и граф који се добија није тежински [57].

5.7 Евалуација извлачења линкова из правних докумената, на односу откривених језичких израза

У овом раду су најпре извлачени сви језички изрази из посматраног скупа прописа, а затим је пажња усмерена на језичке изразе који могу да се употребе за повезивање или референцирање са истим или са другим законима. Јасно је да осим ових језичких израза, у скупу свих језичких израза који су извучени, постоје и други језички изрази, а који се не користе за референцирање или повезивање и који могу да буду предмет неког будућег истраживања. У овој евалуацији ће бити описана процена резултата добијања језичких израза који могу да се користе за повезивање.

Према [34], ефикасност екстрактора ентитета из неког корпуса се одређује бодовањем излазних резултата система према познатим ознакама за исти тип корпуса. У ту сврху смо издвојили тестни скуп прописа, у коме је познато да постоји 3111 веза ка истом или ка другим прописима, и на том скупу података смо тестирали ову ефикасност. Квалитет система за екстракцију ентитета се одређује помоћу „Прецизности“ (ен. *Precision*) и „Одзива“ (ен. *Recall*). **Прецизност екстрактора** је однос броја издвојених именованих ентитета који су тачни и укупног броја издвојених ентитета:

```
Прецизност(p)=#Тачни/#Пронађени                    (1)
```

Одзив је однос броја тачних издвојених ентитета и укупног броја тачних ентитета, који се налазе у свим подацима:

```
Одзив(r)=#Тачни пронађени/#Тачни                   (2)
```

Уобичајено, Прецизност и Одзив се комбинују у једну меру, названу „***F*-мера**“. То је тежински просек Прецизности и Одзива. Најчешће се Прецизност и Одзив узимају са истим тежинама, и у том случају се добија балансирана *F*-мера (*F1*-мера) [34]:

```
F1 = (2*p*r) / (r+p)                               (3)
```

Имајући у виду да је у овом раду урађена екстракција свих језичких израза, а да су затим из ток скупа издвојени и посматрани само они језички изрази који се користе за повезивање, у том случају је Прецизност екстрактора, према формули (1):

```
Прецизност(p)=1
```

Да би било јасније како је добијена максимална вредност за овај параметар, треба подсетити да је на почетку добијен скуп различитих језичких израза, што значи да се ти језички изрази свакако налазе у текстовима прописа. Након тога су из тог скупа мануелно издвојени они изрази за које се зна да се користе за повезивање, а на начин како је то претходно описано. Дакле, међу пронађеним језичким изразима који се користе за повезивање, не постоји ни један који се не налази у текстовима прописа, и не постоји ни један који се не користи за повезивање. Због тога вредност за Прецизност има максималну вредност.

Са друге стране, на примеру посматраног скупа података и према формули (2), добијена вредност за Одзив је:

```
Одзив(r)=0.9955
```

У пробном скупу прописа, у коме је био познат број линкова између њих и који је износио 3111 линкова, успешно је откривено 3097 линкова. Ови линкови су откривени помоћу језичких израза за повезивање, а који су пронађени на раније описан начин. Као што се може видети, у пробном скупу прописа је било и 14 веза између прописа, а који нису откривени, што је грешка од 0.45% У следећој табели су дати језички изрази који нису детектовани у пробном скупу прописа, број њиховог појављивања и одговарајуће везе ка другим законима.

Табела 16 Језички изрази за повезивање са другим законима, а који нису откривени

Језички израз	**Број појављивања**	**Веза са законом**
„Стечајни закон“	1	Закон о стечају
„Грађанскога парничнога поступника“	1	Закон о парничном поступку
„Прелазни споразум о трговини и трговинским питањима“	1	Закон о потврђивању Прелазног споразума о трговини и трговинским питањима између Европске заједнице, са једне стране, и Републике Србије, са друге стране
„ако је пацијент глувонем“	2	Закон о употреби знаковног језика
„право на поверљивост свих личних информација“	1	Закон о заштити података о личности
„право увида у медицинску документацију“	1	Закон о здравственој документацији и евиденцијама у области здравства

Језички израз	Број појављивања	Веза са законом
„дискриминација на основу менталних сметњи“	1	Закон о забрани дискриминације
„спадају у податке о личности“	2	Закон о заштити података о личности
„у интересу јавног здравља“	1	Закон о јавном здрављу
„Подаци из медицинске документације и евиденције“	2	Закон о здравственој документацији и евиденцијама у области здравства
„Медицинска евиденција и документација“	1	Закон о здравственој документацији и евиденцијама у области здравства
	Укупно: 14	

На основу добијених вредности, а према формули (3), може да се израчуна и вредност за *F1*-меру:

```
F1 = (2*1*0.9955) / (0.9955+1) = 0.9977
```

На основу добијених вредности, може да се закључи да су сви пронађени ентитети истовремено и тачни, док је однос броја пронађених ентитета и укупног броја ентитета мањи од 100%. Језички изрази, који се користе за повезивање или референцирање, су ручно препознати међу оним пронађеним језичким изразима који имају велику фрекфенцију појављивања у посматраном скупу прописа и ти језички изрази су издвојени из тог скупа. Свакако, на овај начин су изостављени неки језички изрази, који се заиста користе за повезивање, а који имају малу фрекфенцију коришћења. Због тога је и вредност која је добијена за Одзив, мања од 100%. Ова вредност се може повећати прецизнијом претрагом и издвајањем и оних језичких израза који имају малу фрекфенцију коришћења, а који се такође користе за повезивање.

6 Примена неуронских мрежа за предвиђање језичких израза за повезивање у текстовима закона

Перформансе класичних модела за извлачење информација из текстова увелико зависе од ручно израђених функција, које захтевају пуно познавања домена и дубоко разумевање лингвистике. Последњих година, у истраживањима у овој области почињу да доминирају методе „дубоког учења“, за које није потребна опсежна примена ручних функција. Доњи слојеви неуронских мрежа уче оптималну репрезентацију својстава, док виши слојеви делују као коначни класификатор. Тако да је данас само изазов око избора или изградње исправне неуронске мрежне архитектуре и прикупљања пуно података за обуку.

У овом поглављу ће бити описано прављење модела и примена неуронских мрежа, као могућег решења за потребе машинског предвиђања постојања веза или референци у текстовима нових закона и других прописа. Обучавање и валидација неуронских мрежа ће бити обављени на обележеном скупу података, који је направљен тако што је сваком сегменту текста закона (сваком члану закона) придружена одговарајућа ознака о постојању, или не постојању, везе или референце у том сегменту текста. Циљ овог обучавања је могућност касније примене обучених модела за потребе проналажења или предвиђања веза или референци на необележеним подацима, нпр. у члановима нових закона.

6.1 Машинско учење из текстуалних података

Модели дубоког учења (ен. *Deep learning models*) су последњих година постигли невероватне резултате у рачунарском виду (ен. *Computer vision*), области истраживања чији је основни задатак, развијање техника које помажу рачунарима да „виде“ или разумеју садржај дигиталних слика или видео снимака и пропознавању говора (ен. *speech recognition*). У оквиру обраде природних језика, велики део рада са методама дубоког учења укључује учење векторског представљања речи путем неуронских модела језика и затим примену алгоритама за класификацију, над наученим векторима речи. Вектори речи се преко скривеног слоја пројектују на векторски простор ниже димензије и то су у суштини екстрактори својстава који кодирају семантичка својства [58].

Текст је један од најраспрострањенијих секвенцијалних података и као такав је погодан за примену модела дубоког учења из секвенцијалних података. Дубоко учење преко обраде природног језика је пропознавање образаца, примењено на речи, реченице и пасусе, на готово исти начин као што се примењује препознавање образаца на пикселима [59]. Неуронске мреже, као и други модели дубоког учења не узимају као улаз сиров текст, већ раде само са нумеричким **тензорима**. Примери математичих објеката, у које спадају и тензори су:

- Скалар: `3`
- Вектор: `[3,4,5]`
- Матрица: `[[3,4],[5,6],[7,8]]`
- Тензор:
```
[[[1,2], [3,4]],
 [[5,6], [7,8]]]
```

Векторизација текста је процес претварања текста у нумеричке тензоре и то може да се уради на више начина [59]:

- Сегментација текста у речи и трансформација сваке речи у вектор;
- Сегментација текста у карактере и трансформација сваког знака у вектор;
- Издвајање више узастопних речи или карактера и њихова трансформација у вектор.

Заједнички назив који се користи за све ове јединице на које текст може да се подели је: **токени**, а процес поделе је **токенизација**. У овом раду у поступку токенизације је примењена сегментација текста у речи, односно, трансформација речи у вектор. У примеру скупа података који су предмет овог рада, добијено је 120961 јединствених токена.

За потребе прављења модела за предвиђање постојања језичких израза за повезивање у текстовима нових закона и других прописа, у овом раду ће бити коришћени модели неуронских мрежа за класификацију текстова. Циљ овога је да се текстови аутоматски класификују у једну или више претходно дефинисаних класа, уз употребу неколико различитих приступа, где је имплементација заснована на библиотеци отвореног кода „Керас“ (ен. *Keras, https://keras.io/*). Да би се користила билиотека „Керас“ за текстуалне податке, најпре је потребно те податке претходно обрадити. За потребе токенизације се користи керасова класа „*Tokenizer*“. Овај објекат узима као

аргумент максималан број речи које се чувају након токенизације, на основу њихове учесталости, нпр:

```
MAX_NB_WORDS = 20000
tokenizer = Tokenizer (num_words=MAX_NB_WORDS)
tokenizer.fit_on_texts(texts)
```

Након што се токенизер примени на податке, може да се користи за претварање текстова у низове бројева. Ови бројеви представљају положај сваке речи у речнику.

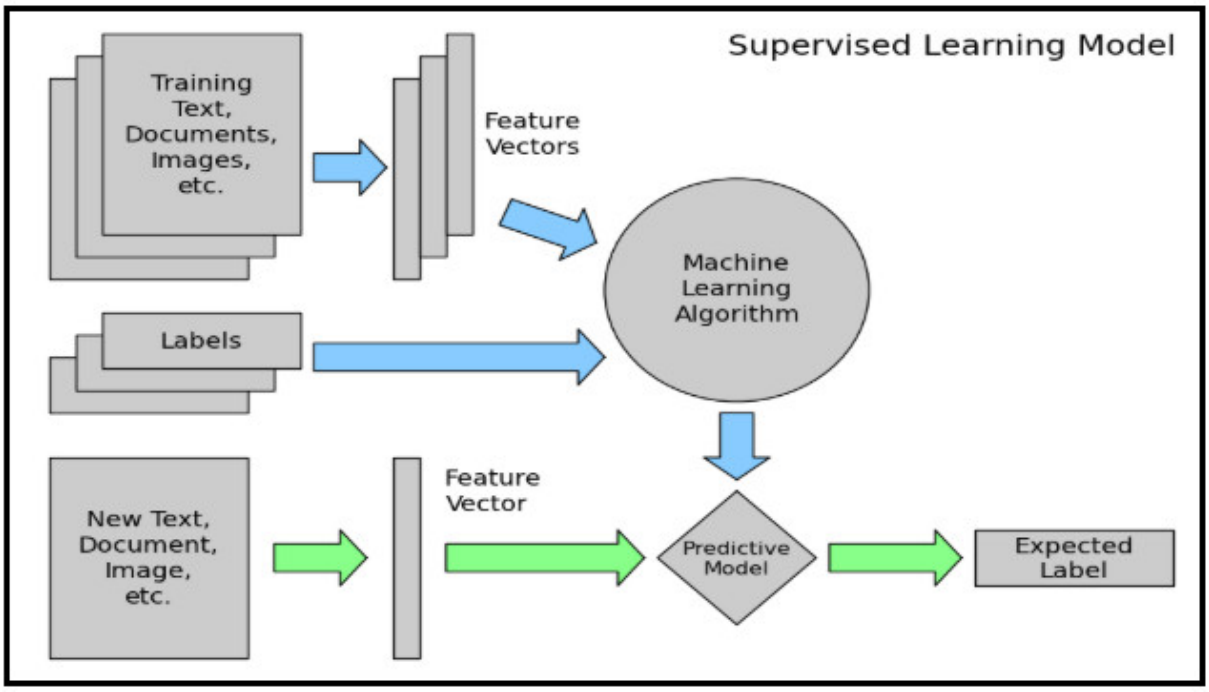
На основу језичких израза за повезивање, који су на претходно описани начин пронађни у посматраној колекцији, прави се обележни скуп података за обуку. Обележени скуп података је направљен тако што је скуп прикупљених текстова закона сегментиран на поједине чланове закона, при чему је сваки члан представљен једним записом у бази података. Скуп је обележен тако што је сваком оваквом сегменту (члану или запису у бази) придружена једна од ознака из следеће табеле:

Табела 17 Ознаке текстова за потребе надгледаног машинског учења

Ознака	**Опис**
0	Без референцирања
1	Ауторефернцирање
2	Референцирање са другим законима
3	Референцирање са међународним законима или прописима
4	Референцирање са нижим правним актима

Након тога се обавља поступак обучавања на основу великог скупа података, који обухвата колекцију 1120 текстова закона, сегментираних на укупно 59167 текстова појединих чланова закона. Затим се врши валидација модела над подацима за валидацију, који су такође обележени, и на тај начин се проверава тачност модела, односно врши се провера у којој мери се предвиђена класа текстова поклапа са стварном класом текстова. Подела скупа података, на податке за обуку и податке за валидацију је направљена у односу 80% према 20%. Дакле, валидација модела је рађена помоћу одвојеног скупа података, зато што је скуп за обучавање ионако изузетно велики. Из тог разлога, у овом раду није коришћен поступак „Унакрсне валидације“ (ен. *Cross-validation*)

Како подаци за обуку пружају информације алгоритму, због тога се овај модел сврстава у надгледано машинско учење.



Слика 2 Надгледано машинско учење[1]

6.1.1 Матрица за уградњу

Један од популарних начина за повезивање вектора са речима је употреба густих вектора речи, који се назива „уграђивање речи“ (ен. *word embeddings*). Ово су вектори чије вредности су са покретним зарезом, ниске су димензије, тј. то су густи вектори. За разлику од ретких вектора, у њима је спаковано више информација у далеко мање димензија и они се уче из података [59]. Постоје два начина за добијање ових вектора:

- Да се научи „уметање речи“, заједно са главним задатком који је предмет истраживања. У овом подешавању, започиње се са случајним векторима речи, а затим се учи вектор речи, на исти начин као што се уче тежински коефициенти неуронске мреже;
- Да се у свој модел учита „уметање речи“, који је претходно израчунат, употребом неког другог задатка машинског учења. То се зове унапред обучено уметање речи (ен. *pretrained word embeddings*)

[1] Слика је преузета са адресе https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f

Најједноставнији начин повезивања густог вектора са речима је одабир вектора са случајним вредностима (ен. *random*). Проблем овог приступа је што резултујући простор за уградњу нема структуру и што је тешко да дубока неуронска мрежа схвати тако неструктуиран простор [59]. Оно што чини добар простор за уметање речи, у великој мери зависи од истраживања које треба да се спроведе. Савршен простор за уметање речи за анализу расположења везаних за филмове, може да изгледа другачије од савршеног простора за уградњу за модел који се бави класификацијом правних докумената, јер важност одређених семантичких односа варира од задатка до задатка и разумно је да се научи нови простор за уградњу за сваки нови задатак. Срећом, повратно ширење (ен. *backpropagation*) и библиотека „Керас“ то чини лакшим. Ради се о учењу тежина слоја за **слој уградње** (ен. *Embedding layer*), нпр.:

```
from keras.layers import Embedding
embedding_layer = Embedding(1000, 64)
```

Слој уградње има два аргумента, а у претходном примеру је:

1000 – број могућих токена

64- димензионалност уградње

Према [59], слој уградње узима као улаз 2Д тензор целих бројева, облика (примери, дужина секвенце), при чему је сваки унос, низ целих бројева. Све секвенце у групи морају да имају исту дужину (јер морају да се спакују у један тензор), тако да секвенце које су краће од осталих треба да буду попуњене нулама, а дуже секвенце треба да буду скраћене. Овај слој враћа 3Д тензор са вредностима са покретним зарезом, облика (примери, дужина секвенце, димензионалност уградње). Такав 3Д тензор затим може да се обради у слоју рекурентне неуронске мреже или у 1Д конволуционом слоју.

На почетку, у слоју уградње су тежине случајне вредности, баш као и код било ког другог слоја. Током обуке, ови вектори речи се постепено прилагођавају и претварају простор у нешто што нижи слојеви могу да искористе. Након потпуне обуке, слој за уградњу ће приказати велику структуираност – врсту структуре која је специјализована за специфични проблем за који се тренира модел. У ситуацијама када се располаже са мало података за обуку, који не могу да се користе за учење одговарајућег вокабулара за одређени задатак, тада могу да се учитају уграђени вектори из унапред израчунатог простора за уградњу, за који се зна да је високо структуиран, да показује корисна својства и да бележи генеричке аспекте језичке структуре. У оваквим случајевима има смисла да

се поново користе својства која су научена на другом проблему. Таква својства уметања речи се обично рачунају коришћењем статистике појављивања речи, користећи различите технике, од којих неке укључују неуронске мреже, а неке друге технике их не укључују.

Постоје разне унапред израчунате базе података са својствима уграђених речи, које могу да се преузму и да се користе да би се направио индекс који пресликава речи (као стрингове) у њихову векторску презентацију (као нумеричке векторе). Затим се праве **матрице за уградњу**, које могу да се учитају у **слој за уградњу**. То мора да буде матрица чије су димензије (*максималан број речи, димензионалност уградње*), где сваки унос садржи вектор димензије за реч са индексом *i* у индексу референтних речи.

Поред тога, слој за уградњу може да буде „замрзнут“, тако што се његов атрибут *trainable* подеси да буде „*False*“, чиме се спречава ажурирање тежина тог слоја, током обуке модела. Ако се то не уради, онда ће тежине који су претходно научене, бити модификоване током обуке.

Такође, модел може да се обучава без учитавања претходно обучених речи за уградњу и без „замрзавања“ слоја за уградњу. У том случају, се током процеса обуке уче својства токена за уградњу која су специфична за неки посебан задатак. Овај начин је генерално моћнији од унапред уграђених својстава речи за уградњу, када је на располагању много података.

За потребе овог рада, направљен је и примењен алгоритам за израду матрице за уградњу. Разлози за ово су следећи:

- На располагању је већа количина текстуалних података, која обухвата колекцију 1120 текстова закона, сегментираних на укупно 59167 текстова појединих чланова закона, који су специфични за област која је предмет овог рада, што омогућава прављење овакве матрице;
- прављењем матрице за уградњу се добро пресликава семантички однос између речи, а који може да варира од задатка до задатка,
- приликом обуке неуронских мрежа са учитаном матрицом за уградњу и са замрзавањем слоја за уградњу се значајно смањује број параметара за обуку и на тај начин се убрзава обучавање модела

Параметри који су примењени приликом прављења матрице за уградњу су:

- Димензија матрице за уградњу је `emb_dim = 400`,

- посматра се 5 речи пре и после речи која се индексира,
- у матрици за уградњу ће бити само речи које се у текстовима појављују најмање 5 пута,
- у односу на посматрану реч, највише 15 других речи које имају негативна својства у односу на њу,
- број итерација је 5,
- број процеса је онолики колико има процесора.

Као резултат овога, добијена је претходно обучена матрица за уградњу, која је димензија 43654x400, што значи да је за 43645 речи направљен вектор димензије 400.

6.1.2 Оптимизација

Оптимизација неуронских мрежа је генерално тежак задатак. Једно од кључних питања за оптимизацију је „**ширење градијента**“ (ен. *gradient propagation*) кроз слојеве неуронских мрежа. Утицај повртног сигнала, који се користи за обучавање неуронских мрежа, слаби како се број слојева повећава. Зато се користи неколико важних алгоритамских побољшања која омогућавају боље ширење градијента [59]:

- Примена бољих активационих функција;
- Примена бољих шема за иницијализацију коефициената и
- Примена бољих шема за оптимизацију, као што су *RMSProp* и *Adam*.

У овом раду ће бити примењена следећа алгоритамска побољшања:

- Као активациона функција ће бити коришћена функција ***ReLU*** (ен. *Rectified Linear Unit*)
- Направљена је и биће примењена сопствена шема за иницијализацију тежина кроз „матрицу за уградњу“, а замрзавањем слоја за уградњу се значајно смањује број параметара за обуку;
- Као шема за оптимизацију ће бити коришћен алгоритам „*Adam*“, зато што је на примеру података који се користе у овом раду, овај алгоритам дао бољу конвергенцију тачности и губитака, него алгоритам „*RMSProp*“.

6.2 Примена неуронских мрежа за потребе учења и предвиђања језичких израза за повезивање, у текстовима закона

По узору на експеримент у коме су за потребе класификовањања текстова коришћене неуронске мреже [36], и у овом раду је вршена обука следећих неуронских мрежа:

- Рекурентна неуронска мрежа (ен. *Recurrent Neural Network, RNN*)
- Конволуциона неуронска мрежа (ен. *Convolutional Neural Network, CNN*)
- Хијерархијска мрежа са уграђеним механизмом пажње (ен. *Hierarchical Attention Network, HAN*)

Експеримент је структуиран на следећи начин:

1. **Текст за обуку** (ен. *Training text*), или улазни текст, користи се да модел надгледаног машинског учења може да буде обучен за предвиђање класе текста Обухвата 80% укупне колекције која се састоји од 1120 текстова закона, сегментираних на укупно 59167 текстова појединих чланова закона;
2. **Вектор својстава** (ен. *Feature Vector*), садржи информације које описују карактеристике улазних података. Облик улазног скупа података је вектор димензија 59167x2, који је у односу 80%:20% подељен на податке за обуку и податке за валидацију. Након обучавања и примене матрице за уградњу, улазни скуп података се трансформише у матрицу димензија 43654x400, што значи да је за 43645 речи направљен вектор димензије 400. Претходно је број јединствених токена од 120961 смањен на 43645 речи, тако што се посматрају само речи које се у текстовима појављују најмање 5 пута
3. **Ознаке** (ен. *Labels*) су предефинисане класе које модел треба да предвиђа. Према подацима који се налазе у *Табели 17*, модел треба да предвиђа припадност сваког сегмента текста, једној од 5 класа.
4. **Алгоритам машинског учења**, је алгоритам помоћу кога модел може да се бави класификацијом текстова, а у овом раду то су Рекурентне (или понављајуће) неуронске мреже (*RNN),* Конволуционе неуронске мреже *(CNN)* и Хијерархијске неуронске мреже са уграђеним механизмом пажње (*HAN)*;
5. **Предиктивни модел** (ен. *Predictive Model*), који је раније обучен и који може да обавља предвиђање ознака.

У наставку је дат приказ обуке наведених неуронских мрежа на примеру скупа текстова закона, по узору на малопре цитирани експеримент.

6.2.1 Класификација текстова употребом Рекурентне неуронске мреже

Рекурентне неуронске мреже (ен. *Recurrent Neural Networks, RNN*) су дизајниране за секвенцијалне податке попут текстуалних реченица, временских серија и других дискретних низова, попут биолошких секвенци. Рад са текстуалним подацима су и најчешћи случајеви употребе рекурентних неуронских мрежа, а ово су неки од примера примене [60]:

- Улаз може да буде низ речи, а излаз може да буде исти низ речи, увећан за једну реч, што омогућава да се у било којој тачки текста предвиђа следећа реч. То је класичан језички модел у коме се покушава предвиђање следеће речи, на основу секвенцијалне историје речи;
- Улаз може да буде реченица на једном језику, а излаз може да буде реченица на другом језику. У овом случају, могу да се повежу две рекурентне неуронске мреже, како би научиле моделе превођења између два језика. Даље, могу се повезати рекурентне неуронске мреже са другом врстом мрежа (нпр. са конволуционим неуронским мрежама), како би се научили наслови слика;
- Улаз може да буде низ речи (нпр. реченица), а излаз може да буде вектор вероватноће припадности класи. Овај приступ се користи за потребе класификације, као нпр. за анализу осећања. У овом раду ће бити коришћен овакав модел.

Рекурентне неуронске мреже су конструисане са идејом да се моделују зависности међу инстанцама [61]. На пример, врста неке речи може да зависи од врсте неке од претходних речи, јер врсте претходних речи носе информацију о врсти наредних речи. На пример, не очекује се низ од три глагола. Међутим, обично није познато колико претходних елемената секвенце носи информацију о наредном елементу. Рекурентне неуронске мреже превазилазе овај проблем тако што се елементи улазне секвенце обрађују у корацима. Кључна поента ових неуронских мрежа је постојање сопствене петље која проузрокује да се скривено стање неуронске мреже промени након сваког уноса. Скривено стање акумулира информације о елементима секвенце обрађеним у претходним корацима, а параметри одређују на који начин се то стање мења из корака у

корак, на основу претходног стања и текућих улаза и како се генерише излаз мреже у зависности од текућег стања. Приликом ажурирања параметара током процеса учења, користи се алгоритам повратног ширења (ен. *backpropagation algorithm*) да би се одредило да ли параметри треба да се повећавају или да се смањују. Због рекурзивне природе, рекурентне неуронске мреже имају могућност израчунавања за улазе промењиве дужине.

У пракси, међутим, рекурентне неуронске мреже нису у могућности да ефикасно користе сву историју уноса [62], због проблема са ишчезавањем (ен. *vanishing*). Боље решење за дугорочно искоришћавање контекста је „*Long Short-Term Memory*“ или „*LSTM*“ архитектура [63]. „*LSTM*“ је концептуално дефинисан као рекурентна неуронска мрежа, али се ажурирања скривеног слоја замењују посебним јединицама које се називају „меморијске ћелије“ (ен. *memory cells*) [64]. По узору на примену која је описана у [36], у овом раду је примењена двосмерна рекурентна мрежа [65] са уграђеним „*LSTM*“ слојем. Основна идеја двосмерних рекурентних мрежа је да се свака секвенца за обуку прослеђује напред и добија назад у две одвојене рекурентне мреже које су повезане у исти излазни слој, што значи да за сваку тачку одређене секвенце оваква неуронска мрежа има комплетне узастопне информације о свим тачкама пре и после ње [66]. За потребе овог рада коришћена је следећа архитектура:

```
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedded_sequences = embedding_layer(sequence_input)
l_lstm = Bidirectional(LSTM(20))(embedded_sequences)
preds = Dense(len(macronum), activation='softmax')(l_lstm)
model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['acc'])
```

Табела 18 Графички приказ архитектуре Рекурентне неуронске мреже

Слој (тип)	**Улаз/Излаз**	**Облик**
input_1 (InputLayer)	Улаз:	(None, 400)
	Излаз:	(None, 400)
↓		
embedding_1 (Embedding)	Улаз:	(None, 400)
	Излаз:	(None, 400, 400)
↓		
bidirectional_1 (Bidirection, LSTM)	Улаз:	(None, 400, 400)
	Излаз:	(None, 100)

Слој (тип)	Улаз/Излаз	Облик
↓		
dense_1 (Dense)	Улаз:	(None, 100)
	Излаз:	(None, 5)

Укупан број параметара ове неуронске мреже је 48,565,705.

Број параметара за које се врши обука: 180,905

Број параметара за које се не врши обука: 48,384,800

6.2.2 Класификација текстова употребом Конволуционе неуронске мреже

Конволуционе неуронске мреже (ен. *Convolutional neural networks*) су инспирисане биолошким неуронским мрежама и најчешће се користе у рачунарској визији (ен. *Computer vision*) за препознавање слика, откривање или детекцију објеката, али се користе и за обраду текстова [60]. Ове мреже су дизајниране за рад са мрежно-структуираним улазима (ен. *grid-structured inputs*), у којима постоје просторне зависности у локалним областима мреже. Очигледан пример мрежно структуираних података је дводимензионална слика. Ипак, остали облици секвенциалних података, као што су текст, временске серије и низови, такође могу да се сматрају посебним случајевима мрежно структуираних података са различитим врстама односа међу суседним ставкама.

Важна карактеристика која дефинише Конволуционе неуронске мреже је операција која се назива „конволуција“ (ен. *convolution*). **Конволуција** је скаларни производ између мрежно-структуираног (ен. *grid-structured*) скупа параметара и сличног мрежно-структуираног улаза, извученог из различитих просторних локалитета у улазној запремини [60]. Због тога су конволуционе неуронске мреже дефинисане као мреже које операцију конволуције користе најмање у једном слоју, мада већина конволуционих неуронских мрежа ову операцију користи у више слојева.

Слој конволуционе неуронске мреже је основна градивна компоненета и саставни део мреже. Постоје следећи типови слојева:

- **Улазни слој** је слој који је примењив на сваку неуронску мрежу и путем овог слоја се подаци уводе у мрежу. За улазни слој, вредности се одређују на основу природе улазних података и њихове претходне обраде;
- **Конволуциони слој**, обавља многа веома захтевна израчунавања и реализује основну операцију обучавања неурона мреже;

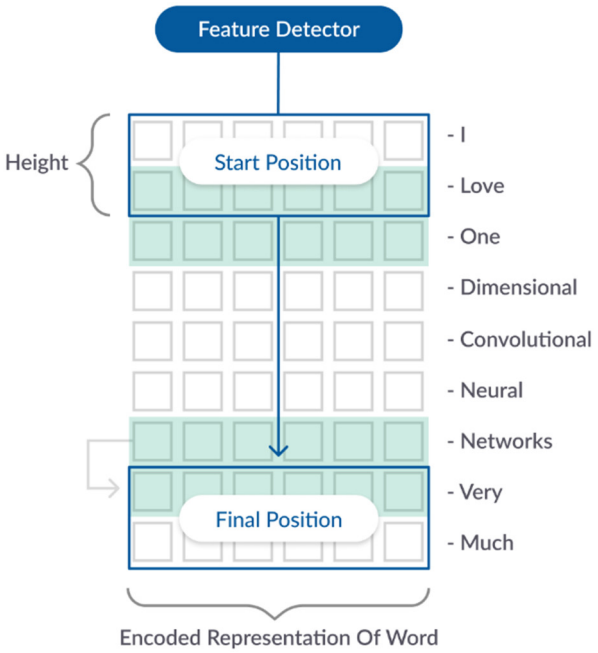
- **Слој сажимања** (ен. *pooling layer*) се уобичајено умеће између сукцесивних конволуционих слојева јер се применом операције сажимања смањује број параметара, а самим тим се смањује и комплексност израчунавања унутар мреже. Поред најчешће коришћених сажимања, као што су сажимање усредњавањем или сажимање максимумом, постоји и већи број других метода сажимања, који се примењују у Конволуционим неуронским мрежама;
- **Слој активационе функције** - прави избор активационе функције може значајно да побољша перформансе мреже. Једна од најчешће коришћених активационих функција је *ReLU* (ен. *Rectified Linear Unit*)
- **Потпуно повезани слој** (ен. *FC layer*, а понекад се назива и *Dense* слој), као што му име говори је слој у коме су сви неурони у овом слоју потпуно повезани са свим излазима претходног слоја. Типично за овај слој је да се користи као последњи слој;

Иако је препоручени начин машинског учења из текстуалних секвенци онај који се јавља у рекурентним неуронским мрежама, употреба конволуционих неуронских мрежа постаје све популарнија у последње време. Неки од разлога зашто конволуционе неуронске мреже на први поглед не изгледају као природно погодне за послове који се баве истраживањем текстова су:

- Када се конволуционе неуронске мреже користе за рад са сликама, облици који се налазе на сликама се тумаче на исти начин, без обзира на то где се на слици налазе. То није случај са текстовима, зато што је положај речи у реченицама прилично важан.
- Питања, попут транслације позиције и измене слика, не могу да се третирају на исти начин у текстуалним подацима. Суседни пиксели на слици су обично врло слични, док суседне речи у тексту готови никада нису исте.

Упркос овим разликама, системи који су засновани на конволуционим неуронским мрежама су последњих година показали побољшане перформансе. Баш као што је слика представљена као дводимензионални објекат са додатном димензијом дубине, која је дефинисана бројем канала са бојама, текстуална секвенца је представљена као једнодимензионални објекат са дубином која је одређена њеном димензионалношћу представљања [60]. Приликом рада са текстовима, уместо тродимензионалних „кутија“ које се примењују у раду са сликама, филтери за текстуалне податке су

дводимензионалне „кутије“, чије димензије су дужина секвенце која се помера дуж текста и дубина.



Слика 3 Пример поделе текста на секвенце[2]

На слици 3 је приказан пример дводимензионалне „кутије“, чија дужина секвенце је 2, а дубина 6. У овом примеру, „кутија“ се помера 8 пута да би се обрадила цела реченица.

У примеру који је описан у [36] је објашњено да ће се резултат сваке конволуције активирати када се препозна посебан образац. Променом величине кернела и повезивањем излаза се омогућава детектовање образаца различитих величина, састављених од више речи. Обрасци могу да буду језички изрази и зато конволуционе мреже могу да их препознају у реченици, без обзира на њихов положај.

У овом раду је примењена конволуциона архитектура која користи 128 филтера величине 5 и максималног сажимања 5 и 11:

```
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedded_sequences = embedding_layer(sequence_input)
l_cov1= Conv1D(128, 5, activation='relu')(embedded_sequences)
l_pool1 = MaxPooling1D(5)(l_cov1)
l_cov2 = Conv1D(128, 5, activation='relu')(l_pool1)
```

[2] Слика је преузета са адресе https://missinglink.ai/guides/keras/keras-conv1d-working-1d-convolutional-neural-networks-keras/

```
l_pool2 = MaxPooling1D(5)(l_cov2)
l_cov3 = Conv1D(128, 5, activation='relu')(l_pool2)
l_pool3 = MaxPooling1D(11)(l_cov3)  # global max pooling
l_flat = Flatten()(l_pool3)
l_dense = Dense(128, activation='relu')(l_flat)
preds = Dense(len(macronum), activation='softmax')(l_dense)
```

Табела 19 Графички приказ архитектуре Конволуционе неуронске мреже

Слој (тип)	**Улаз/Излаз**	**Облик**
input_1 (InputLayer)	Улаз:	(None, 400)
	Излаз:	(None, 400)
↓		
embedding_1 (Embedding)	Улаз:	(None, 400)
	Излаз:	(None, 400, 3)
↓		
conv1d_1 (Conv1D)	Улаз:	(None, 400, 3)
	Излаз:	(None, 396, 128)
↓		
max_pooling1d_1 (MaxPoolingD)	Улаз:	(None, 396, 128)
	Излаз:	(None, 79, 128)
↓		
conv1d_2 (Conv1D)	Улаз:	(None, 79, 128)
	Излаз:	(None, 75, 128)
↓		
max_pooling1d_2 (MaxPoolingD)	Улаз:	(None, 75, 128)
	Излаз:	(None, 15, 128)
↓		
conv1d_3 (Conv1D)	Улаз:	(None, 15, 128)
	Излаз:	(None, 11, 128)
↓		
max_pooling1d_3 (MaxPoolingD)	Улаз:	(None, 11, 128)
	Излаз:	(None, 1, 128)
↓		
flatten_1 (Flatten)	Улаз:	(None, 1, 128)
	Излаз:	(None, 128)
↓		
dense_1 (Dense)	Улаз:	(None, 128)
	Излаз:	(None, 128)
↓		
dense_2 (Dense)	Улаз:	(None, 128)
	Излаз:	(None, 5)

Укупан број параметара ове неуронске мреже је 48,822,181

Број параметара за које се врши обука: 437,381

Број параметара за које се не врши обука: 48,384,800

6.2.3 Класификација текстова употребом Хијерархијске неуронске мреже са уграђеним моделом пажње

Људска бића не уче из конкретног појма из података за обуку. Учење код људи је непрекидан процес, вођен искуством, у коме се доносе одлуке, а награде или казне које се примају од околине се користе за усмеравање процеса учења на будуће одлуке [60]. Другим речима, учење код интелигентних бића је вођено наградним процесом покушаја и грешака. Осим тога, велики део људске интелигенције и инстинкта је кодиран генетиком, која се развијала милионима година у још једном процесу који је вођен окружењем и који се назива еволуција. Стога, скоро сва биолошка интелигенција, како је знамо, потиче у једном или другом облику кроз интерактиван процес покушаја и грешака са околином [60].

Наградни процес покушаја и грешака, у коме систем учи да комуницира са сложеним окружењем ради постизања корисних резултата, у машинском учењу се назива **подстицајно учење** (ен. *reinforcement learning*). У подстицајном учењу, процес покушаја и грешака је вођен потребом да се током времена максимизирају очекиване награде. Подстицајно учење омогућава стварање истински интелигентних агената или општих облика вештачке интелигенције.

У својој књизи [60], аутор тзв. „**моделе пажње**“ (ен. *Attention models*) сврстава у напредне теме дубоког машинског учења. У било ком тренутку, људи не користе активно све информације које су им доступне из околине. Уместо тога, усредсређени су на одређене делове података који су им релевантни за одређени задатак. Овај биолошки феномен се назива „пажњом“. Сличан принцип може да се примени и на апликације за вештачку интелигенцију. Модели пажње користе подстицајно учење, или друге методе, да би се фокусирали на мање делове података, који су релевантни за задатак који се обавља. Такве методе се од недавно користе и за побољшање перформанси неуронских мрежа.

Пример примене модела пажње су слике које су снимљене у склопу система „*Google Streetview*“, који је створио Гугл, како би омогућио проналажење слика разних улица у многим земљама. Овде је било потребно да се нађе начин да се слике кућа повежу за кућним бројевима унутар неке улице. Иако се може евидентирати кућни број током снимања слике, ове податке треба извући са слике. Кључно је овде да се велика слика предњег дела неке куће систематично фокусира на мале делове слике, како би се

пронашло оно што се тражи. Главни изазов је да не постоји начин идентификовања релевантног дела слике, са информацијама које су унапред доступне. Стога је потребан итеративан приступ у претраживању одређених делова слике уз коришћење знања стеченог из претходних итерација.

Модели пажње су такође врло погодни за обраду природног језика, у којој се информације које се траже, скривају у дугачком сегменту текста. Овај проблем се често појављује у апликацијама попут машинског превођења и система за одговарање на питања. Рекурентна неуронска мрежа често не може да се усредсреди на одређене делове изворне реченице за превођење у циљану реченицу. У таквим случајевима је корисно да се ускладе циљана реченица са одговарајућим деловима изворне реченице, током превођења. У таквим случајевима, механизми пажње су корисни за изоловање релевантних делова изворне реченице, истовремено стварајући одређени део циљане реченице.

Овај процес може да се посматра као својеврсни **хијерархијски инежењеринг својстава** (ен. *hierarchical feature engineering*). Ова врста понашања је посебно видљива у неким доменима попут Конволуционих неуронских мрежа за сликовне податке. У Конволуционим неуронским мрежама својства у старијим слојевима обухватају детаљне, али примитивне облике, док својства у каснијим слојевима добијају облике веће сложености. Овакви, семантички интерпретативни облици, често имају ближе корелације са ознакама класа у домену слике, што олакшава класификацију. Овај општи принцип спајања једноставних својстава за стварање сложенијих својстава, лежи у сржи успеха који су постигнути са неуронским мрежама.

Хијерархијске неуронске мреже са механизмом пажње су постигле изванредне перформансе за класификацију докумената на датом језику [67]. Овај тип неуронских мрежа је предложен да се примењује за класификацију докумената [68], као модел који има следеће карактеристике:

- Има хијерархијску структуру, која одражава хијерархијску структуру докумената;
- Поседује два нивоа механизама пажње, који се примењују на нивоу реченице и који омогућавају различито присуство за више или за мање важан садржај приликом конструкције репрезентације документа.

Интуиција која стоји у основи овог модела је да нису сви делови документа подједнако релевантни за одговор на упит и да одређивање релевантних секција

укључује моделовање интеракције речи, а не само њихово присуство. Архитектура која је осмишљена прикупља два основна увида у структуру документа. Као прво, пошто документи имају хијерархијску структуру (речи формирају реченице, а реченице формирају документ), по овом моделу се приказ документа конструише тако што се прво изгради приказ реченица, а затим се ти прикази обједињују у приказ документа. Као друго, примећено је да су различите речи и реченице у документима, различито информативне природе. Поврх тога, значај речи и реченица зависи од контекста, тј. иста реч или реченица могу да буду различито важни у различитом контексту. Да би се укључила осетљивост на ту чињеницу, овај модел укључује два нивоа механизма пажње, један на нивоу речи и други на нивоу реченице. На тај начин је моделу омогућено да обрати више или мање пажње на поједине речи или реченице приликом конструкције репрезентације документа. Кључна новина у овом приступу је да овај систем користи контекст да би открио када је редослед токена релевантан, а не да једноставно филтрира секвенце токена, извађених из контекста. Експерименти, које су аутори [68] спровели, су показали да предложена архитектура знатно надмашује претходне методе. Визуелизација слојева пажње илуструје да модел бира квалитативно информативне речи и реченице.

По узору на примену која је описана у [36], да би се направила хијерархијска неуронска мрежа, уместо улазних података, који су у претходна два примера били у 2 димензије, овде су улазни подаци у 3 димензије. Након тога се користи Керасова функција (ен. *Keras, https://keras.io/*) ***TimeDistributed***, функција која захтева улазне податке у 3 димензије:

```
embedding_layer = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix],
                            input_length=MAX_SENT_LENGTH)
sentence_input = Input(shape=(MAX_SENT_LENGTH,), dtype='int32')
embedded_sequences = embedding_layer(sentence_input)
#l_lstm = Bidirectional(LSTM(100))(embedded_sequences)
l_lstm = Bidirectional(LSTM(20))(embedded_sequences)
sentEncoder = Model(sentence_input, l_lstm)

review_input = Input(shape=(MAX_SENTS,MAX_SENT_LENGTH), dtype='int32')
review_encoder = TimeDistributed(sentEncoder)(review_input)
```

```
l_lstm_sent = Bidirectional(LSTM(20))(review_encoder)
preds = Dense(len(macronum), activation='softmax')(l_lstm_sent)
model = Model(review_input, preds)
```

Табела 20 Графички приказ архитектуре Хијерархијске неуронске мреже са уграђеним моделом пажње

Слој (тип)	Улаз/Излаз	Облик
input_1 (InputLayer)	Улаз:	(None, 10, 100)
	Излаз:	(None, 10, 100)
↓		
time_distributed_1 (TimeDistributed)	Улаз:	(None, 10, 100)
	Излаз:	(None, 10, 100)
↓		
bidirectional_1 (Bidirectiontional, LSTM)	Улаз:	(None, 10, 100)
	Излаз:	(None, 40)
↓		
dense_1 (Dense)	Улаз:	(None, 40)
	Излаз:	(None, 5)

Укупан број параметара ове неуронске мреже је 48,584,765.

Број параметара за које се врши обука: 199,965.

Број параметара за које се не врши обука: 48,384,800.

6.2.4 Резултати обучавања неуронских мрежа за потребе учења и предвиђања језичких израза за повезивање, у текстовима закона

На истом скупу података је обучавано 3 различита модела неуронских мрежа, за потребе учења и предвиђања језичких израза за повезивање, у посматраном скупу текстова закона. Све три неуронске мреже су обучавана у 5 епоха, из разлога што је обучавање извршено на кућном рачунару, просечних карактеристика (CPU: *AMD Ryzen 3 2200G 3.5 GHz*, RAM: 8GB). У следећој табели је дат приказ добијених резултата и времена које је било потребно за обуку ових неуронских мрежа:

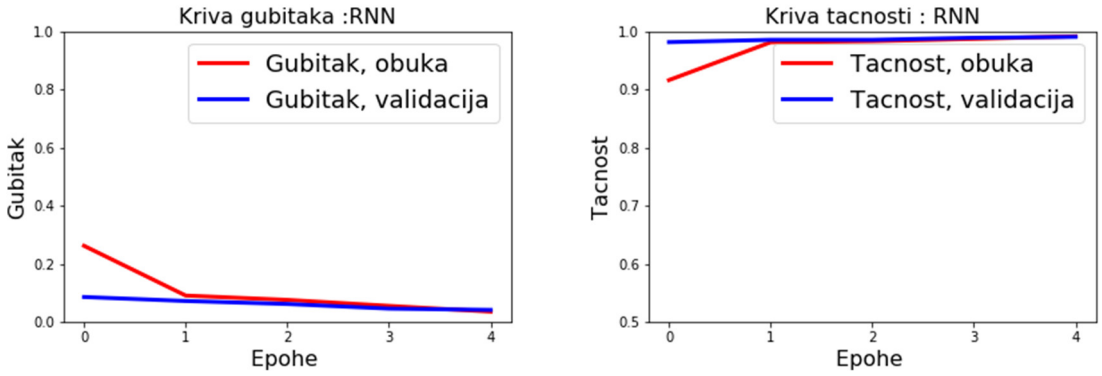
Табела 21 Резултати обучавања неуронских мрежа

Алгоритам	Укупно време за обуку са 5 епоха	Просечно време по епохи	Тачност обуке	Тачност валидације
RNN	~7h 8min (25637s)	~1h 26min (5127.4s)	99.11%	99.02%
CNN	~49.18min (2951s)	~9.84min (590.2s)	95.44%	95.614%
HAN	~3h 38 min (13073s)	~43.58min (2614.6s)	65.81%	72.323%

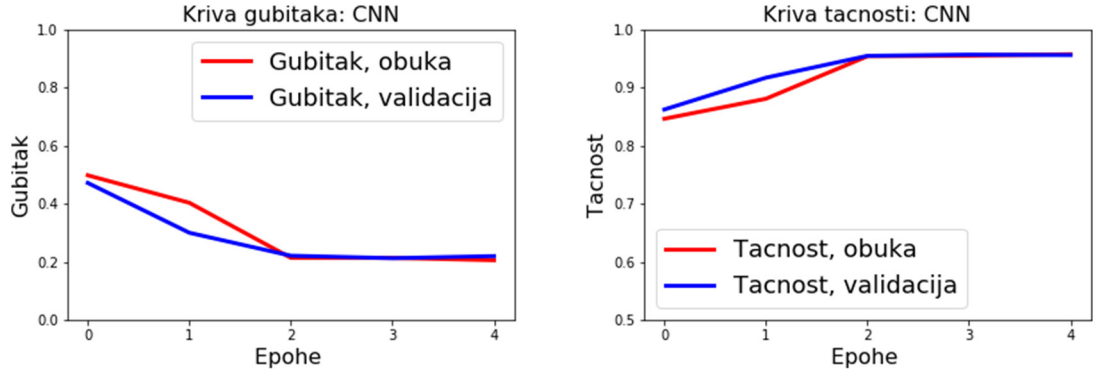
На основу података из претходне табеле о обуци претходно описаних неуронских мрежа на истом посматраном скупу података, може се приметити следеће:

- Највећа тачност валидације је постигнута обуком Рекурентне неуронске мреже, која износи 99.02%, док је најмања тачност валидације постигнута обуком Хијерархијске неуронске мреже са уграђеним механизмом пажње. Исти однос важи и за тачност обуке;
- Најмање време је било потребно за обуку Конволуционе неуронске мреже, које је просечно по епохи износило нешто мање од 10 минута, на рачунару са раније наведеним карактеристикама, а надуже време је потребно за обуку Рекурентне неуронске мреже, које је просечно по епохи износило око 90 минута;

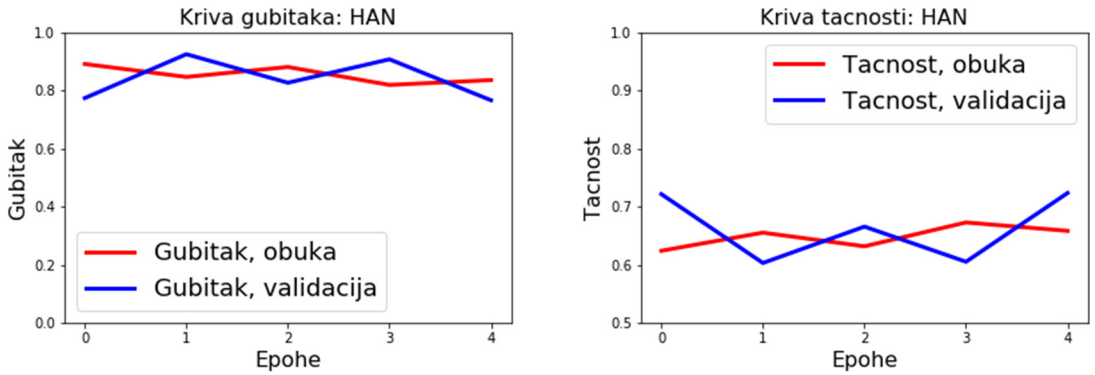
У наставку су дати графикони за губитке и тачност, по епохама, током обуке претходно описаних неуронских мрежа на истом посматраном скупу података:



Слика 4 Криве губитака и тачности, по епохама, током обуке Рекурентне неуронске мреже



Слика 5 Криве губитака и тачности, по епохама, током обуке Конволуционе неуронске мреже



Слика 6 Криве губитака и тачности, по епохама, током обуке Хијерархијске неуронске мреже

На основу приказаних резултата обуке, по епохама, може се приметити да Ниво тачности од преко 90% врло брзо постиже Рекурентна неуронска мрежа, већ у првој епохи, док Конволуциона неуронска мрежа овај ниво тачности постиже у трећој епохи. Ниво тачности за обе ове неуронске мреже конвергирају жељеној вредности, с'тим што Ниво губитака Конволуционе неуронске мреже, од друге епохе не показује конвергенцију. Са друге стране подаци који су за првих пет епоха добијени за Хијерархијску неуронске мреже са уграђеним механизмом пажње, са архитектуром која је примењена, не показују конвергенцију током обуке. Да би се то исправило, потребно је спровођење даљих експеримената са архитектуром и са подешавањима параметара, како ове, тако и осталих неуронских мрежа. Осим тога, могуће је и даље побољшање перформанси неуронских мрежа, што се може постићи ако се уради нешто од следећег:

- Фино подешавање параметара за приказане архитектуре неуронских мрежа
- Експериментисање са другачијим архитектурама неуроских мрежа и евентуално увођење нових слојева, нпр. *Dropout* слоја, уз пажљиво подешавање параметара овог слоја;
- Боља припрема улазних података за обуку.

6.2.5 Предвиђање израза за повезивање у текстовима нових закона применом обучених модела

Циљ обучавања неуронских мрежа је могућност касније примене обучених модела за потребе предвиђања на необележеним подацима, а у областима за које је вршено обучавање. На примеру текста закона који се не налази у посматраном скупу података, а на коме ће се показати примена предвиђања израза за повезивање, применом

обучених модела, послужиће тест „Закона о науци и истраживањима“, који је донет 8. јула 2019. године и који је објављен у Службеном гласнику Републике Србије, број 49/19.

Укратко, и на начин како је то раније у овом документу описано, текст овог закона је припремљен на следећи начин:

- Са веб сајта Народне скупштине Републике Србије (http://www.parlament.gov.rs), у секцији „Акти“, линк „Донети закони“, је пронађен текст овог закона, у **.DOCX* формату;
- Документ је конвертован у **.TXT* формат,
- Урађена је сегментација текста, тако да сваки члан закона буде засебан текст и то је сачувано у **.CSV* формату;
- Овако сачувани подаци су учитани применом програмског језика *Python* и припремљени на начин на који су већ припремани текстови за потребе обучавања неуронских мрежа;
- Учитан је обучени модел неуронске мреже помоћу команде
  `load_model = load_model('model-save.hdf5');`
- Урађено је предвиђање, тачније, примена обученог модела на подацима текста новог закона, уз напомену да је обучавање модела неуронских мрежа обављено на латиничним текстовима закона, па је приликом примене предвиђања потребно да се учита латинична верзија текста;
- На послетку су резултати предвиђања, за све три обучене неуронске мреже, извезени из програма.

Провером добијених резултата, а у складу са добијеним резултатима за тачност валидације, резултати који су добијени применом раније описане Хијерархијске неуронске мреже са уграђеним механизмом пажње, нису били задовољавајући и нису даље разматрани. Увидом у резултате, који су добијеним применом модела који су обучени за откривање језичких израза за повезивање, употребом Рекурентне и Конволуционе неуронске мреже, примењеног на текст новог закона (Закона о науци и истраживањима), односно примењеног на закон који нема обележене податке, може се приметити нешто од следећег:

- Ове две неуронске мреже су дале исто предвиђање за 92.5%, а за 7.5% текстова се предвиђања ове две мреже разликују;

- У посматраном тексту, обучени модел Рекурентне неуронске мреже је предвидео да се у два сегмента закона налазе везе ка нижим правним актима (класа 4), где је провером установљено да се налазе језички изрази „*у складу са актом*“ и „*у складу са статутом*“, док обучени модел Конволуциона неуронске мреже ове везе није предвидео
- Ни један од ових обучених модела није предвидео везе ка међународним законима или прописима (класа 3), а у тексту посматраног закона се налазе језички изрази „*примени међународних стандарада*“ и „*у складу да потврђеним међународним уговором*“, који могу да се посматрају као везе из класе 3;
- Што се тиче предвиђања веза за аутореференцирање (класа 1) и веза ка другим законима (класа 2), у 4.8% случајева је обучени модел Рекурентне неуронске мреже предвидео постојање веза, а да модел Коволуционе неуронске мреже није предвидео њихово постојање. Постоје и примери у којима су обучени модели предвидели постојање различитих класа за повезивање. Један такав пример у тексту посматраног закона је језички израз „у складу са правним схватањем Националног савета“.

У моделу који користи обучене моделе неуронских мрежа за предвиђање припадности неког текста или неког сегмента текста, некој одређеној класи, ако се такав модел примењује за откривање веза у текстовима закона и других прописа, изазов за одређивање класе представља ситуација када се у истом тексту или у истом сегменту текста налази више од једне везе. То значи да текст или сегмент текста може да има припадност више од једној класи.

Помоћу команде `preds = load_model.predict(n_data)` се добија **матрица**, која за сваки посматрани сегмент текста закона даје **тежине**, тачније процене припадности класама које су на почетку дефинисане. Управо ове вредности су веома значајне за откривање веза у текстовима закона, у случајевима када се у истом сегменту текста налази више различитих веза. Ипак, првобитно је у овом истаживању за неки сегмент текста посматраног закона предвиђана припадност једној класи, тако што су узете највеће вредности из добијене **матрице тежина**, помоћу команде `predicted = np.argmax(preds, axis=1)` . Примери у којима се у неком сегменту текста налази више различитих типова веза, а у посматраном тексту закона су језички изрази као нпр. „у складу са законом и статутом“ или „у складу са Статутом заједнице и са овим

законом“, где се налази ауторефернца (класа 1), референца ка другим законима (класа 2) или референца са нижим правним актима (класа 4). Управо за овакве ситуације, решење може да буде коришћење матрице која за сваки посматрани сегмент текста закона даје коефициенте (или вероватноће), тачније процене припадности једној од класа, а без издвајања класе са највећим коефициентом, тачније без предвиђања припадности само једној класи. За два примера, који су малопре наведени, вектори који се добијају из матрице тежина, која је добијена обуком Рекуренте неуронске мреже су:

- Пример 1, „у складу са законом и статутом“:
  [0.033% 0.415% 99.322% 0.006% 0.224%]
- Пример 2, „у складу са Статутом заједнице и са овим законом“
  [0.047% 99.203% 0.107% 0.001% 0.642%]

Бројеви представљају процену припадности посматраног сегмента текста, класама које су на почетку дефинисане. То значи да је обучени модел неуронске мреже предвидео да сегмент текста у коме се налази језички израз из првог примера, са проценом од 99.322% припада класи „Референцирање са другим законом“, а у одговарајућем проценту осталим класама. За сегмент текста у коме се налази језички израз из другог примера, обучени модел је предвидео са проценом од 99.203% да припада класи „Ауторефернцирање“ и са проценом од 0.642% да припада класи „Референцирање са нижим правним актима.

Применом овако добијене матрице тежина и даљом анализом добијених вредности је могуће предвиђање постојања више различитих типова веза у посматраним текстовима. Осим тога, поновном обуком модела са циљем препознавања и предвиђања језичких израза за повезивање, а које овако обучени модели нису препознали би се даље побољшали резултати оваквог система.

7 Примена теорије графова на правне документе и везе између њих

Скуп текстуалних података, који су предмет овог рада, и скуп веза које су откривене, су погодни да се на тај модел примене методе и алати који спадају у теорију графова. Према теорији графова [69], сваки граф се састоји од чворова, а чворови могу, али и не морају да буду повезани један са другим.

Да би се на неки реални модел применили алгоритми и алати, који спадају у теорију графова, најпре је потребно направити разлику између различитих типова графова и на посматрани модел применити одговарајући тип графа. Прва подела графова се односи на тип веза између чворова. Везе које спајају два чвора су потези или гране и оне могу да буду усмерене или неусмерене. У примеру података у овом раду, очигледно је да су везе између прописа усмерене, па је тако и граф који оне формирају, **усмерен**. Према конвенцији, за усмерени граф, први чвор у пару је почетни чвор, а други чвор је завршни чвор гране. У случају усмереног графа, када постоји линија **i** → **j**, за чвор **j** се каже да је следбеник чвора, а са друге стране, за чвор **i** се каже да је претходник чвора **j**.

Друга подела графова се односи на то да ли чворови и везе имају неке атрибуте. На пример, ако везе имају нумеричке атрибуте, који се називају „тежински фактори“ или „пондери“, онда је и граф „пондерисан“ или „тежински“. Упримеру података у овом раду, коришћен је модел веза без „тежинских фактора“. Трећа подела графова се односи на то да ли у моделу постоје паралелне везе, односно, да ли постоји више од једне везе између истог пара чворова. Иако је јасно да је у моделу који описује прописе и везе између њих, могуће постојање паралелних веза између прописа (у ситуацијама када се текст једног прописа више пута референцира на неки други пропис), у овом раду је коришћен модел у коме се посматра само постојање везе између прописа, а не и колико таквих веза има.

Постоје бројни алгоритми који могу да се примене на графове. Неки од њих су израчунавање улазног и излазног степена чворова, анализа путања, проналажење елементарних путања, пречника графа, проналажење и анализа циклуса, проналажење чворова веће важности и многи други алгоритми. У наставку ће бити показана примена неких од алгоритама.

У овом раду, за потребе анализе графа, коришћен је пакет „NetworkX“ (https://networkx.github.io/). Овај пакет је погодан за прављење, манипулацију и проучавање структуре, динамике и фукција сложених мрежа, односно графова.

7.1 Анализа путања унутар посматраног скупа прописа

Путања од чвора **i** до чвора **j** је низ различитих чворова који почињу са **v** и завршавају се са **w**, тако да су узастопни чворови суседни [70]. У усмереном графу, усмерена путања је такође низ који повезује чворове, али уз додатно ограничење да су линије усмерене у истом правцу [71].

7.1.1 Проналажење елементарних путања

Следећи циљ овог рада је да се пронађу све „елементарне“ или „једноставне путање“ унутар посматраног скупа прописа. „**Елементарна путања**“ је путања у којој нема понављања чворова, односно, сви чворови су различити. Другим речима, у елементарној путањи се сваки чвор појављује само једном [57].

Да би се у овом моделу података поједноставила анализа путања, посматране су само везе између прописа, а није рађена анализа конкретних чланова прописа, који су линковани. За проналажење свих елементарних путања, примењен је следећи алгоритам:

```
for all node_source in Graph:
    for all node_target in Graph:
        find all simple paths(Graph) from node_source to node_target
        insert into table all simple path(Graph)
```

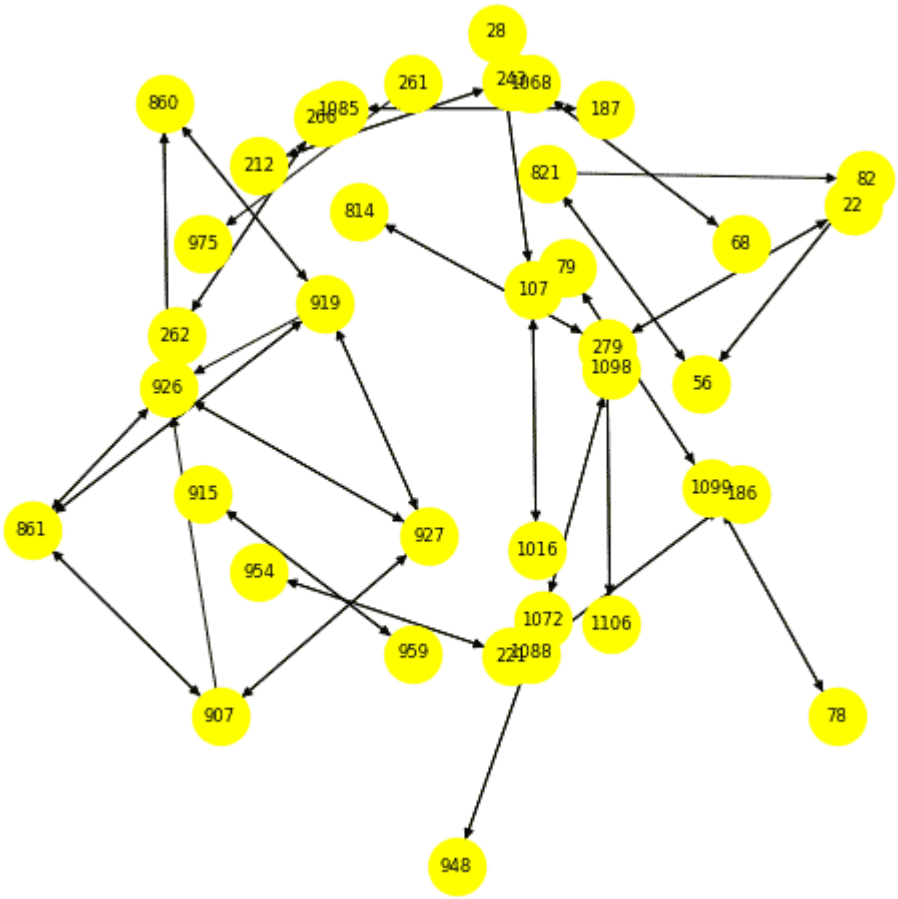
На овај начин је у посматраном скупу података и примењујући описани поступак, пронађено 210230 различитих путања између чворова у графу, односно између различитих прописа. Највећи број путања почиње од Закона о мировању и отпису дуга по основу доприноса за обавезно здравствено образовање, док највећи број путања води ка међународним законима, а на другом месту је Закон о општем управном поступку.

Уобичајено, за било која два чвора **i** и **j**, у неком усмереном графу, може да се одреди растојање између тих чворова као најкраћа путања од чвора **i** до чвора **j**. У овом моделу података је пронађено 57429 најкраће путање, чије дужине су од 2 до 14 чворова.

7.1.2 Проналажење циклуса у графу

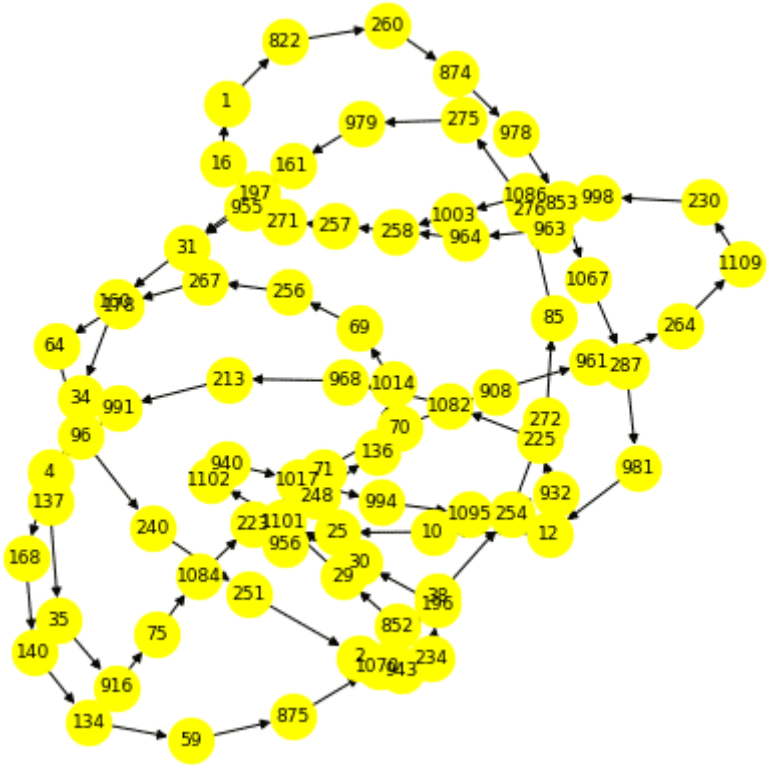
Циклус или контура (ен. *cycle*) је путања у графу која се завршава у истом чвору у коме и почиње. Према [72], део веза које не учествују у циклусима у усмереном графу се

назива „Хијерархија протока“ (ен. *Flow Hierarchy*). Израчунавајући „Хијерархију протока“ за граф који је предмет овог рада, добијена је вредност од 0.487. То значи да 48.7% веза не учествује у циклусима, односно да 51.3% веза учествује у стварању великог броја циклуса. На основу овога можемо да закључимо да модел текстова прописа, представљених као чворови у графу, заједно са откривеним везама између њих, је модел који обилује циклусима, односно, овај модел графа спада у тзв. „Усмерене цикличне графове“.



Слика 7 Субграф који приказује неке циклусе у графу

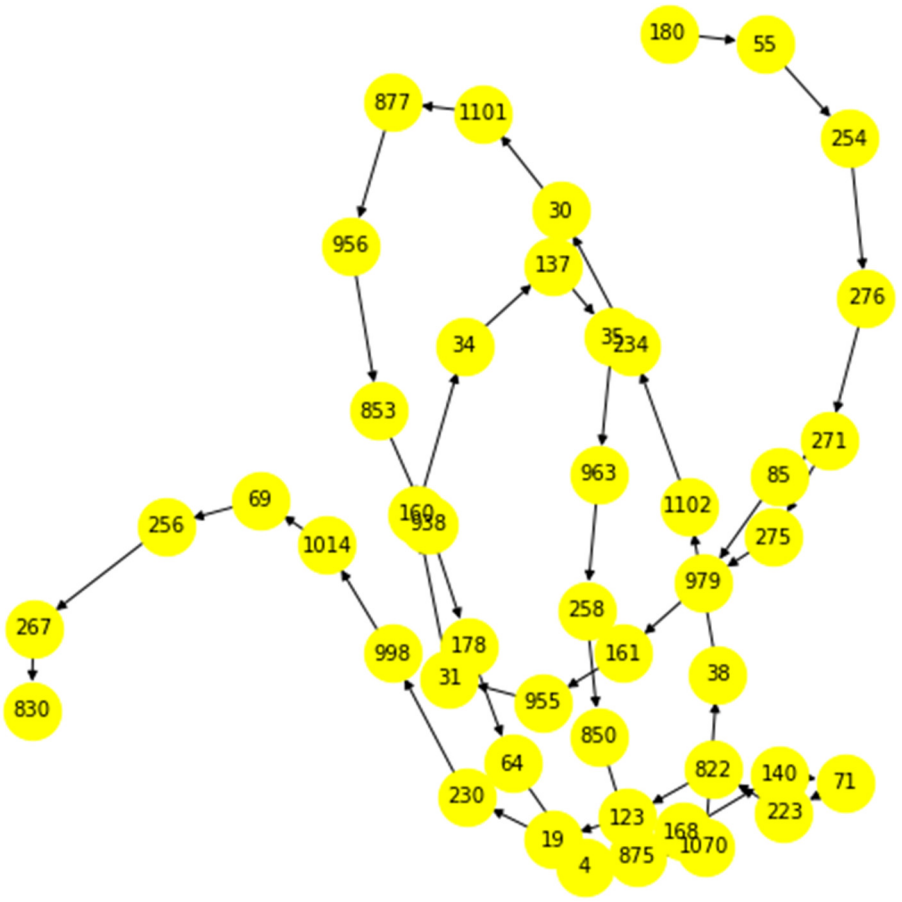
На Слици 7 је дат субграф, односно мали део графа који приказује неколико циклуса. У најједноставнијем облику, два чвора су међусобно повезана, као на пример, чворови са ознакама „915“ и „959“, па све до сложенијих циклуса у којима је већи број чворова повезан.



Слика 8 Субграф на коме су приказани циклуси који се састоје од 79 чворова

На Слици 8 је субграф на коме су приказани циклуси који почињу од чвора са ознаком „1“, затим путања садржи 78 чворова, завршава се у чвору са ознаком „16“ и након тога постоји веза са почетним чвором „1“.

Пречник неког графа је највеће растојање између два различита чвора [70] и према томе, пружа информацију о два најудаљенија чвора у графу. Међутим у графовима у којима постоје циклуси, вредност за највеће растојање између чворова може да буде бесконачно. У моделу података који је предмет овог рада, посвећена је пажња проналажењу најдужих елементарних или једноставних путања, односно, најдужих путања између појединих чворова, у којима нема понављања чворова. На пример, у посматраном скупу података најдуже елементарне путање између чворова 180 и 830, су путање које се састоје од 45 чворова, тачније од 45 различитих, међусобно повезаних закона, као што је приказано на следећој Слици 6. Као поређење, најкраћа путања између чворова 180 и 830 је дужине 7, односно састоји се од 5 чворова, који се налазе на путањи између ова два чвора.



Слика 9 Графички приказ две најдуже путање између чворова 180 и 830, које су пронађене у посматраном скупу прописа

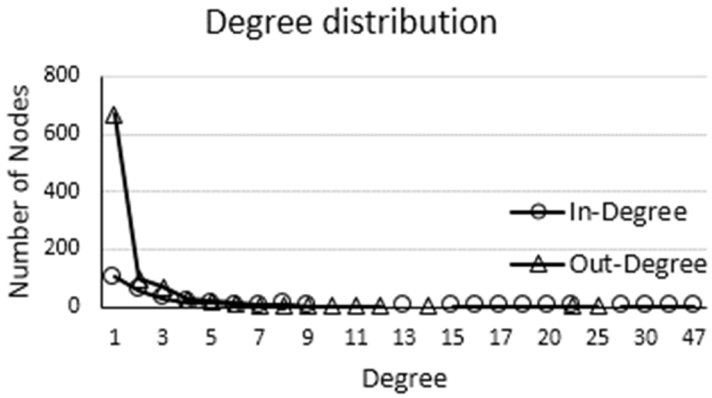
У овој путањи, почетни закон је Закон о мировању и оптпису дуга по основу доприноса за обавезно здравствено осигурање (у овом примеру, идентификатор овог закона је 180), а завршни закон је Закон о претварању друштвене својине на пољопривредном земљишту у друге облике својине (идентификатор 830). Ове две путање се разликују само по томе што у једном случају део путање иде од Закона о пореском поступку и пореској администрацији (271) ка Закону о порезу на додату вредност (275) а затим ка Закону о туризму (979), а у другом случају од Закона о пореском поступку и пореској администрацији (271) ка Закону о финансирању локалне самоуправе (85) и затим ка Закону о туризму (979).

7.2 Проналажење правних докумената „веће важности“ са становишта веза између њих

У Теорији графова су дефинисане многе мере престижа и централности чвора, а неки аутори те мере називају „важност“, „положај“, „истакутост“ или „популарност“

[57]. Када је граф усмерен, онда се говори о престижу и важности (термини који се користе наизменично).

У општем случају, **Степен поверења** (ен. *degree of confidence*) чвора **i** у чвор **j** је број референци из чвора **i** ка чвору **j** [57]. На примеру података о везама између закона, у посматраном скупу закона, у овом раду је степен поверења у неки закон **j** посматран као укупан број закона од којих постоје долазни линкови ка посматраном закону **j**. Дакле, **долазни степен поверења** (ен. *In-degree*) је број долазних линкова од других закона, а **одлазни степен поверења** (ен. *Out-degree*) је број одлазних линкова од тог закона, ка другим законима. На следећој слици је приказана расподела долазних и одлазних степени поверења у посматраном скупу закона.



Слика 10 Расподела долазних и одлазних степени поверења

Према приказаној расподели, може се видети да највећи број закона има мали долазни или одлазни степен поверења, односно да је број долазних и одлазних линкова ка тим законима или од тих закона, мали. Са друге стране, може се видети и да постоје закони који имају већи долазни или одлазни степен поверења. Највећи долазни степен поверења има Устав Републике Србије, који према подацима са којима располажемо, има 47 долазних линкова од других закона. Највећи одлазни степен поверења има Закон о пореском поступку и пореској администрацији, који према подацима са којима располажемо, има 25 одлазних линкова.

Развијени су бројни алгоритми за потребе одређивања мере значаја неког чвора у усмереном графу, а одређивање ове мере се најчешће одређује на основу броја цитата или потврда (бројем долазних линкова) ка том чвору. Према [57], у најпопуларније алгоритме за одређивање мере значаја неког чвора у усмереном графу, који се углавном примењују у друштвеним наукама, спадају алгоритми из групе за класична мерења важности чворова (ен. *Classic Node Prestige Measures*), као што су:

- Улазни степен чвора (ен. *Node Indegree*),
- Важност на основу непосредне близине (ен. *Prestige by Proximity*),
- Спектрална мера важности (ен. *A Spectral Measure of Prestige*) и
- Важност на основу индиректних веза (ен. *Prestige Based on Indirect Links*).

Према истом извору, у остале алгоритме, који су развијени углавном у контексту библиометрике или машина за претрагу Интернета, спадају: „*Citation Influence*“, „*Rating Methods Based on Least Squares*“, „*PageRank Algorithm*“, „*HITS: Hubs and Authorities*“, „*Probabilistic HITS*“ и „*A Simple Bag-of-Paths Prestige Measure*“.

У циљу проналажења правног документа веће важности, са становишта веза између њих, у овом раду су коришћени алгоритми који се углавном примењују у друштвеним наукама.

Улазни степен чвора (ен. *Node Indegree*) је најједноставнија мера важности неког чвора у графу и ова мера одговара броју долазних грана до чвора. Примењујући овај алгоритам на посматрани скуп података, највећу важност има Устав Републике Србије (1.00), а затим Закон о општем управном поступку (0.66), Закон о привредним друштвима (0.64), Закон о изменама закона којима су одређене новчане казне за привредне преступе и прекршаје (0.55), Закон о рачуноводству (0.45), Закон о заштити података о личности (0.43), Закон о државној управи (0.40), Закон о здравственој заштити (0.40), Закон о планирању и изградњи (0.40), Закон о буџетском систему (0.36), Закон о облигационим односима (0.34), итд. Вредности које су дате у заградама су релативне вредности, добијене према описаном алгоритму, где највећа вредност може да буде 1, док су све остале вредности приказане у односу на највећу вредност.

Важност на основу непосредне близине (ен. *Prestige by Proximity*) је алгоритам који мери степен до кога је неки чвор у графу директно или индиректно (тј. преко посредника) цитиран од стране других чвороврa у графу, при чему овај алгоритам квантификује значај региона утицаја чвора у усмереном графу [57]. Да би се израчунала важност на основу непосредне близине, неког чвора **j**, први корак је израчунавање „цене“ најкраће путање између свих парова чворова. Према овом алгоритму, неки чвор **j**, ће имати већи индекс важности ако велики број чворова мреже директно или индиректно цитира овај чвор, при чему је просечна удаљеност тих чворова од чвора **j** мала. Примењујући овај алгоритам, на посматрани скуп података, највећу важност има Закон о председнику Републике (1.00), а затим следе Закон о локалној самоуправи (1.00), Закон о Влади (1.00), Устав Републике Србије (0.70), Закон о наслеђивању (0.64), Закон о

спољнотрговинском пословању (0.64), Закон о тржишту капитала (0.64), Закон о парничном поступку (0.64), итд.

Спектрална мера важности (ен. *A Spectral Measure of Prestige*) је мера престижа, која је прилагођена за усмерене графове, која формулише да је важност чвора ***j*** пропорционална збиру тежина чворова који цитирају овај чвор, помножен важности тих чворова који га цитирају.. Према овом алгоритму, важност неког чвора ***j*** ће бити већа ако је тај чвор цитиран од значајног броја других чворова и ако чворови који цитирају чвор **j** сами имају велику важност. Идеја овог алгоритма је веома слична концепту алгоритма „*PageRank*“, који је касније развијен за потребе рангирања веб страница. Примењујући овај алгоритам на посматрани скуп података, највећу важност има Устав Републике Србије (1.00), а затим Закон о општем управном поступку (0.66), Закон о привредним друштвима (0.64), Закон о рачуноводству (0.47), Законо о заштити података о личности (0.43), Закон о планирању и изградњи (0.40), Законо о здравственој заштити (0.40), итд.

Важност на основу индиректних веза (ен. *Prestige Based on Indirect Links*) је алгоритам (или скуп алгоритама), који мере степен важности неког чвора уграфу, узимајући у обзир директне и индиректне линкове [57]. Према овом алгоритму, на важност неког чвора у графу највећи утицај имају директни линкови, затим мањи утицај имају индиректни линкови из другог корака, још мањи утицај имају индиректни линкови из трећег корака, итд. Код примене овог алгоритма, важну улогу игра фактор умањења α (при чему је $0<\alpha<1$), којим се мери умањење индиректних линкова из k-тог корака, тако што се утицај тих линкова множи са α^k. Када је фактор умањења α велики, утицај дугачних путања је само мало умањен и резултат има тенденцију да буде повезан са резултатом који се добија према алгоритму Спектралне мере важности. Обрнуто, када је фактор умањења α близу нуле, утицај дугачких путања на важност неког чвора се нагло смањује и добијени резултат има тенденцију да буде повезан са резултатом који се добија према алгоритму Улазног степена чвора. Како примена овог алгоритма уз употребу граничних вредности за фактор умањења α, има тенденцију добијања резултата као у претходно описаним алгоритмима, у овом раду није коришћен овај алгоритам. Истраживање о утицају фактора умањења важности индиректних линкова α, на мерење важности правних докумената, са становишта веза између њих, може да буде предмет неког будућег истраживања.

Осим употребе алгоритама за откривање чворова вешће важности у усмереним графовима, а који се примењују у друштвеним наукама, у овом раду је коришћен и

алгоритам **Утицај цитирања** (ен. *Citation Influence*) [57], који се у библиометрици углавном користи за мерење важности часописа. Основна идеја овог алгоритма је мерење баланса између долазних и одлазних линкова или цитата у сваком чвору у графу. Примењујући овај алгоритам на посматрани скуп података, највећу важност има Закон о општем управном поступку (1.00), а затим следе Закон о државној управи (0.61), Закон о рачуноводству (0.34), Закон о управним споровима (0.29), Закон о високом образовању (0.26), Законо о безбедности и здрављу на раду (0.26), Устав Републике Србије (0.25), итд.

Од алгоритама, који су развијени у контексту машина за претрагу Интернета, у на овом моделу података су примењени „*PageRank*“ [73], „*HITS: Hubs and Authorities*“ [74]. Применом алгоритма „*PageRank*“ се добија ранг листа веб страница (у овом случају, листа прописа), која је заснована на локацији сваког појединог документа у структури графа. Примењујући алгоритам „*PageRank*“, на посматрани скуп података, највећу важност има Устав Републике Србије (0.0559), а затим следе Закон о заштити података о личности (0.0416), Закон о општем управном поступку (0.0316), Закон о буџетском систему (0.0241), Закон о слободно приступу информација од јавног значаја (0.0223), Закон о тајности података (0.0203), Закон о рачуноводству (0.0200), итд. Вредности које су дате у заградама су апсолутне вредности, добијене према описаном алгоритму.

Алгоритам „*HITS: Hubs and Authorities*“ [74] додељује две оцене свакој веб страници (у овом случају пропису), од којих се једна оцена назива „*hub score*“, а друга „*authority score*“. Основна идеја је да се добију две листе рангираних резултата, у складу са идејом да постоје две врсте страница, оне које су ауторитативни извори информација и оне које садрже листу веза ка ауторитативним страницама. Примењујући алгоритам „*HITS: Hubs and Authorities*“, на посматрани скуп података, на врху добијене листе ауторитативних прописа се налази Устав Републике Србије (0.0925), а затим следе Закон о општем управном поступку (0.0477), Закон о привредним друштвима (0.0359), Закон о планирању и изградњи (0.0304), Закон о рачуноводству (0.0239), Закон о државној управи (0.0219), Закон о зашити података о личности (0.0202) итд. Применом истог алгоритма, на врху листе прописа који садрже везе ка „ауторитативним прописима“ се налази Закон о државном премеру и катастру (0.0170), Закон о пореском поступку и пореској администрацији (0.0167), Закон о утврђивању надлежности Аутономне покрајине Војводине (0.0142), Закон о националним саветима националних мањина (0.0137) итд.На основу добијених резултата, може се закључити да различити алгоритми,

који су примењени за потребе одређивања мере важности неког правног документа, са становишта веза између њих, а у посматраној збирци прописа, дају различите резултате. Ипак, имајући у виду да је у правном систему неке државе, Устав „кровни“ закон, из кога проистичу сви остали закони, може се закључити да алгоритми Улазни степен чвора и Спектрална мера важности дају баш такав резултат, док резултати добијени применом осталих алгоритама нису дали такав резултат.

8 Дискусија и будући рад

За било које истраживање текста, потребно је најпре да буду примењене методе и алати за припрему тих текстова за даље истраживање, другим речима, потребно је да буде урађено препроцесирање текстова. Методе и алати који се користе за ту намену, могу да се сврстају у две групе: оне који нису зависни од језика на којима су посматрани текстови и оне који јесу зависни од језика.

Примена метода и алата који нису зависни од језика на коме су написани текстови који су предмет истраживања, имају широку примену. Углавном се користе за потребе смањења вектора за мултидимензионално представљање текстова и њиховом употребом се постижу одговарајући резултати.

Проблем може да настане у примени метода и алата за препроцесирање који су зависни од језика на којима су текстови написани. За текстове који су на српском језику, према информацијама којима располажемо, не постоји јавно доступан алгоритам, ни табела за консолидацију токена која је заснована на коришћењу (ен. *Usage-Based Consolidation*). Што се тиче имплементације процеса за извлачење морфолошког корена речи, таквих имплементција постоји неколико, али још увек ни једна од њих не даје максималну тачност, нити је у општој употреби. На основу истраживања које је овде описано, дошло се до закључка да примена различитих метода за препроцесирање текстова на српском језику, за потребе одређивања сличности између њих, може да произведе разлике у добијеним резултатима.

Сва ова запажања указују на то да треба бити пажљив када се користе различите методологије и алати за препроцесирање текстова. Резултати који су добијени у овом раду би требало да олакшају унапређење поменутих алата, у циљу достизања нивоа у коме ће примена алата и методологија за истраживање тектова на српском језику довести до добијања уједначених резултата, без обзира не домен истраживања, изворе података и друге разлике.

У делу истраживања које се бави применом неуронских мрежа је најпре показано да је могуће направити модел и извршити обучавање неуронских мрежа, а за потребе проналажења, односно предвиђања постојања, језичких израза за повезивање у текстовима. Направљени су једноставнији модели и примењене су три различите архитектуре неуронских мрежа, чије обучавање је извршено на обележеном скупу података, а затим је извршена валидација модела. Након тога је извршено поређење ефикаснисти обуке и резултата, који су добијени. Иако је применом модела Рекурентне

неуронске мреже постигнута тачност валидације од преко 99%, предложени су начини за даље побољшање перформанси, како ове архитектуре, тако и других архитектура неуронских мрежа. Даљи рад у овој области подразумева спровођење даљих експеримената са архитектурама неуронских мрежа, експериментисање са подешавањем параметара мрежа, али и поновно обучавање модела на новим подацима и са новим информацијама о језичким изразима који се у текстовима закона и других прописа користе за повезивање, а који нису били раније препознати и неуронске мреже нису обучаване са тим информацијама.

У делу истраживања које се бави применом теорије графова је показано да постоје сви услови и да је могуће применити методе за анализу веза између правних докумената, као и за анализа путања. Осим добијених информација о законима од којих почиње највећи број линкова или ка којима води највећи број линкова, пронађене су и анализиране све „елементарне путање“ унутар посматраног скупа закона. Тако је откривено да у том скупу постоје путање које повезују неколико десетина закона. Осим тога, откривено је постојање великог броја циклуса, односно путања у графу које се завршавају у истом чвору у коме и почињу.

Постојање информације о систему веза између правних докумената има посебан значај када је потребно спровести измене у тим документима. Мењањем закона који је чвор неке дуже путање у графу, не утиче се само на област којом се тај закон бави, већ се утиче и на све оне законе и области којима се повезани закони баве, а који припадају истој путањи у графу. Промене у правним документима, тачније, измене и допуне, врше се када пропис треба ускладити са изменама у правном систему или изменама у политици у одређеној области или га треба прилагодити стварним потребама. Изменама и допунама неког прописа, по правилу се не могу извршити измене и допуне других прописа, али се изузетно може утврдити престанак важења појединих одредаба другог прописа [3]. Због тога је веома важно ефикасно сагледавање ефеката промене неког прописа, на све остале прописе. За сваки пропис се може претпоставити да постоје други прописи од којих или ка којима постоји упућивање или повезивање са посматраним документом. Самим тим, промене у неком пропису имају утицај на све повезане прописе. Успостављање система веза између правних докумената, закона и других прописа, може да се постигне ефикасно управљање променама у тим документима, имајући у виду сву међузависност између њих.

Осим тога, примењено је неколико алгоритама, који се баве мерењем значаја неког чвора у графу. Са становишта правне науке, постоји јасна хијерархија између прописа и према тој хијерархији сви закони имају једнаку важност и не треба их тумачити на начин да неки закон има мању или већу важност. Због тога, процена ефеката претходно наведених алгоритама је рађена само у односу на Устав. Ипак, са становишта веза између закона и применом наведених алгоритама за проналажење у графу чвора, који је веће важности, може се проценити утицај који могу да произведу измене неког од закона, када је потребно да до њих дође.

Што се тиче практичне примене аутоматског управљања комплексним информацијама које се налазе у правним документима, један од начина је да се у овој области стандардизује начин техничке припреме или кодификације свих нових прописа. На пример, постоје мета-шеме за семантичку репрезентацију правних докумената, попут "Akoma Ntoso" [32], који могу да се искористе. Уколико би се то урадило, тиме би се олакшало и управљање променама у хијерархији прописа, имајући у виду сву међузависност између њих. Што се тиче постојећих правних докумената, уз употребу неке од техника које спадају у истраживање текстова или које спадају у извлачење информација, потребно је такве документе технички припремити, тако да се омогући неки напреднији облик њихове аутоматске размене и повезивања, за потребе шире друштвене заједнице.

У овом истраживању је показано да се најчешће коришћени језички изрази у правним документима употребљавају за референцирање или повезивање. Али, поред њих свакако постоје и многи други језички изрази. Даље истраживање може да иде у два правца.

Један правац би се односио на даљу анализу веза на основу пронађених језичких израза. У овом раду су посматране само везе између појединих закона, а анализа веза ка члановима закона или ка ставовима појединих чланова, није била предмет овог рада. Свакако, у неком будућем истраживању, може да се обухвати и ово. Осим тога, у овом раду су обухваћени само текстови закона. У неком будућем истраживању могу да се обухвате и други прописи и нижи правни акти. У правним текстовима се често користе одреднице које дефинишу [3]

- однос између прописа који престају да важе и новог прописа у погледу њиховог дејства на случајеве, ситуације и односе који су настали за време важења ранијег прописа,

- поступање са предметима чије решавање је у току,
- рокове за доношење подзаконских прописа, као и овлашћења за доношење подзаконских прописа који су донети на основу ранијег прописа, као и потреба за изменама и доношењем нових подзаконских прописа на основу новог прописа,
- утврђивање повратног дејства после одредаба закона (ретроактивност),
- информације о посебним ограничењима у примени прописа у односу на време (временски ограничавајуће одредбе).

Дакле, може се направити даља анализа веза између законских и подзаконских прописа, чиме се добија хијерархија прописа, али и везе између прописа који су престали да важе и нових прописа.

Други правац будућег истраживања би могао да укључи издвајање информација из правних докумената, на начин који је описан у [45]. Међу језичким изразима који су пронађени уочавају се и језички изрази који омогућавају извлачење значења (ен. *extract meaning*) из правних докумената, затим језички изрази који се односе на временске одреднице, догађаје и друга обележја (мета податке) правних докумената и слично. Пробемом постојања различитих верзија закона су се у свом раду бавили [6]. У пракси, пропис ступа на снагу истеком одређеног рока након објављивања. Могуће је да су раздвојени почетак важења и почетак примене неког прописа, односно примене појединих његових одредаба, као што је могуће и постојање временског размака између дана ступања на снагу прописа и почетка његове примене, односно примење његових одредаба [3]. Ово су још неки од језичких израза, који су откривени у овом раду, а који се односе на временске одреднице и који могу да буду предмет неког будућег истраживања.

У језичке изразе, који су такође пронађени, а који омогућавају „извлачење значења“ из правних докумената, извлачење и анализу релевантних чињеница и слично, спадају:

- језички изрази којима се дефинишу права и обавезе правних субјеката,
- језички изрази којима се дефинишу овлашћења – одредбе о подзаконским прописима које треба донети ради спровођења закона,
- језички изрази којима се дефинишу казнене одредбе – наређујуће или забрањујуће норме,
- језички изрази којима се прописује у ком року се дешава неки правни статус, итд.

За потребе аутоматског управљања информацијама које се налазе у правним документима, издвајање и анализа језичких израза која је описана у овом раду, може да се користи и за проналажење обележја или метаподатака у текстовима правних докумената. Проблемом препознавања структуре и метаподатака правних докумената, на основу њиховог текста, и затим моделирање према мета-шеми, која се користи за семантичко представљање правних ресурса, су се бавили аутори [31] у свом раду. Метод који су описали, користи заједничке формате правних докумената за идентификацију блокова структурних и семантичких информација и моделира их према популарној правној мета-шеми. Сваки документ, па и правни документ, може да има већи број обележја о том документу. Обележја правних докумената се по правилу налазе у самом тексту прописа, у завршном делу, а који може да садржи прелазне и завршне одредбе. Као обележја која се односе на правне документе, могу се посматрати:

- датум подношења прописа,
- под којим бројем је пропис објављен или заведен,
- орган који је пропис донео,
- овлашћено лице које је потписало пропис и друго.

Као језички изрази који омогућавају одређивање метаподатака о томе где је неки закон објављен, најчешће коришћен језички израз је „закон је објављен у %“, где се ознака „%“ односи на назив јавног гласила у коме је закон објављен.

9 Закључак

Циљ овог рада није био тумачење прописа нити се ово истраживање бавило правном науком. У овом раду су примењене неке од техника и алгоритама из области истраживања текстова. На примеру збирке закона који се примењују на територији једне државе, најпре је урађено истраживање и проналажење језичких израза који се користе у текстовима закона, у намери да се даље анализирају добијене информације. За издвојене језичке изразе је утврђена фрекфенција њиховог појављивања у посматраним текстовима. На тај начин су добијени тачни подаци о томе који језички изрази се користе и у којој мери, а за израду текстова у посматрном скупу закона.

Међу пронађеним језичким изразима, одмах се примећује да се најчешће коришћени језички изрази употребљавају за дефинисање веза или референци са другим правним документима, и ти језички изрази су били предмет даљег истраживања. Поступак који је описан и примењен на посматрани скупа прописа, најпре показује да је могуће у текстовима закона открити референце ка текстовима других закона и прописа. Затим су показани неки од начина за аутоматско или машинско откривање поменутих веза или референци.

На основу добијених података, дефинисан је модел структуре колекције правних докумената и урађена је анализа те структуре. Поменута анализа обухвата анализу путања између повезаних правних докумената, као и одређивање мере значаја неког правног документа са становишта веза између њих. Овако добијене информације могу да буду корисне за процену утицаја који могу да произведу измене неког од закона, када је потребно да до њих дође, на све оне законе и области којима се повезани закони баве.

Број могућих начина примене истраживања језичких израза у правним документима свакако превазилази примере који су претходно набројани, а информације које могу да буду добијене на основу ове методологије су одлична основа и представљају улазне податке за многе даље анализе текстова у правним документима. Методологија, која је овде описана, може да се користи не само у наведеној области, већ и на другом скупу текстова, који су писани на другим језицима.

10 Литература

[1] J. Žižka, F. Dařena и A. Svoboda, Text Mining with Machine Learning: Principles and Techniques, CRC Press, 2020.

[2] A. Stranieri и J. Zeleznikow, Knowledge Discovery from Legal Databases, Springer, 2005.

[3] Законодавни одбор Народне скупштине РС, Јединствена методолошка правила за израду прописа, Службени гласник РС, 21/2010, 2010.

[4] F. Zimmermann, „Dublin Core and legal informatics, VoxPopuLII, jurMeta - New Metadata Initiative for Legal Documents,“ 2010. [На мрежи]. Available: https://blog.law.cornell.edu/voxpop/category/dublin-core-and-legal-informatics/. [Последњи приступ 25 01 2017].

[5] R. S. Wagh, "Exploratory Analysis of Legal Documents using Unsupervised Text Mining Techniques," International Journal of Engineering Research and Technology, vol. 3, no. 2, 2014.

[6] J. Garofalakis, K. Plessas и A. Plessas, „A Semi-automatic System for the Consolidation of Greek Legislative Texts,“ у Proceedings of the 20th Pan-Hellenic Conference on Informatics, Patras, Greece, ACM, 2016, pp. 1:1-1:6.

[7] Đ. Petrović и I. Stanišević, „Izvlačenje podataka sa Interneta i skladištenje u bazu, studija slučaja o tržištu polovnih automobila,“ у 25th Telecommunications forum TELFOR 2017, Beograd, Srbija, 2017.

[8] Đ. Petrović и M. Stanković, „The influence of text preprocessing methods and tools on calculating text similarity,“ Facta Universitatis, Series: Mathematics and Informatics, т. 34, бр. 5, pp. 973-994, 2019.

[9] Đ. Petrović и M. Stanković, „Use of linguistic forms mining in the link analysis of legal documents,“ Computer Science and Information Systems , т. 15, бр. 2, pp. 369-392, 2018.

[10] Đ. Petrović и S. Janićijević, „Domain Specific word Embedding Matrix for Training Neural Networks,“ у 2019 International Conference on Artificial Intelligence: Applications and Innovations, IC-AIAI 2019, Vrdnik Banja, Serbia, 2019.

[11] C. C. Aggarwal, Machine Learning for Text, Springer, 2018.

[12] D. Vitas, L. Popović, C. Krstev, I. Obradović, G. Pavlović-Lažetić и M. Stanojević, The serbian language in the digital age, Springer, Berlin, Heidelberg, 2012.

[13] E. Kajan, A. Pljasković и A. Crnišanin, „Normalizacija tekstualnih dokumenata na sprskom jeziku u cilju efikasnijeg pretraživanja u sistemima e-uprave,“ у ETRAN, Zlatibor, 2012.

[14] V. Kešelj и D. Šipka, „For the greedy and the optimal subsumption-based stemmer for Serbian: A Suffix Subsumption-Based Approach to Building Stemmers and Lemmatizers for Highly Inflectional Languages with Sparse Resources,“ Infotheca, т. 9, бр. 1-2, pp. 23a-33a, 2008.

[15] N. Milošević, „Stemmer for Serbian language,“ arXiv preprint arXiv:1209.4471, 2012.

[16] V. Batanović и B. Nikolić, „Sentiment Classification of Documents in Serbian: The Effects of Morphological Normalization and Word Embeddings,“ Telfor Journal, т. 9, бр. 2, 2017.

[17] N. Ljubešić, D. Boras и O. Kubelka, „Retrieving Information in Croatian: building a simple and effcient rule-based stemmer,“ у 1st International Conference The Future of Information Sciences (INFuture), Zagreb, 2007.

[18] Z. Ceska и C. Fox, „The Influence of Text Pre-processing on Plagiarism Detection,“ у International Conference RANLP, Borovets, Bulgaria, 2009.

[19] R. Alshammari, „Arabic Text Categorization using Machine Learning,“ International Journal of Advanced Computer Science and Applications, т. 9, бр. 3, pp. 226-230, 2018.

[20] J. Liu, J. Shang и J. Han, Phrase Mining from Massive Text and Its Applications, Morgan & Claypool, 2017.

[21] N. N. Karanikolas и C. Skourlas, „Text Classification: Forming Candidate Key-Phrases from Existing Shorter Ones,“ FACTA UNIVERSITATIS, т. 19, бр. 3, pp. 439-451, 2006.

[22] N. N. Karanikolas и C. Skourlas, „A parametric methodology for text classification,“ Journal of Information Science, т. 36, бр. 4, pp. 421-442, 2010.

[23] M. Saravanan, B. Ravindran и S. Raman, „Automatic Identification of Rhetorical Roles using Conditional Random Fields for Legal Document Summarization,“ 2008.

[24] A. R. Lodder и A. Oskamp, Information Technology and Lawyers: Advanced Technology in the Legal Domain, from Challenges to Daily Routine, Springer Science & Business Media, 2006.

[25] M. v. Opijnen, N. Verwer и J. Meijer, „Beyond the Experiment: The Extendable Legal Link Extractor," 2015.

[26] A. L. Monroy, H. Calvo, A. Gelbukh и G. G. Pacheco, „Link Analysis for Representing and Retrieving Legal Information," у Computational Linguistics and Intelligent Text Processing, A. Gelbukh, Ур., Springer Berlin Heidelberg, 2013, pp. 380-393.

[27] T. Neale, "Citation Analysis of Canadian Case Law," Journal of Open Access to Law, vol. 1, no. 1, 2013.

[28] M. J. Bommarito II и D. M. Katz, „Properties of the United States Code Citation Network," SSRN´s eLibrary, 2009.

[29] N. Sakhaee, M. C. Wilson и G. Zakeri, „New Zealand Legislation Network," у Legal Knowledge and Information Systems, IOS Press, 2016, pp. 199-202.

[30] B. Waltl, J. Landthaler и F. Matthes, „Differentiation and Empirical Analysis of Reference Types in Legal Documents," у Jurix: International Conference on Legal Knowledge and Information Systems, Sofia Antipolis, France, 2016.

[31] M. Koniaris, G. Papastefanatos и Y. Vassiliou, „Towards Automatic Structuring and Semantic Indexing of Legal Documents," у PCI'2016: Proceedings of the 20th Pan-Hellenic Conference on Informatics, Patras, Greece, 2016.

[32] G. Barabucci, L. Cervone, M. Palmirani, S. Peroni и F. Vitali, „Multi-layer Markup and Ontological Structures in Akoma Ntoso," у AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue, Springer, Berlin, Heidelberg, 2009, pp. 133-149.

[33] N. Vasiljević, „Automatic processing of legal text in Serbian language: doctoral dissertation," 29 06 2015. [На мрежи]. Available: https://phaidrabg.bg.ac.rs/detail_object/o:10687. [Последњи приступ 05 10 2017].

[34] G. Miner, J. Elder, A. Fast, T. Hill, R. Nisbet и D. Delen, Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications, Academic Press, 2012.

[35] B. Furlan, V. Batanović и B. Nikolić, „Semantic similarity of short texts in languages with a deficient natural language processing support,“ Decision Support Systems, т. 55, бр. 3, pp. 710-719, 2013.

[36] A. Maheshwari, „Report on Text Classification using CNN, RNN & HAN,“ 2018.. [На мрежи]. Available: https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f. [Последњи приступ 07. 2019.].

[37] T. Berners-Lee, J. Hendler и O. Lassila, „The Semantic Web,“ Scientific American Magazine, т. 284, 2001.

[38] R. Mitchell, Web Scraping with Python, Collecting Data from the Modern Web, O’Reilly Media, Inc., 2015.

[39] A. Mccallum, D. Freitag и F. Pereira, „Maximum Entropy Markov Models for Information Extraction and Segmentation,“ у ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, California, 2000.

[40] J. Lafferty, A. Mccallum и F. Pereira, „Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,“ Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282-289, 01 2001.

[41] B. Scholkopf и A. J. Smola, Learning with Kernels, Cambridge, MA, USA: MIT Press, 2001.

[42] Службени гласник РС, „Закон о високом образовању, Службени гласник Републике Србије, број 73/18,“ 29 09 2018. [На мрежи]. Available: http://www.parlament.gov.rs. [Последњи приступ 2018].

[43] Службени гласник РС, „Правилник о стандардима и поступку за акредитацију високошколских установа, Службени лгасник РС, број 88/17,“ 29 09 2017. [На мрежи]. Available: http://www.kapk.org/sr/акредитација/. [Последњи приступ 2018].

[44] KAPK, „Commission for accreditation and quality assurance, Guide for students,“ 2018. [На мрежи]. Available: http://www.kapk.org. [Последњи приступ 2018].

[45] R. Feldman и J. Sanger, The Text Mining Handbook, Cambridge University Press, 2006.

[46] S. Bird, E. Klein и E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, O'Reilly Media, Inc., 2009.

[47] L. V. Lita, A. Ittycheriah, S. Roukos и N. Kambhatla, „Truecasing,“ у ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Sapporo, Japan, 2003.

[48] srpskijezik.rs, „Morfologija - gramatika srpskog jezika,“ 2019. [На мрежи]. Available: http://www.srpskijezik.rs/gramatika/morfologija. [Последњи приступ 08 2019].

[49] A. Šilić, J.-H. Chauchat, B. Dalbelo Bašić и A. Morin, „N-grams and Morphological Normalization in Text Classiflcation: a Comparison on a Croatian-English Parallel Corpus,“ у EPIA'07 Proceedings of the aritficial intelligence 13th Portuguese conference on Progress in artificial intelligence, Portugal, 2007.

[50] M. F. Porter, „An algorithm for suffix stripping,“ Program, т. 14, бр. 3, pp. 130-137, 1980.

[51] V. Batanović, B. Nikolić и M. Milosavljević, „Reliable Baselines for Sentiment Analysis in Resource-Limited Languages: The Serbian Movie Review Dataset,“ у Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia, 2016.

[52] T. Jones, „Serbian Stemmer Analysis,“ November 2017. [На мрежи]. Available: https://www.mediawiki.org/wiki/User:TJones_(WMF)/Notes/Serbian_Stemmer_Analysis. [Последњи приступ October 2018].

[53] V. Batanovic, B. Furlan и B. Nikolic, „A Software System for Determining the Semantic Similarity of Short Texts in Serbian,“ у 19thTelecommunications Forum (TELFOR) Proceedings of Papers, Belgrade, 2011.

[54] C. D. Manning, P. Raghavan и H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[55] V. Prince и A. Labadié, „Text segmentation based on document understanding for information retrieval,“ у Natural Language Processing and Information Systems, Z. Kedad, N. Lammari, E. Métais, F. Meziane и Y. Rezgui, Уредници, Springer Berlin Heidelberg, 2007, pp. 295-304.

[56] Oxford University Press, „Oxford Dictionaries - Dictionary, Thesaurus, & Grammar,“ 2017. [На мрежи]. Available: https://en.oxforddictionaries.com/. [Последњи приступ 22 05 2017].

[57] F. Fouss, M. Saerens и M. Shimbo, Algorithms and Models for Network Data and Link Analysis, Cambridge University Press, 2016.

[58] Y. Kim, „Convolutional Neural Networks for Sentence Classification," у Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 2014.

[59] F. Chollet, Deep Learning with Python, Manning Publications Co., 2018.

[60] C. C. Aggarwal, Neural Networks and Deep Learning, Springer International Publishing, 2018, 2.

[61] M. Nikolić и A. Zečević, Mašinsko učenje, Beograd: Prirodno matematički fakultet, 2019.

[62] S. Hochreiter, Y. Bengio, P. Frasconi и J. Schmidhuber, „Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies," у A Field Guide to Dynamical Recurrent Neural Networks, S. C. Kremer и J. F. Kolen, Уредници, IEEE Press, 2001.

[63] S. Hochreiter и J. Schmidhuber, „Long Short-Term Memory," у Neural Computation, Massachusetts Institute of Technology, 1997, pp. 1735-1780.

[64] M. Basaldella, E. Antolli, G. Serra и C. Tasso, „Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction," у Italian Research Conference on Digital Libraries (IRCDL), Udine, Italy, 2018.

[65] M. Schuster и K. K. Paliwal, „Bidirectional Recurrent Neural Networks," IEEE Transactions on Signal Processing, т. 45, бр. 11, pp. 2673-2681, 1997.

[66] A. Graves и J. Schmidhuber, „Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures," Neural networks, the official journal of the International Neural Network Society, т. 18, бр. 5-6, pp. 602-610, 2005.

[67] N. Pappas и A. Popescu-Belis, „Multilingual Hierarchical Attention Networks for Document Classification," у Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP), Taipei, Taiwan, 2017.

[68] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola и E. Hovy, „Hierarchical Attention Networks for Document Classification," у Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, 2016.

[69] A. Bondy и U. Murty, „Graphs and Subgraphs; Directed Graphs,“ у Graph Theory with Applications, NORTH-HOLLAND, 1982., pp. 1-20, 171-178.

[70] D. Avis, A. Hertz и O. Marcotte, Graph Theory and Combinatorial Optimization, Springer US, 2005.

[71] N. Robertson и P. D. Seymour, „Graph Structure Theory,“ 1993.

[72] J. Luo и C. L. Magee, „Detecting evolving patterns of self-organizing networks by flow hierarchy measurement,“ Complexity, т. 16, бр. 6, pp. 53-61, 2011.

[73] L. Page, S. Brin, R. Motwani и T. Winograd, „The PageRank Citation Ranking: Bringing Order to the Web.,“ Stanford InfoLab, 1999.

[74] J. M. Kleinberg, „Authoritative Sources in a Hyperlinked Environment,“ Journal of the ACM (JACM), т. 46, бр. 5, pp. 604-632, 1999.

[75] D. Chakrabarti и K. Punera, „Event Summarization Using Tweets,“ у ICWSM Conference, 2011..

[76] B. O’Connor, R. Balasubramanyan, B. Routledge и N. Smith, „From tweets to polls: Linking text sentiment to public opinion time series,“ 2010..

[77] B. Pang и L. Lee, „A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,“ 2004..

11 Прилози

11.1 Прилог 1: Списак табела:

11.2 Прилог 2: Списак илустрација:

11.3 Прилог 3: Кодирање

У овом делу је дат програмски код у програмском језику *Python* (https://www.python.org/), који је писан за потребе истраживања која су спроведена, применом платформе *Anaconda* (https://www.anaconda.com/) и употребом окружења *Jupyter*.

11.3.1 Сегментација текстова

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:
# Сегментација текстова нових закона


# In[ ]:
# Учитавање неопходних библиотека
import pandas as pd
from pandas import DataFrame
import csv


# In[ ]:
# Учитавање текста за сегментацију
file=r'Закон о науци и истраживањима.txt'

openFile= open(file, 'r', encoding='utf-8-sig')
clanovi = []
# Следећи ред се мења у зависности да ли је ћирилица или латиница
clanovi= openFile.read().split('\nЧлан ')

for i in range(1,len(clanovi)):
    # Следећи ред се мења у зависности да ли је ћирилица или
латиница
    clanovi[i]="Члан "+clanovi[i]


# In[ ]:
# Примери приказа неког одређеног члана закона
#print(clanovi[51])
#print(clanovi[0])


# In[ ]:
# Прављење CSV датотеке, са заглављем за сегментирање (употребити
САМО ЈЕДНОМ)
filecsv=r'Zakon o nauci i istrazivanjima.csv'
Clanovi = {'PropisID': ['0'],
        'PropisDeoID': ['0'],
```

```
            'PropisDeo': ['0']
        }
df = DataFrame(Clanovi, columns= ['PropisID', 'PropisDeoID',
'PropisDeo'])

export_csv = df.to_csv (filecsv, encoding='utf-8-sig', index = None,
header=True)

print (df)


# In[ ]:
#Upisivanje podataka u CSV tabelu
for i in range(0,len(clanovi)):
    novi_zapis = ["1802-19", i, clanovi[i]]
    with open(filecsv, 'a', encoding='utf-8-sig', newline='') as
datoteka:
        writer = csv.writer(datoteka)
        writer.writerow(novi_zapis)


# In[ ]:
# Prikaz odredjenog clana zakona
CSVFile = pd.read_csv(filecsv, encoding='utf-8-sig')
#print(CSVFile.iat[10, 2] )
print(CSVFile.at[10, 'PropisDeo'] )


# In[ ]:
# Приказ заглавља из CSV датотеке
print(CSVFile.head())


# In[ ]:
# Приказ свих података из CSV датотеке
print(CSVFile)
```

11.3.2 Израчунавање фрекфенције језичких израза

```
#!/usr/bin/env python
# coding: utf-8

# In[]:
# Израчунавање фрекфенције језичких израза
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import operator


# In[]:
#Functions
#Thanks to https://github.com/SambitAcharya/Mini-
Projects/tree/master/Python/Phrase%20Frequency%20Counter

#Function to create the phraselist based on the user input
def createPhraseList(clean_list,number_of_phrases):
    '''
        @param Input: List without special symbols and the number of
phrases
        @return Output: List containing all the phrases of
@number_of_phrases length.
    '''
    phrase_list = []
    phrase = ''
    count = 0
    max_count = len(clean_list) - number_of_phrases + 1
    while count<max_count:
        for i in range(number_of_phrases):
            phrase = phrase + clean_list[count+i] + " "
        phrase_list.append(phrase)
        phrase = ''
        count+=1
    return(phrase_list)

#Function to create the dictionary which would store all the phrases
and their counts.
def createDictionary(phrase_list):
    '''
        @param Input: List containing all the phrases.
        @return Output: Dictionary containing all the phrases and
their counts.
    '''
    phrase_count = {}
    for phrase in phrase_list:
        if phrase in phrase_count:
            phrase_count[phrase] += 1
        else:
            phrase_count[phrase] = 1
    return phrase_count
```

```
#Function to clean up the list from special symbols so as to allow
only alphanumeric characters.
def cleanUpList(word_list):
    '''
        @param Input: List containing all the words.
        @return Output: @param List devoid of special symbols
    '''
    clean_word_list = []
    for word in word_list:
        word = ''.join(e for e in word if e.isalnum())
        if len(word)>0:
            clean_word_list.append(word)
    return(clean_word_list)

#Function to get input from the user, split them into words and pass
on for further processing.
def getWords(content):
    '''
        @param Input: String containing users input.
        @return Output: Symbol free list and the length of the list
made from @param string
    '''
    word_list = []
    words = content.lower().split()
    for each_word in words:
        word_list.append(each_word)
    clean_list = cleanUpList(word_list)
    length_of_content = len(clean_list)
    return clean_list,length_of_content

#Function to count all the phrases
def countPhrases(clean_list,number_of_phrases):
    '''
        @param Input: A clean list and the number of phrases given
by the user.
        @return Output: Dictionary containing the phrase count.
    '''
    phrase_list = createPhraseList(clean_list,number_of_phrases)
    phrase_count = createDictionary(phrase_list)
    return phrase_count


# In[]:
# Успостављање везе са базом података
import mysql.connector
conn = mysql.connector.connect(user='root', password='',
host='127.0.0.1', database='zakoni')
cursor = conn.cursor(buffered=True)

query = ("SELECT `PropisID`, `PropisDeoID`, `PropisDeo` FROM
`tblpropisitekst`")
cursor.execute(query)
```

```
for (PropisID, PropisDeoID, PropisDeo) in cursor:
    conn2 = mysql.connector.connect(user='root', password='',
host='127.0.0.1', database='zakoni')
    cursor2 = conn2.cursor(buffered=True)

    rows_affected=cursor.rowcount

    content = PropisDeo
    number_of_phrases = 5   #Broj reci u izrazu
    clean_list,length_of_content = getWords(content)
    condition = length_of_content - number_of_phrases
    phrase_count = countPhrases(clean_list,number_of_phrases)
    #Sorting the dictionary in increasing order of values.
    for key,value in sorted(phrase_count.items(),
key=operator.itemgetter(1)):
        sqlAdd = "insert into `tblpropisitekstphrases` VALUES('%s',
'%d', '%d', '%s', %d)" % (PropisID, PropisDeoID, number_of_phrases,
key, value)
        cursor2.execute(sqlAdd)
    conn2.commit()   #commit() method to save changes to the
database

    cursor2.close()
    conn2.close()

cursor.close()
conn.close()
print("Uradjeno.")
```

11.3.3 Прављење матрице за уградњу

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:
# Прављење матрице за уградњу


# In[ ]:
# Учитавање неопходних библиотека
from nltk.tokenize import sent_tokenize, word_tokenize
import warnings

warnings.filterwarnings(action = 'ignore')

import pandas as pd
import numpy as np
import re

import gensim
from gensim.models import Word2Vec

import multiprocessing


# In[ ]:
def clean_str(string):
    string = re.sub(r"\\", "", string)
    string = re.sub(r"\'", "", string)
    string = re.sub(r"\"", "", string)
    return string.strip().lower()


# In[ ]:
# Читање података
datoteka=r'tblPropisiTekst2.xlsx'
df = pd.read_excel(datoteka, encoding='utf-8-sig')
df = df.dropna()
df = df.reset_index(drop=True)
print('Shape of dataset ',df.shape)
print('Заглавље ', df.columns)
print('Број јединствених класа ',len(set(df['JILabelID'])))


# In[ ]:
data = []

for h in df['PropisDeo']:
    text=clean_str(h).lower()
    #print ("Član ", j)
    for i in sent_tokenize(text):
        temp = []
        #print ("Rečenica ", i)
```

```
        # tokenize the sentence into words
        for j in word_tokenize(i):
            temp.append(j.lower())
            #print ("Reč ", j)
        data.append(temp)


# In[ ]:
print(data[:3])


# In[ ]:
# Прављење вектора речи

# Димензија матрице за уградњу
emb_dim = 400

# window=5, посматра се 5 речи пре и после речи која се индексира
# min_count=5, у матрици за уградњу ће бити само речи које се у
текстовима појављују најмање 5 пута
# negative=15, у односу на посматрану реч, највише 15 других речи
које имају негативна својства у односу на њу
# iter=5, број итерација
# workers=multiprocessing.cpu_count(), број процеса је онолики
колико има процесора

w2v = Word2Vec(data, size=emb_dim, window=5, min_count=5,
negative=15, iter=5, workers=multiprocessing.cpu_count())


# In[ ]:
word_vectors = w2v.wv


# In[ ]:
# Приказ величине речника
vocabulary = w2v.wv.vocab
len(vocabulary)


# In[ ]:
# Претварање вектора речи у нумеричку матрицу, која је погодна за
TensorFlow и Keras моделе
embedding_matrix = np.zeros((len(w2v.wv.vocab), 400))
for i in range(len(w2v.wv.vocab)):
    embedding_vector = w2v.wv[w2v.wv.index2word[i]]
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector


# In[ ]:
# Приказ величине матрице за уградњу
embedding_matrix.shape
```

```
# In[ ]:
# Чување модела у TXT формату
np.savetxt('embedding_matrix_np2.txt', embedding_matrix, delimiter='
', encoding='utf-8-sig')


# In[ ]:
# Сумарни подаци о моделу
print(w2v)


# In[ ]:
# Употреба вектора речи за неку реч
#print(w2v['zakon'])


# In[ ]:
# Чување модела у бинарном формату
w2v.save('embedding_matrix_2.bin')


# In[ ]:
# Учитавање модела
#new_model = Word2Vec.load('embedding_matrix_2.bin')
#print(new_model)
```

11.3.4 Обучавање модела Рекурентне неуронске мреже и његова примена за предвиђање веза у необележеним текстовима закона

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:
# Рекурентна неуронска мрежа (ен. Recurrent Neural Network, RNN)
import numpy as np
import pandas as pd
import pickle
from collections import defaultdict
import re
import sys
import os
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils.np_utils import to_categorical
from keras.layers import Embedding
from keras.layers import Dense, Input, Flatten
from keras.layers import Conv1D, MaxPooling1D, Embedding, Dropout,
LSTM, GRU, Bidirectional
from keras.models import Model
from keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt
plt.switch_backend('agg')
from keras import backend as K
from keras.engine.topology import Layer, InputSpec
from keras import initializers
get_ipython().run_line_magic('matplotlib', 'inline')


# In[ ]:
def clean_str(string):
    string = re.sub(r"\\", "", string)
    string = re.sub(r"\'", "", string)
    string = re.sub(r"\"", "", string)
    return string.strip().lower()


# In[ ]:
MAX_SEQUENCE_LENGTH = 400
MAX_NB_WORDS = 1000
EMBEDDING_DIM = 400
VALIDATION_SPLIT = 0.2


# In[ ]:
# Читање података
datoteka=r'tblPropisiTekst2.xlsx'
df = pd.read_excel(datoteka, encoding='utf-8-sig')
df = df.dropna()
df = df.reset_index(drop=True)
print('Облик скупа података ',df.shape)
```

```
print('Заглавље ', df.columns)
print('Број јединствених класа',len(set(df['JILabelID'])))


# In[ ]:
macronum=sorted(set(df['JILabelID']))
macro_to_id = dict((note, number) for number, note in
enumerate(macronum))

def fun(i):
    return macro_to_id[i]

df['JILabelID']=df['JILabelID'].apply(fun)


# In[ ]:
texts = []
labels = []
for j in df['PropisDeo']:
    text=clean_str(j).lower()
    texts.append(text)

for i in df['JILabelID']:
    labels.append(i)


# In[ ]:
tokenizer = Tokenizer(num_words=MAX_NB_WORDS)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

word_index = tokenizer.word_index
print('Број јединствених токена: ',len(word_index))


# In[ ]:
# Узимање секвенци у једнаким интервалима
data = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)


labels = to_categorical(np.asarray(labels))
print('Облик тензора са подацима:', data.shape)
print('Облик тензора са ознакама:', labels.shape)

indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
labels = labels[indices]
nb_validation_samples = int(VALIDATION_SPLIT * data.shape[0])

x_train = data[:-nb_validation_samples]
y_train = labels[:-nb_validation_samples]
x_val = data[-nb_validation_samples:]
y_val = labels[-nb_validation_samples:]
```

```
# In[ ]:
embeddings_index = {}
f = open('embedding_matrix_np2.txt',encoding='utf-8-sig')
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()

print('Укупно %s вектора за речи у матрици за уградњу.' %
len(embeddings_index))


# In[ ]:
embedding_matrix = np.random.random((len(word_index) + 1,
EMBEDDING_DIM))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector


# In[ ]:
embedding_layer = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix],
                            input_length=MAX_SEQUENCE_LENGTH,
                            #trainable=True)
                            trainable=False)


# In[ ]:
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedded_sequences = embedding_layer(sequence_input)
l_lstm = Bidirectional(LSTM(50))(embedded_sequences)
preds = Dense(len(macronum), activation='softmax')(l_lstm)
model = Model(sequence_input, preds)
#model.compile(loss='categorical_crossentropy',
              #optimizer='rmsprop',
              #metrics=['acc'])
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['acc'])

print("Bidirectional LSTM")
model.summary()


# In[ ]:
```

```
cp=ModelCheckpoint('model_rnn-
3.hdf5',monitor='val_acc',verbose=1,save_best_only=True)
history=model.fit(x_train, y_train, validation_data=(x_val,
y_val),epochs=5, batch_size=2,callbacks=[cp])


# In[ ]:
fig1 = plt.figure()
plt.plot(history.history['loss'],'r',linewidth=3.0)
plt.plot(history.history['val_loss'],'b',linewidth=3.0)
plt.legend(['Gubitak, obuka', 'Gubitak, validacija'],fontsize=18)
plt.xlabel('Epohe ',fontsize=16)
plt.ylabel('Gubitak',fontsize=16)
plt.title('Kriva gubitaka :RNN',fontsize=16)
plt.ylim((0,1))
plt.xticks(np.arange(0, 5, 1))
fig1.savefig('rnn_gubitak-2.png')
plt.show()


# In[ ]:
fig2=plt.figure()
plt.plot(history.history['acc'],'r',linewidth=3.0)
plt.plot(history.history['val_acc'],'b',linewidth=3.0)
plt.legend(['Tacnost, obuka', 'Tacnost, validacija'],fontsize=18)
plt.xlabel('Epohe ',fontsize=16)
plt.ylabel('Tacnost',fontsize=16)
plt.title('Kriva tacnosti : RNN',fontsize=16)
plt.ylim((0.5,1))
plt.xticks(np.arange(0, 5, 1))
fig2.savefig('rnn_tacnost-2.png')
plt.show()


# In[ ]:
# Чување модела
model.save('model_rnn-model-save-2.hdf5')



# # Предвиђање израза за повезивање у текстовима нових закона

# In[ ]:
# Учитавање података за предвиђање (текстови треба да буду
латинични)
n_datoteka=r'Zakon o nauci i istrazivanjima.csv'
n_df = pd.read_csv(n_datoteka, encoding='utf-8-sig')
print('Облик скупа података ',n_df.shape)
print('Заглавље ', n_df.columns)
#print('Број јединствених класа',len(set(df['JILabelID'])))


# In[ ]:
# Текстови нових закона
```

```
novi_tekstovi = []
for j in n_df['PropisDeo']:
    novi_tekst=clean_str(j).lower()
    novi_tekstovi.append(novi_tekst)


# In[ ]:
# Токенизација нових текстова
n_tokenizer = Tokenizer(num_words=MAX_NB_WORDS)
n_tokenizer.fit_on_texts(novi_tekstovi)
n_sequences = tokenizer.texts_to_sequences(novi_tekstovi)


# In[ ]:
# Узимање секвенци у једнаким интервалима
n_data = pad_sequences(n_sequences, maxlen=MAX_SEQUENCE_LENGTH)

#labels = to_categorical(np.asarray(labels))
print('Облик тензора са подацима:', n_data.shape)
#print('Облик тензора са ознакама:', labels.shape)

n_indices = np.arange(n_data.shape[0])
n_data = n_data[n_indices]


# In[ ]:
# Учитавање модела
from keras.models import load_model
load_model = load_model('model_rnn-model-save-2.hdf5')


# In[ ]:
# Предвиђање
preds = load_model.predict(n_data)


# In[ ]:
# Приказ предвиђања за све класе
print(preds)


# In[ ]:
# Извоз резултата предвиђања
np.savetxt("rnn_matrica.csv", preds, delimiter=",")


# In[ ]:
# Предвиђање једне од класа
predicted = np.argmax(preds, axis=1)


# In[ ]:
# Приказ предвиђања класе
predicted
```

```
# In[ ]:
# Извоз резултата предвиђања
np.savetxt("rnn_predicted.csv", predicted, delimiter=",")
```

11.3.5 Обучавање модела Конволуционе неуронске мреже и његова примена за предвиђање веза у необележеним текстовима закона

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:
# Конволуциона неуронска мрежа (ен. Convolutional Neural Network, 
CNN)
import numpy as np
import pandas as pd
import pickle
from collections import defaultdict
import re
import sys
import os
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils.np_utils import to_categorical
from keras.layers import Embedding
from keras.layers import Dense, Input, Flatten
from keras.layers import Conv1D, MaxPooling1D, Embedding, Dropout, 
AveragePooling1D
from keras.models import Model
from keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt
plt.switch_backend('agg')
get_ipython().run_line_magic('matplotlib', 'inline')


# In[ ]:
def clean_str(string):
    string = re.sub(r"\\", "", string)
    string = re.sub(r"\'", "", string)
    string = re.sub(r"\"", "", string)
    return string.strip().lower()


# In[ ]:
MAX_SEQUENCE_LENGTH = 400
MAX_NB_WORDS = 1000
EMBEDDING_DIM = 400
VALIDATION_SPLIT = 0.2


# In[ ]:
# Читање података
datoteka=r'tblPropisiTekst2.xlsx'
df = pd.read_excel(datoteka, encoding='utf-8-sig')
df = df.dropna()
df = df.reset_index(drop=True)
print('Облик скупа података ',df.shape)
print('Заглавље ', df.columns)
print('Број јединствених класа',len(set(df['JILabelID'])))
```

```
# In[ ]:
macronum=sorted(set(df['JILabelID']))
macro_to_id = dict((note, number) for number, note in
enumerate(macronum))

def fun(i):
    return macro_to_id[i]

df['JILabelID']=df['JILabelID'].apply(fun)


# In[ ]:
texts = []
labels = []
for j in df['PropisDeo']:
    text=clean_str(j).lower()
    texts.append(text)

for i in df['JILabelID']:
    labels.append(i)


# In[ ]:
tokenizer = Tokenizer(num_words=MAX_NB_WORDS)
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

word_index = tokenizer.word_index
print('Број јединствених токена: ',len(word_index))


# In[ ]:
# Узимање секвенци у једнаким интервалима
data = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)

labels = to_categorical(np.asarray(labels))
print('Облик тензора са подацима:', data.shape)
print('Облик тензора са ознакама:', labels.shape)

indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
labels = labels[indices]
nb_validation_samples = int(VALIDATION_SPLIT * data.shape[0])

x_train = data[:-nb_validation_samples]
y_train = labels[:-nb_validation_samples]
x_val = data[-nb_validation_samples:]
y_val = labels[-nb_validation_samples:]


# In[ ]:
```

```
embeddings_index = {}
f = open('embedding_matrix_np2.txt',encoding='utf-8-sig')
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()

print('Укупно %s вектора за речи у матрици за уградњу.' %
len(embeddings_index))


# In[ ]:
embedding_matrix = np.random.random((len(word_index) + 1,
EMBEDDING_DIM))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector


# In[ ]:
embedding_layer = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix],
                            input_length=MAX_SEQUENCE_LENGTH,
                            #trainable=True)
                            trainable=False)


# In[ ]:
from keras.optimizers import RMSprop

sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedded_sequences = embedding_layer(sequence_input)
l_conv1= Conv1D(128, 5, activation='relu')(embedded_sequences)
l_pool1 = MaxPooling1D(5)(l_conv1)
l_conv2 = Conv1D(128, 5, activation='relu')(l_pool1)
l_pool2 = MaxPooling1D(5)(l_conv2)
l_conv3 = Conv1D(128, 5, activation='relu')(l_pool2)
l_pool3 = MaxPooling1D(11)(l_conv3)  # global max pooling
l_flat = Flatten()(l_pool3)
l_dense = Dense(128, activation='relu')(l_flat)
preds = Dense(len(macronum), activation='softmax')(l_dense)

model = Model(sequence_input, preds)
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['acc'])

print("Simplified convolutional neural network")
model.summary()
```

```
# In[ ]:
cp=ModelCheckpoint('model_cnn-
2.hdf5',monitor='val_acc',verbose=1,save_best_only=True)
history=model.fit(x_train, y_train, validation_data=(x_val,
y_val),epochs=5, batch_size=2,callbacks=[cp])


# In[ ]:
fig1 = plt.figure()
plt.plot(history.history['loss'],'r',linewidth=3.0)
plt.plot(history.history['val_loss'],'b',linewidth=3.0)
plt.legend(['Gubitak, obuka', 'Gubitak, validacija'],fontsize=18)
plt.xlabel('Epohe ',fontsize=16)
plt.ylabel('Gubitak',fontsize=16)
plt.title('Kriva gubitaka: CNN',fontsize=16)
plt.ylim((0,1))
plt.xticks(np.arange(0, 5, 1))
fig1.savefig('cnn_gubitak-2.png')
plt.show()


# In[ ]:
fig2=plt.figure()
plt.plot(history.history['acc'],'r',linewidth=3.0)
plt.plot(history.history['val_acc'],'b',linewidth=3.0)
plt.legend(['Tacnost, obuka', 'Tacnost, validacija'],fontsize=18)
plt.xlabel('Epohe ',fontsize=16)
plt.ylabel('Tacnost',fontsize=16)
plt.title('Kriva tacnosti: CNN',fontsize=16)
plt.ylim((0.5,1))
plt.xticks(np.arange(0, 5, 1))
fig2.savefig('cnn_tacnost-2.png')
plt.show()


# In[ ]:
# Чување модела
model.save('model_cnn-model-save-2.hdf5')



# # Предвиђање израза за повезивање у текстовима нових закона

# In[ ]:
# Учитавање података за предвиђање (текстови треба да буду
латинични)
n_datoteka=r'Zakon o nauci i istrazivanjima.csv'
n_df = pd.read_csv(n_datoteka, encoding='utf-8-sig')
print('Облик скупа података ',n_df.shape)
print('Заглавље ', n_df.columns)
#print('Број јединствених класа',len(set(df['JILabelID'])))
```

```
# In[ ]:
# Текстови нових закона
novi_tekstovi = []
for j in n_df['PropisDeo']:
    novi_tekst=clean_str(j).lower()
    novi_tekstovi.append(novi_tekst)


# In[ ]:
# Токенизација нових текстова
n_tokenizer = Tokenizer(num_words=MAX_NB_WORDS)
n_tokenizer.fit_on_texts(novi_tekstovi)
n_sequences = tokenizer.texts_to_sequences(novi_tekstovi)


# In[ ]:
# Узимање секвенци у једнаким интервалима
n_data = pad_sequences(n_sequences, maxlen=MAX_SEQUENCE_LENGTH)

#labels = to_categorical(np.asarray(labels))
print('Облик тензора са подацима:', n_data.shape)
#print('Облик тензора са ознакама:', labels.shape)

n_indices = np.arange(n_data.shape[0])
n_data = n_data[n_indices]


# In[ ]:
# Учитавање модела
from keras.models import load_model
load_model = load_model('model_cnn-model-save-2.hdf5')


# In[ ]:
# Предвиђање
preds = load_model.predict(n_data)


# In[ ]:
# Приказ предвиђања за све класе
print(preds)


# In[ ]:
# Предвиђање једне од класа
predicted = np.argmax(preds, axis=1)


# In[ ]:
# Приказ предвиђања класе
predicted
```

```
# In[ ]:
# Извоз резултата предвиђања
np.savetxt("cnn_predicted.csv", predicted, delimiter=",")
```

11.3.6 Обучавање модела Хијерархијске неуронске мреже са уграђеним моделом пажње и његова примена за предвиђање веза у необележеним текстовима закона

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:
# Хијерархијска мрежа са уграђеним механизмом пажње (ен.
Hierarchical Attention Network, HAN)


# In[ ]:
import numpy as np
import pandas as pd
import pickle
from collections import defaultdict
import re
import sys
import os
import nltk
from nltk import tokenize
from keras.preprocessing.text import Tokenizer,text_to_word_sequence
from keras.preprocessing.sequence import pad_sequences
from keras.utils.np_utils import to_categorical
from keras.layers import Embedding
from keras.layers import Dense, Input, Flatten
from keras.layers import Conv1D, MaxPooling1D, Embedding, Dropout,
LSTM, GRU, Bidirectional, TimeDistributed
from keras.models import Model
from keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt
plt.switch_backend('agg')
from keras import backend as K
from keras.engine.topology import Layer, InputSpec
from keras import initializers
get_ipython().run_line_magic('matplotlib', 'inline')


# In[ ]:
def clean_str(string):
    string = re.sub(r"\\", "", string)
    string = re.sub(r"\'", "", string)
    string = re.sub(r"\"", "", string)
    return string.strip().lower()


# In[ ]:
MAX_SENT_LENGTH = 100
MAX_SENTS = 10
MAX_NB_WORDS = 1000
EMBEDDING_DIM = 400
VALIDATION_SPLIT = 0.2
```

```
# In[ ]:
# Читање података
datoteka=r'tblPropisiTekst2.xlsx'
df = pd.read_excel(datoteka, encoding='utf-8-sig')
df = df.dropna()
df = df.reset_index(drop=True)
print('Облик скупа података ',df.shape)
print('Заглавље ', df.columns)
print('Број јединствених класа',len(set(df['JILabelID'])))


# In[ ]:
macronum=sorted(set(df['JILabelID']))
macro_to_id = dict((note, number) for number, note in
enumerate(macronum))

def fun(i):
    return macro_to_id[i]

df['JILabelID']=df['JILabelID'].apply(fun)


# In[ ]:
reviews = []
labels = []
texts = []

for j in df['PropisDeo']:
    text=clean_str(j).lower()
    texts.append(text)
    sentences = tokenize.sent_tokenize(text)
    reviews.append(sentences)

for i in df['JILabelID']:
    labels.append(i)


# In[ ]:
tokenizer = Tokenizer(num_words=MAX_NB_WORDS)
tokenizer.fit_on_texts(texts)

data = np.zeros((len(texts), MAX_SENTS, MAX_SENT_LENGTH),
dtype='int32')

for i, sentences in enumerate(reviews):
    for j, sent in enumerate(sentences):
        if j< MAX_SENTS:
            wordTokens = text_to_word_sequence(sent)
            k=0
            for _, word in enumerate(wordTokens):
                #print(word)
                #print(tokenizer.word_index[word])
```

```
                if k<MAX_SENT_LENGTH and
tokenizer.word_index[word]<MAX_NB_WORDS:
                    data[i,j,k] = tokenizer.word_index[word]
                    k=k+1


# In[ ]:
word_index = tokenizer.word_index
print('Број јединствених токена: ',len(word_index))


# In[ ]:
labels = to_categorical(np.asarray(labels))
print('Облик тензора са подацима:', data.shape)
print('Облик тензора са ознакама:', labels.shape)

indices = np.arange(data.shape[0])
np.random.shuffle(indices)
data = data[indices]
labels = labels[indices]
nb_validation_samples = int(VALIDATION_SPLIT * data.shape[0])


# In[ ]:
x_train = data[:-nb_validation_samples]
y_train = labels[:-nb_validation_samples]
x_val = data[-nb_validation_samples:]
y_val = labels[-nb_validation_samples:]


# In[ ]:
embeddings_index = {}
f = open('embedding_matrix_np2.txt',encoding='utf-8-sig')
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()

print('Укупно %s вектора за речи у матрици за уградњу.' %
len(embeddings_index))


# In[ ]:
embedding_matrix = np.random.random((len(word_index) + 1,
EMBEDDING_DIM))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector
```

```
# In[ ]:
embedding_layer = Embedding(len(word_index) + 1,
                            EMBEDDING_DIM,
                            weights=[embedding_matrix],
                            input_length=MAX_SENT_LENGTH,
                            #trainable=True)
                            trainable=False)


# In[ ]:
sentence_input = Input(shape=(MAX_SENT_LENGTH,), dtype='int32')
embedded_sequences = embedding_layer(sentence_input)
l_lstm = Bidirectional(LSTM(50))(embedded_sequences)
sentEncoder = Model(sentence_input, l_lstm)

review_input = Input(shape=(MAX_SENTS,MAX_SENT_LENGTH),
dtype='int32')
review_encoder = TimeDistributed(sentEncoder)(review_input)
l_lstm_sent = Bidirectional(LSTM(20))(review_encoder)
preds = Dense(len(macronum), activation='softmax')(l_lstm_sent)
model = Model(review_input, preds)

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['acc'])

print("Hierachical LSTM")
model.summary()


# In[ ]:
cp=ModelCheckpoint('model_han-
2.hdf5',monitor='val_acc',verbose=1,save_best_only=True)
history=model.fit(x_train, y_train, validation_data=(x_val, y_val),
          epochs=5, batch_size=2,callbacks=[cp])


# In[ ]:
fig1 = plt.figure()
plt.plot(history.history['loss'],'r',linewidth=3.0)
plt.plot(history.history['val_loss'],'b',linewidth=3.0)
plt.legend(['Gubitak, obuka', 'Gubitak, validacija'],fontsize=18)
plt.xlabel('Epohe',fontsize=16)
plt.ylabel('Gubitak',fontsize=16)
plt.title('Kriva gubitaka: HAN',fontsize=16)
plt.ylim((0,1))
plt.xticks(np.arange(0, 5, 1))
fig1.savefig('han_gubitak-2.png')
plt.show()


# In[ ]:
fig2=plt.figure()
plt.plot(history.history['acc'],'r',linewidth=3.0)
```

```
plt.plot(history.history['val_acc'],'b',linewidth=3.0)
plt.legend(['Tacnost, obuka', 'Tacnost, validacija'],fontsize=18)
plt.xlabel('Epohe',fontsize=16)
plt.ylabel('Tacnost',fontsize=16)
plt.title('Kriva tacnosti: HAN',fontsize=16)
plt.ylim((0.5,1))
plt.xticks(np.arange(0, 5, 1))
fig2.savefig('han_tacnost-2.png')
plt.show()


# In[ ]:
# Чување модела
model.save('model_han-model-save-2.hdf5')



# # Предвиђање израза за повезивање у текстовима нових закона

# In[ ]:
# Учитавање података за предвиђање (текстови треба да буду
латинични)
n_datoteka=r'Zakon o nauci i istrazivanjima.csv'
n_df = pd.read_csv(n_datoteka, encoding='utf-8-sig')
print('Облик скупа података ',n_df.shape)
print('Заглавље ', n_df.columns)
#print('Број јединствених класа',len(set(df['JILabelID'])))


# In[ ]:
# Текстови нових закона
n_reviews = []
n_texts = []

for j in n_df['PropisDeo']:
    n_text=clean_str(j).lower()
    n_texts.append(n_text)
    n_sentences = tokenize.sent_tokenize(n_text)
    n_reviews.append(n_sentences)


# In[ ]:
n_reviews


# In[ ]:
# Токенизација нових текстова
n_tokenizer = Tokenizer(num_words=MAX_NB_WORDS)
n_tokenizer.fit_on_texts(n_texts)

n_data = np.zeros((len(n_texts), MAX_SENTS, MAX_SENT_LENGTH),
dtype='int32')

for i, n_sentences in enumerate(n_reviews):
```

```
    for j, sent in enumerate(n_sentences):
        if j< MAX_SENTS:
            n_wordTokens = text_to_word_sequence(sent)
            k=0
            for _, word in enumerate(n_wordTokens):
                #print(word)
                #print(tokenizer.word_index[word])
                if k<MAX_SENT_LENGTH and
n_tokenizer.word_index[word]<MAX_NB_WORDS:
                    n_data[i,j,k] = n_tokenizer.word_index[word]
                    k=k+1


# In[ ]:
# Узимање секвенци у једнаким интервалима
print('Облик тензора са подацима:', n_data.shape)
#print('Облик тензора са ознакама:', labels.shape)

n_indices = np.arange(n_data.shape[0])
n_data = n_data[n_indices]
n_nb_validation_samples = int(0 * n_data.shape[0])


# In[ ]:
n_data


# In[ ]:
n_x_train = data[:-n_nb_validation_samples]
#y_train = labels[:-nb_validation_samples]
#x_val = data[-nb_validation_samples:]
#y_val = labels[-nb_validation_samples:]


# In[ ]:
# Учитавање модела
from keras.models import load_model
load_model = load_model('model_han-model-save-2.hdf5')
#print(new_model)


# In[ ]:
# Предвиђање
preds = load_model.predict(n_data)


# In[ ]:
# Приказ предвиђања за све класе
print(preds)


# In[ ]:
# Предвиђање једне од класа
predicted = np.argmax(preds, axis=1)
```

```
# In[ ]:
# Приказ предвиђања класе
predicted


# In[ ]:
# Извоз резултата предвиђања
np.savetxt("han_predicted.csv", predicted, delimiter=",")
```

12 Биографија аутора

Ђорђе Петровић, дипл. инж. из Ваљева, рођен је 1970. године, од оца Крсте и мајке Мире. Ожењен је супругом Бојаном и отац је троје деце, Михаила, Огњена и Неде.

Има преко 23 године радног искуства у производњи, у државној управи и у настави у високом образовању. Од 1996. године је запослен, најпре у предузећу „Минелопрема“ у Пачеву, а затим у предузећу „Босс компани“ у Ваљеву. Од 2000. године се професионално бави информатиком и информационим системима у Градској управи у Ваљеву, али и самостално, на пројектима по уговору. Од 2003. године је ангажован у настави у високом образовању у Високој пословној школи струковних студија у Ваљеву (данас Академија струковних студија западне Србије), где је учествовао или учествује у реализацији наставе из више од 10-ак предмета. 2019. године је ангажован у Високој школи за информационе технологије у Београду у реализацији наставе из предмета Data mining.

Аутор је или коаутор на 20-ак радова, који су објављени у домаћим и међународним часописима и конференцијама. Аутор је или коаутор 5 књига из области информатике и информационих система.

13 Изјаве аутора

13.1 Изјава 1.

ИЗЈАВА О АУТОРСТВУ

Изјављујем да је докторска дисертација, под насловом:

Анализа структуре колекције правних докумената на основу њихове повезаности преко одређених језичких израза

која је одбрањена на Електронском факултету Универзитета у Нишу:

- резултат сопственог истраживачког рада;
- да ову дисертацију, ни у целини, нити у деловима, нисам пријављивао/ла на другим факултетима, нити универзитетима;
- да нисам повредио/ла ауторска права, нити злоупотребио/ла интелектуалну својину других лица.

Дозвољавам да се објаве моји лични подаци, који су у вези са ауторством и добијањем академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада, и то у каталогу Библиотеке, Дигиталном репозиторијуму Универзитета у Нишу, као и у публикацијама Универзитета у Нишу.

У Нишу, 07.02.2020.

Потпис аутора дисертације:

Горје К. Петровић

(Име, средње слово и презиме)

13.2 Изјава 2.

ИЗЈАВА О ИСТОВЕТНОСТИ ЕЛЕКТРОНСКОГ И ШТАМПАНОГ ОБЛИКА ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Наслов дисертације:

Анализа структуре колекције правних докумената на основу њихове повезаности преко одређених језичких израза

Изјављујем да је електронски облик моје докторске дисертације, коју сам предао/ла за уношење у **Дигитални репозиторијум Универзитета у Нишу,** истоветан штампаном облику.

У Нишу, 07.02.2020.

Потпис аутора дисертације:

Горан К. Петровић

(Име, средње слово и презиме)

13.3 Изјава 3:

ИЗЈАВА О КОРИШЋЕЊУ

Овлашћујем Универзитетску библиотеку „Никола Тесла“ да у Дигитални репозиторијум Универзитета у Нишу унесе моју докторску дисертацију, под насловом:

Анализа структуре колекције правних докумената на основу њихове повезаности преко одређених језичких израза

Дисертацију са свим прилозима предао/ла сам у електронском облику, погодном за трајно архивирање.

Моју докторску дисертацију, унету у Дигитални репозиторијум Универзитета у Нишу, могу користити 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 прераде (CC BY-NC-ND)**
4. Ауторство – некомерцијално – делити под истим условима **(CC BY-NC-SA)**
5. Ауторство – без прераде **(CC BY-ND)**
6. Ауторство – делити под истим условима **(CC BY-SA)**[3]

У Нишу, 07.02.2020.

Потпис аутора дисертације:

Горан К. Петровић

(Име, средње слово и презиме)