Multi-label klasifikacija klauzula iz pravnih ugovora

Katarina Aleksić
Softversko inženjerstvo i informacione tehnologije
Fakultet tehničkih nauka, Univerzitet u Novom Sadu
Novi Sad, Srbija
katarina.aleksic97@gmail.com

Nikolina Batinić
Softversko inženjerstvo i informacione tehnologije
Fakultet tehničkih nauka, Univerzitet u Novom Sadu
Novi Sad, Srbija
nina.batinic@yahoo.com

***Sažetak*— U poslednjim godinama, sve je veći fokus na primeni veštačke inteligencije, naročito NLP (eng. *Natural Language Processing*) tehnika, u rešavanju zadataka iz pravnog domena. Kroz ovaj rad su opisana rešenja za problem klasifikacije klauzula iz pravnih ugovora u jednu ili više kategorija. Zadatak je da se iz jednog pravnog dokumenta izdvoje sve klauzule koje pripadaju jednoj ili više kategorija, te da se naznači koje su te kategorije. To bi u velikoj meri olakšalo i smanjilo napore za pregled dugih i kompleksnih pravnih ugovora od strane ljudi. Pravnicima bi u tom slučaju bilo mnogo lakše da pronađu najvažnije stavke iz ugovora. Kroz rad je predloženo nekoliko algoritama mašinskog učenja i tehnika za *embedding* u cilju klasifikacije teksta. Skup podataka koji se koristi u radu je ručno labeliran od pravnih eksperata iz *The Atticus Project*-a[1] i sadrži preko 4.000 labeliranih klauzula.**

Ključne reči—NLP; Lineaer SVC; XGBoost; KNN; Decision Tree; neuronske mreže; LSTM; klauzule; EDGAR;

I. UVOD

Primena veštačke inteligencije u pravnom domenu dobija sve veći značaj u poslednjih nekoliko godina. Iako su se mnogi pravnici ranije opirali promenama, nakon velikog napretka u NLP-u (*Natural langauge processing*), sve je veća zainteresovanost, kako pravnika, tako i inženjera koji se bave veštačkom inteligencijom[1].

NLP može da omogući brojne benefite pravnoj informatici i sve više se primenjuje u[2]:

1. Pravnom istraživanju – pronalaženje informacija relevantnih za pravnu odluku,
2. Elektronskom otkrivanju - utvrđivanje relevantnosti dokumenata za zahtev za informacijama,
3. Pregled ugovora(eng. *contract-review*) - provera da li je ugovor potpun i izbegavanje rizika,
4. Automatizacija dokumenata - generiranje rutinskih pravnih dokumenata,
5. Pravnom savetu: korišćenje dijaloga za pitanja i odgovore za pružanje prilagođenih saveta

Ovaj rad bavi se pregledom ugovora.

Automatizovani sistemi za pregled ugovora mogu se koristiti za pregled dokumenata koji su relativno standardizovani i predvidljivi u pogledu vrsta sadržaja koje sadrže. Proces uključuje razlaganje ugovora na njegove pojedinačne odredbe ili klauzule, a zatim procenjivanje svake od njih, bilo da bi se izdvojile ključne informacije ili da bi se uporedile sa nekim standardom[2].

Prilikom pregleda ugovora, pravnici moraju često da temeljno prolaze kroz čitave ugovore koji mogu biti dugački od nekoliko strana, pa do čak nekoliko stotina strana, samo da bi ručno pronašli par značajnih klauzula koje se nalaze u tom ugovoru, šta je tim klauzulama određeno, da li neke klauzule nedostaju u ugovoru, i slično. Prema podacima Razvojne banke Saveta Evrope[2](eng. *Council of Europe Development Bank*, CEB), mnoge advokatske firme potroše oko 50% vremena na pregledanje ugovora[3].

Pregled ugovora, osim što zahteva veliku količinu vremena, kao i neefikasno korišćenje veština pravnih stučnjaka, često je i veoma skup proces za klijenta. Umesto da se pravnici koncentišu na značajnije probleme, oni moraju da troše svoje dragoceno vreme na dosadne zadatke koji mogu biti automatizovani.

Iako već postoje sistemi bazirani na pravilima koji mogu donekle da olakšaju ovaj posao, sa napretkom NLP-a stvaraju se mnogo veće mogućnosti za poboljšanje ove oblasti, jer je u osnovi svih ovih dokumenata jezik, tekst[4].

Glavni problemi u automatizaciji pregledanja ugovora i izvdajanja klauzula jeste to što su izrazi koji se koriste u ugovorima vezani za specifičan pravni domen. Zbog toga, neophodni su podaci koji su vezani za ovaj domen, ali danas, iako postoji neki napredak, i dalje ima jako malo podataka koji su javno dostupni za obučavanje modela, i to ne samo u oblasti pregleda ugovora, već generalno u pravnoj informatici. Glavni razlozi za to su pre svega što su često u pitanu osetljive informacije koje ni ne smeju da budu javno dostupne, ali i što anotiranje tih podataka zahteva ekspertsko zananje pravnih stručnjaka, pa samim tim iziskuje dosta vremena i novca.

[1] https://www.atticusprojectai.org/

[2] https://coebank.org/en/

U 2. poglavlju opisani su radovi koji su povezani i značajni za ovaj rad. U sledećem poglavlju opisan je skup podataka korišten u radu, a potom metodologije korištene za pretprocesiranje podataka, *embedding* i klasifikaciju. U 5. poglavlju prikazani su dobijeni rezultati, i u poslednjem poglavlju je zaključak rada.

II. Povezani radovi

Kada uzmemo u obzir da je NLP svoj nagli napredak doživeo u poslednjih par godina, kao i da još uvek ima veoma malo javno dostupnih podataka vezanih za pravnu informatiku, nije ni čudo što ne postoji preveliki broj radova koji se bave ovom temom, kao i to da su svi koji postoje skorijeg datuma. U ovom poglavlju će biti navedeni neki od njih.

A. LEGAL-BERT

BERT[3]*(Bidirectional Encoder Representations from Transformers)* predstavlja *state-of-the-art* u nekoliko NLP zadataka sa generčkim podacima. Međutim, pošto je za treniranje BERT-a korišćen generički korpus kao što su podaci sa Vikipedije[4](*Wikipedia*), dečije knjige i slično, BERT često daje slabije rezultate kada su u pitanju specifični domeni. Zbog toga postoje mnoge verzije BERT-a koje su specijalizovane za određene domene, poput biomedicinskog, tako što je postojeći BERT dotreniran ili se pretrenira od nule, i takvi domenski specijalizovani modeli daju značajno bolje rezultate.

U ovom radu[4] predstavljen je prvi, i za sada jedini model BERT-a koji je specijalizovan za prvni domen tako što je pretreniran od nule. Ovaj model daje značajno bolje rezultate u odnosu na standardni BERT u pogledu različitih složenijih problema. Što se tiče multi-label klasifikacije, poboljšanje je čak 2.5%. Obučavan je na čak 12 GB različitog pravnog teksta na engleskom jeziku iz nekoliko oblasti (npr. zakonodavstvo, sudski sporovi, ugovori).

Korpus za treniranje LEGAL-BERT-a sastoji se iz:

- 116.062 pravnih dokumenta Evropske unije, javno dostupnih na EURLEKS[5]-u, repozitorijumu koji je pod upravom Ureda za publikacije Evropske Unije[6](eng. *EU Publication Office*).
- 61.826 dokumenata britanskog zakonodavstva, javno dostupnih sa portala britanskog zakonodavstva[7].
- 19.867 predmeta Evropskog suda pravde (eng. *European Court of Justice*, ECJ), takođe dostupnih na EURLEKS-u.
- 12.554 predmeta sa HUDOC[8]-a (*Human Rights Documentation)*, repozitorijuma Evropskog suda za ljudska prava[9] (eng. *European Court of Human Rights*, ECHR).
- 164.141 predmet različitih sudova širom SAD-a, dostupnih na portalu *Project Law Access Project*[10].
- 76.366 američkih ugovora sa EDGAR-a (*Electronic Data Gathering, Analysis, and Retrieval*), baze podataka Američke komisije za hartije od vrednosti[11] (eng. *US Securities and Exchange Commission*, SECOM)

U radu su predstavljene različite varijacije LEGAL-BERT-a, koje su i javno dostupne za korišćenje, i to su:

- CONTRACTS-BERT-BASE - pretreniran na američkim ugovorima, pogodan za prepoznavanje imenovanih entiteta(eng. *Named entity recognition*, NER)
- EURLEX-BERT-BASE – pretreniran na dokumentima sa EURLEKS repozitorijuma, pogodan za *multi-label* klasifikaciju,
- ECHR-BERT-BASE – pretreniran na slučajevima Evropskog suda pravde, pogodan za binarnu i *multi-label* klasifikaciju,
- LEGAL-BERT-BASE – pretreniran na celokupnom korpusu,
- LEGAL-BERT-SMALL – pretreniran na celokupnom korpusu, čak 33% manji od BERT-BASE-a, sa približnim peromansama kao on, ali čak 4 puta brži.

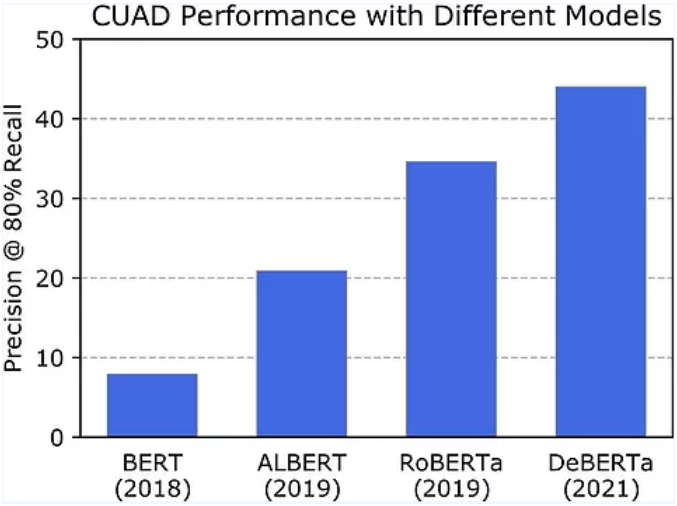
B. Revizija ugovora pomoću transformer modela

*The Atticus Projec*t[12] predstavlja neprofitnu organizaiju sa ciljem da ubrza efikasnost pregledanja ugovora primenom veštačke inteligencije. Zahvaljujući tom projektu, sada postoji labelirani skup podataka sa preko 13000 labela iz 510 ugovora preuzetih sa EDGAR-a. U ovom korpusu postoji 41 labela. Veliki značaj ovog skupa podataka ogleda se u tome da on trenutno predstavlja jedini veliki skup podataka koji postoji za ovaj problem pregledanja ugovora i specijalizovan je za NLP zahvaljujući tome što su klauzule u ugovorima ručno su anotirane od strane pravnih stručnjaka i više puta pažljivo proverene.

Glavni cilj rada[3] je primena transformer modela da se iz ugovora bojenjem izdvoje važne klauzule kako bi advokatima, ljudima, bio olakšan pregled ugovora.

Većina klauzula u ugovoru nije labelirana jer one nisu važne prilikom pregledanja ugovora, što dovodi do velike neravnoteže između relevantnih i nerelevantnih klauzula u

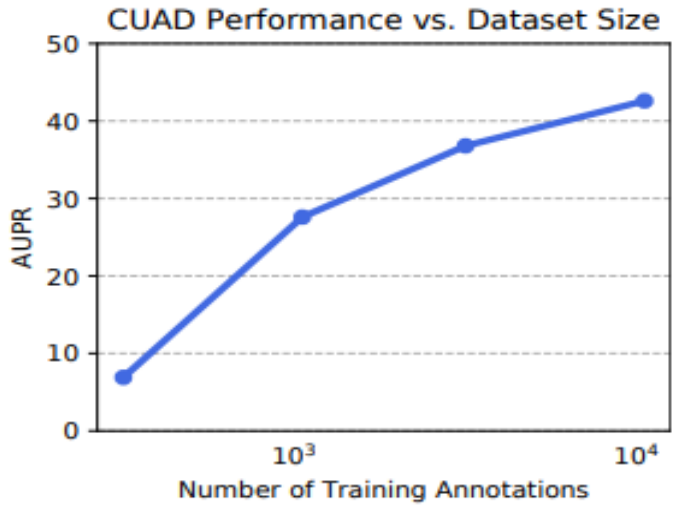
[3] https://github.com/google-research/bert
[4] https://en.wikipedia.org/
[5] https://eur-lek.europa.eu
[6] https://op.europa.eu
[7] https://www.legislation.gov.uk
[8] https://hudoc.echr.coe.int/
[9] https://www.echr.coe.int/
[10] https://case.lav
[11] https://www.sec.gov/edgar.shtml
[12] https://www.atticusprojectai.org/



Slika 1. Performanse različitih modela korišćenih u radu nad istim skupom podataka[6].

opozivu(eng. *Precision at recall)* mera performansi modela. Npr. u ugovorima postoji 100 relevantnih klauzula, a model izdvoji 500 klauzula, onda je preciznost modela 100/500=20%. Od tih 500, ako je 80 klauzula je relevantno, onda znači da je pronšao 80 od 100 relevntnih klauzula, pa je odziv 80%. Korisniku u ovom slučaju neće biti izdvojeno 20 važnih klauzula i moraće da pročita dodatnih 420 nerelevntnih klauzula. U ovom primeru mera je *20% precision@80% recall*.

Korišćena mera je i veličina površine ispod krive koja meri preciznost pri opozivu(eng. *Area Under the Precision-Recall curve*, AUPR). S obzirom na to da svaka preikcija ima svoju verovatnoću pouzdanosti, pa se može postaviti prag(eng. *Threshold*) kolika je minimalna vrednost pouzdanosti neophodna za predikciju, i na osnovu toga dobijati željene opozive, a zatim za svaku vednost opoziva izračunati preciznost.



Slika 2. Performanse modela u zavisnosti od veličine skupa podataka za obučavanje[6].

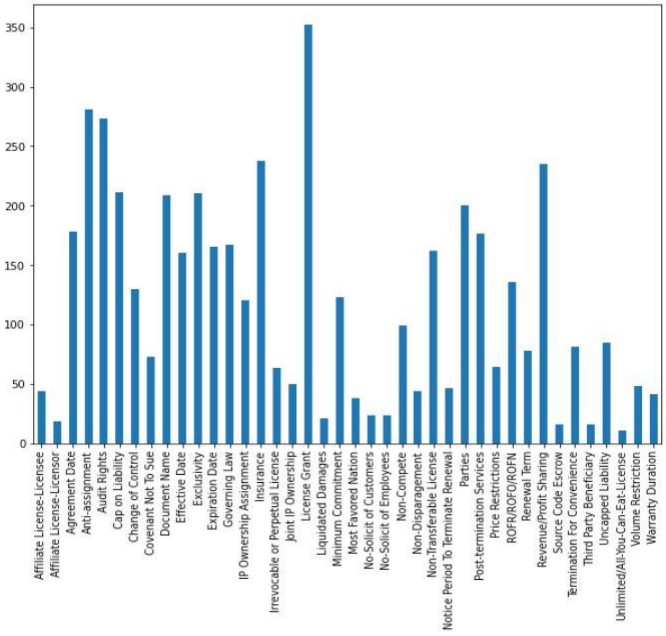
Modeli koji su korišćeni i njihove performanse su prikazani su na slici 1. Iz priloženog se može zaključiti da kako se poboljšavaju NLP modeli, tako se poboljšava i tačnost.

Međutim, mnogo veći uticaj na performance ima veličina skupa podataka. Na slici 2 prikazan je nagli skok performansi kada broj podataka dođe do 10^4, što nije ni čudo, s obzirom na to da su ovo kompleksni modeli i da se za obučavanje transformer modela koriste i milioni podataka.

Projekat Atikus ne prestaje sa daljim razvojem. Skup podataka, modeli i rezultati konstantno se poboljsavaju.

III. Skup podataka

Skup podataka je kreiran na osnovu 200 pravnih ugovora dostupnih u pdf i csv formatu. Za treniranje i testiranje modela korišteni su dokumenti koji sadrže labelirane klauzule u csv formatu. Svi ugovori su komercijalni i potiču iz sistema EDGAR koji koristi Komisija za hartije od vrednosti i berze SAD (SEC). Važna karakteristika ovih ugovora je da su kompleksniji i da sadrže veliki broj klauzula koje je teško naći u opštoj populaciji ugovora. Podaci su prikupljeni i labelirani od strane neprofitne organizacije *The Atticus Project* čiji je cilj da se iskoriste sve prednosti veštačke inteligencije u pravnom domenu, naročito u postupku pregleda pravnih ugovora. Svaki od komercijalnih ugovora spada u jedan od 25 tipova ugovora. Neki od tipova ugovora su: Sporazum o zajedničkom brendiranju (eng. *Co-Branding Agreement*), Ugovor o licenciranju (eng. *Licence Agreement*) itd. Iz tog razloga ugovori su prilično varirajućih dužina, od onih koji zauzimaju samo par strana, do kompleksnih ugovora sa preko 100 strana.



Slika 3. Broj primera koji pripadaju svakoj od 40 kategorija.

Organizacija *The Atticus Project* se zasniva na volonterima, među kojima su i pravni stručnjaci. Upravo oni su kreirali i ručno labelirali skup podataka tako da se sastoji od preko 4000 klauzula kategorisanih u jednu ili više kategorija. Ukupno je 40 kategorija, koje iskusni pravnici smatraju važnim u postupku pregledanja ugovora. Sve kategorije su međusobno nezavisne.

Neke od 40 kategorija su:

- *Governing Law* – klauzula u kojoj se navodi u kojoj državi se reguliše ugovor (primer: "*This Agreement is accepted by Company in the State of Nevada and shall be governed by and construed in accordance with thelaws thereof, which laws shall prevail in the event of any conflict*.");
- *License Grant* – klauzula u kojoj se navodi da li ugovor sadrži licencu koju je jedna od strana odobrila drugoj ugovornoj strani (primer: "*i-Escrow hereby grants to 2TheMart a worldwide, non-exclusive right to use, reproduce, distribute, publicly perform, publicly display and digitally perform the i-Escrow Content on or in conjunction with 2TheMart auctions*.");
- *Non-Disparagement* – klauzula kojoj se navodi da li postoji zahtev jedne ugovorne strane da druga strana ne sme da priča negativno o njoj. To je najčešće slučaj kod poslodavca i zaposlenog, gde zaposleni potpisuje da kompaniju u kojoj radi neće spominjati u negativnom kontekstu u bilo kojoj formi komunikacije (primer: „*The Company shall not tarnish or bring into disrepute the reputation of or goodwill associated with the Seller Licensed Trademarks or Arizona*.");
- *Expiration Date* – klauzula u kojoj se navodi kada ističe prvobitni rok ugovora (primer: "*The term of this Agreement shall continue for one (1) year following the Launch Date, unless earlier terminated as provided herein*.").

Dobijeni skup podataka je prilično nebalansiran, što se može videti na slici 3, zbog čega je isproban i *oversampling* podataka. Na primer, labela „*Liquidated Damages*" sadrži samo 21 primer (toliko klauzula pripada toj kategoriji), dok labela „*License Grant*" ima 352 primera. Takođe, podaci su nebalansirani i u smislu broja kategorija kojima labele pripadaju: 3751 klauzula pripada samo jednoj kategoriji, 422 labele pripadaju dve kategorije, 47 klauzula spadaju u 3 kategorije i samo 2 klauzule spadaju u 4 kategorije.

IV. METODOLOGIJA

A. Pretprocesiranje podataka

Priprema podataka (eng. *Data wrangling*) predstavlja proces prikupljanja, transformisanja i čišćenja podataka. Osim toga, mnogi drugi aspekti poput provere kvaliteta podataka, spajanja sa različitih izvora, poboljšanja sa drugim podacima i slično, predstavljaju bitan deo pripreme podataka[3]. Ovaj proces je krucijalan u postupku klasifikacije podataka, jer je to jedini način da sirovi podaci postanu upotrebljivi.

Prvi korak u pretprocesiranju podataka za *multi-label* klasifikaciju klauzula u pravnim ugovorima je prikupljanje podataka i spajanje labeliranih klauzula iz svakog *csv* dokumenta u jedan *dataframe*. Nakon toga smo izbacili nepotrebne kolone (npr. *Label 1-Answer, Label 2-Answer*). S obzirom da vrednosti kolona predstavljaju nazive kategorija, potrebno ih je tansformisati u nove kolone, gde će za svaku klauzulu vrednost tog reda i kolone iznositi 1 ako klauzula pripada toj kategoriji, a 0 ako ne pripada. Nakon ovog postupka *dataframe* ima 41 kolonu, gde su vrednosti prve kolone same klauzule, dok ostalih 40 predstavljaju svaku od kategorija kojima te klauzule potencijalno pripadaju, tj. ciljne labele, čije su vrednosti 1 ili 0.

U sledećoj fazi pretprocesiranja su najpre uklonjeni svi specijalni karakteri iz klauzula, zatim reči koje sadrže samo jedno slovo, višestruke prazne linije itd. Nakon toga je primenjen *stemming* algoritam. *Stemming* predstavlja postupak svođenja različitih gramatičkih oblika reči poput imenica, prideva, glagola itd. na njihov korenski oblik. Osnovna razlika između *stemming*-a i lematizacije je to što lematizacija vrši i morfološku analizu reči zbog čega je neophodno proslediti i informaciju o vrsti reči[6]. Završna faza pre *embedding*-a klauzula predstavlja izbacivanje zaustavnih reči (eng. *stop words*). *Stop words* predstavljaju reči koje se najčešće pojavljuju u svakom prirodnom jeziku. U svrhu analize teksta i formiranja NLP modela za za procesiranje teksta, ove reči najverovatnije neće dati nikakvu vrednost značenju dokumenta, zbog čega se najčešće izbacuju. Za klasifikaciju klauzula u PDF dokumentima, potrebno je najpre izvršiti tokenizaciju dokumenta (transformacija dokumenta u sekvencu rečenica).

Nakon pripreme i pretprocesiranja podataka prelazi se na fazu *word embedding*-a. *Word embedding* predstavlja transformaciju teksta u vektorski oblik koji kodira značenje reči tako da se očekuje da reči koje su slične po značenju imaju sličnu vektorsku reprezentaciju. Tehnike *embedding*-a korištene u ovom radu su sledeće: TF-IDF, *GloVe* i *Bert*.

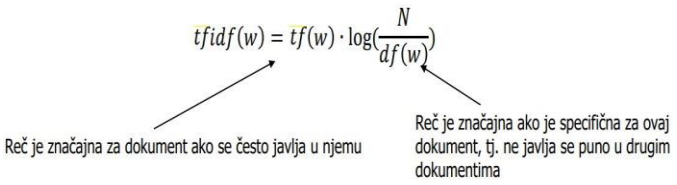
1) TF-IDF

Term Frequency–Inverse Document Frequency (TF-IDF) predstavlja najčešće korištenu statističku meru koja treba da odrazi koliko je neka reč relevantna za dokument. Zasniva se na dodeljivanju težina svakoj reči u korpusu, gde težine zavise od broja pojavljivanja reči u dokumentima. Drugim rečima, TF-IDF mera beleži relevanstnost reči, tekstualnih dokumenata i kategorija[7]. Veće vrednosti se dodeljuju rečima koje su specifične baš za dati dokument (u ovom slučaju klauzulu, s obzirom da se vrši klasifikacija klauzula).

TF-IDF se zasniva na kombinaciji dve različite metrike:

- Koliko puta se reč pojavljuje u dokumentu (klauzuli);
- Broj pojavljivanja reči u ostalim dokumentima (klauzulama).

Samim tim, reči koje se često pojavljuju u svakoj klauzuli imaju manju težinu iako se možda pojavljuju dosta puta u klauzuli koja se trenutno procesira (slika 4).



Slika 4. Formula TF-IDF mere; tf - *term frequency,* koliko se puta reč javlja u dokumentu; df - document frequency, broj dokumenata koji sadrže tu reč u korpusu; N - broj dokumenata u korpusu.

2) GloVe

GloVe predstavlja nenadgledan algoritam učenja za vektorsku reprezentaciju reči. Osnovna karakteristika ovog modela je da se oslanja na matricu brojeva međusobnog zajedničkog pojavljivanja reči (eng. *co-occurance matrix*). Ta matrica se formira za okolinu (prozor oko) reči. Ideja na osnovu koje se formiraju *GloVe* vektori je - Rastavljanje *co-occurance* matrice na delove pomoću metode koja se zove *Sigular Value Decomposition* (SVD). S obzirom da se postupak primene SVD u praksi pokazao kao problematičan, autori *GloVe*-a su SVD sveli na optimizacioni problem, zbog čega je napravljen jako robustan i skalabilan model. Jednostavno rečeno, *GloVe* omogućava da nad korpusom reči transformišemo svaku reč u određenu poziciju u višedimenzionom prostoru. To znači da će reči sa sličnim značenjem biti relativno blizu.

Za vektorizaciju pravnih klauzula, osim TF-IDF-a isprobali smo i pretrenirani GloVe vektor[13]. On predstavlja korpus reči gde je svaka reč praćena sa 100 brojeva koji opisuju vektor pozicije reči. Najpre je nad klauzulama primenjen *tokenizer* koji kreira rečnik indeksa, gde je vrednost svake reči broj njenih pojavljivanja u svim klauzulama. Nakon toga je od rečnika indeksa formirana sekvenca brojeva, nakon čega svaka rečenica sadrži numeričku reprezentaciju reči. S obzirom da sekvence nisu jednakih dužine nad njima je izvršena *pad_sequence* metoda (vrednosti koje nedostaju se popunjavaju 0). Zatim se primenjuje pretrenirani *GloVe* vector od koga se formira *embedding* matrica koja se kasnije će se kasnije koristiti kao vrednost početnih težina u *Embedding* sloju kod rekurentnih neuronskih mreža.

3) BERT

Pored prethodno navednih metoda, za *word embedding* isprobali smo i BERT. Ovaj model je nastao u *Google*-u kao primena transformer arhitekture 2018. godine. On je dostigao *state-of-the-art* performanse u mnogim NLP zadacima. Zasniva se na dve tehnike: maskiranje reči (eng. *Masked Language Model* – MDM), gde se maskira 15% nasumično izabranih tokena sa ulaza, i predikcija da li jedna rečenica sledi iza druge (eng. *Next Sentence Prediction* – NSP). BERT kodira ulaznu rečenicu, koja predstavlja ulaz, kroz tri tipa *embedding*-a :

- *Token embedding* (za svaku reč se dobija njen indeks u rečniku) – prvi token se koristi za klasifikaciju zajedno sa *softmax* slojem;
- *Sentence embedding* (za svaku reč se dodaje indeks rečenice kojoj pripada);
- *Positional embedding* (za svaku reč se dodaje njena relativna pozicija u rečenici).

Sam BERT se sastoji od enkoder blokova (12 ili 24, u zavisnosti da li je u pitanju *Base* ili *Large* model)[11]. U ovom radu korišten je pre-trenirani LEGAL-BERT model o kome je bilo više reči u prethodnom poglavlju.

B. Arhitektura rešenja

Za klasifikaciju klauzula koristili smo više različitih modela mašinskog učenja koji su dali različite rezultate. Korišteni modeli su: *Linear* SVC, K-*Nearest Neighbors* (KNN), *Decision Tree* algoritam, *XGBoost* i *Long short-term memory* (LSTM).

1) Linear SVC

Linear Support Vector Classifiers predstavlja algoritam mašinskog učenja čiji je cilj definisanje linearne granice u više-dimenzionalnom prostoru u cilju razdvajanja podataka koji pripadaju različitim klasama. Kada je u pitanju multi-label klasifikacija (kao što je ovde slučaj*) Linear SVC* koristi *One-vs-Rest* metod gde je cilj pronaći hiperravan koja najbolje kategorizuje podatke u n-dimenzionalnom prostoru. Identifikovanje odgovarajuće hiperravni zavisi od margine (maksimalna distanca od najbliže tačke podataka) i *loss* funkcije koja upravlja marginom[8]. *Loss* funkcija koju smo koristili za ovaj zadatak je *squared_hinge*. Kada su u pitanju ostali hiper-parametri:

- Parameter regularizacije C=10;
- Parametar *fit_intercept=True*, što znači da podaci neće biti centrirani u odnosu na koordinatni početak;
- Parametar *dual=False*, što se i preporučuje u slučaju kad je broj primera veći od broja *feature*-a.

Ostali parametri imaju podrazumevane vrednosti. *Linear* SVC je pokazao dosta bolje performanse u odnosu na druge *Linear* SVM implementacije i koristi mnoge druge napredne optimizacione tehnike, zbog čega predstavlja dobar izbor modela za klasifikaciju dokumenata[8]. Ovaj model smo trenirali nad TF-IDF vektorizovanim podacima.

2) K-Nearest Neighbors

Za problem *multi-label* klasifikacije klauzula isproban je i *k-nearest neighbors* algoritam. On predstavlja nadgledani algoritam mašinskog učenja koji se zasniva na sličnosti između novih slučajeva (podataka) sa istreniranim podacima (koji predstavljaju ulaz) na osnovu čega se predviđa klasa (u našem slučaju klase) kojoj novi slučaj pripada. On polazi od pretpostavke da se slične stvari nalaze u neposrednoj blizini. Bitno je napomenuti da je KNN neparametarski algoritam. U trening fazi KNN algoritam samo smešta podatke, i kada stigne novi primer (podatak) klasifikuje ga u kategoriju koja mu je najsličnija. Za računanje distance između podataka koristi se Euklidska distanca.

Kada je u pitanju *multi-label* klasifikacija koristeći KNN, što se se i koristi u ovom radu, algoritam se zasniva na primeni

[13] https://www.kaggle.com/danielwillgeorge/glove6b100dtxt

KNN algoritma na svaku pojedinačnu labelu: pronalazi se *k* najbližih primera za instancu testa i uzima u obzir samo dve kategorije: one koje pripadaju datoj labeli predstavljaju pozitivnu klasu, a ostale negativnu[9].

Jedini parameter koji je potrebno odrediti za ovaj algoritam je broj suseda. U našem slučaju *n_neighbors*=3. Poput *Linear* SVC modela, i ovaj model je treniran nad vektorima dobijenim TF-IDF *word embedding*-om.

3) Decision Tree Classifier

Pored navedenih algoritama, za problem *multi-label* klasifikacije korišten je i *Decision Tree* klasifikator. On predstavlja nadgledani algoritam mašinskog učenja gde se podaci kontinuirano dele na osnovu određenog parametra. Zasniva se na principu strukture stabla nalik dijagramu toka. Jedan je od najkorištenijih algoritama zbog lakoće implementacije, kao i jednostavnosti razumevanja u odnosu na druge algoritme za klasifikaciju[10]. Jednom kada se binarno stablo formira, klasifikacija testnih primera postaje jednostavna.

Za multi-label klasifikaciju, *Decision Tree* algoritam podrazumeva primenu nad svakom pojedinačnom labelom (*True* i *False*). Vrednosti parametara koje smo koristili su sledeći:

- Parametar koji predstavlja maksimalnu dubinu stabla *max_depth*=50;
- Parametar koji predstavlja funkciju koja meri kvalitet podele *criterion*= "*gini*";

Ostali parametri imaju podrazumevane vrednosti. Poput prethodna dva, i ovaj model je treniran nad TF-IDF podacima.

4) XGBoost

Još jedan model koji smo isprobali za klasifikaciju klauzula je *XGBoost*. On predstavlja *decion-tree* baziran ansambl algoritam mašinskog učenja zasnovan na *gradient boosting* algoritmu. On optimizuje *Gradient boosting* algoritam kroz paralelizaciju procesa, smanjenje veličine stabla, rukovanje nedostajućim vrednostima, visok nivo fleksibilnosti i regularizaciju kako bi se izbeglo preprilagođavanje.

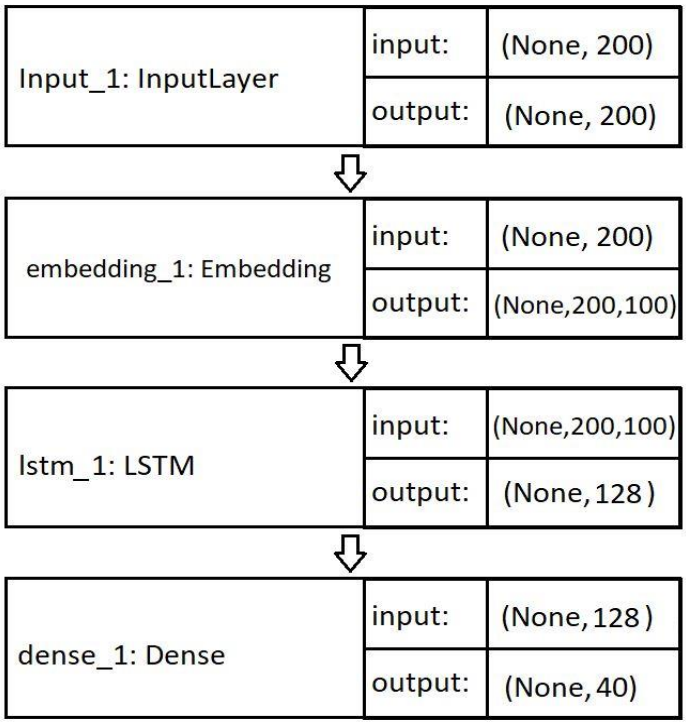
Hiper-parametri ovog modela koje smo optimizovali kroz validaciju su sledeći[12]:

- *n_estimators*=500 – Broj iteracija u treningu. Premali *n_estimators* može dovesti do male tačnosti modela, dok preveliki *n_estimators* može dovesti do preprilagođavanja zbog čega je bitno pažljivo odabrati odgovarajuću vrednost.
- *learning_rate*=0.05 – Veličina koraka u svakoj iteraciji dok se funkcija gubitka kreće ka minimum.
- *min_child_weight*=1 – Definiše minimalnu sumu težina svih listova stabla. Glavna svrha je da se izbegne *overfitting*.
- *gamma*=0.5 – Čvor se deli samo kada rezultujuće razdvajanje dovodi do smanjenja funkcije gubitka. *Gamma* specificira minimalno smanjenje gubitka potrebno za podelu.
- *max_depth*=3 – Maksimalna dubina stabla. Što je veća dubina stable, model je kompleksniji i bolji za predikciju, ali istovremeno i podložniji preprilagođavanju.
- *tree_method*= '*gpu_hist*' – Algoritam konstrukcije stable koji se koristi. Odabrani metod predstavlja GPU implementaciju *hist* algoritma.
- nthread=-1 – Broj paralelnih niti koji se koriste za treniranje modela.

Model je treniran i testiran nad vektorskim podacima dobijenim TF-IDF merom, kao i podacima koji su dobijeni uz pomoć pre-treniranog *Legal* BERT modela.

5) LSTM

Rekurentne neuronske mreže (eng. *Reccurent Neural Network* – RNN) predstavljaju jednu od najpopularnijih arhitektura koje se koriste u NLP-u. One su vrsta neuronskih mreža gde se izlaz neurona vraća na njegov ulaz, čime se pravi neka vrsta internog stanja mreže – memorije. *Long Short Term Memory* (LSTM) predstavlja specijalnu arhitekturu RNN-a, uz pomoć koje je rešen problem kratkoročne memorije kod rekurentnih mreža. Uvode se tri nove komponente: *Input gate*, *Forget gate* i *Output gate*. Sve komponente predstavljaju obične logističke sigmoidalne funkcije, zbog čega interno stanje (memorija) nije vidljivo spolja, već samo kroz aktivacionu funkciju.



Slika 5. Arhitektura LSTM mreže korištene za klasifikaciju klauzula u pravnim ugovorima.

Kao što se može videti na slici 5, arhitektura LSTM mreže korištene u ovom radu se sastoji od 4 sloja. Pristup koji smo koristili za *multi-label* klasifikaciju se zasniva na jednom izlaznom *Dense* sloju sa 40 izlaznih neurona, gde je vrednost svakog izlaza verovatnoća da klauzula pripada labeli na toj poziciji. Sigmoid aktivaciona funkcija vraća vrednost 1 ili 0. Ako je vrednost svakog izlaznog neurona veća od 0.5, smatra se da klauzula pripada labeli koju predstavlja taj neuron. *Embedding* sloj omogućava konverziju svake reči u vektor fiksne veličine. Rezultujući vektor ima stvarne vrednosti, umesto 0 ili 1. Za težine mu je prosleđena prethodno formirana *embedding* matrica uz pomoć *GloVe* pre-treniranog vektora. Parametar *trainable* je False, iz razloga što koristimo pre-trenirani *word embedding*. LSTM sloj je bidirekcioni sloj, što znači da model obrađuje sekvencu od početka do kraja, kao i unazad. Ovo je veoma bitno za razumevanje konteksta reči u rečenici. *Dense* sloj predstavlja potpuno povezan izlazni sloj. Funkcija gubitka (eng. *loss function*) koja je korištena pri kompajliranju modela je *binary_crossentropy*. Ona se koristi za zadatke binarne klasifikacije (odgovor na pitanje ima dve opcije – 0 ili 1) i bitna karakteristika je da može istovremeno prediktovani podatke za više klasa, što je pogodno za *multi-label* klasifikaciju. *Optimizer* korišten pri kompajliranju modela je *Adam*. Model je treniran u 200 epoha, a *batch size* iznosi 16.

V. Evaluacija i rezultati

Skup podataka, koji se sastoji od 4186 podataka, je podeljen na trening i test u razmeri 80:20. Za optimizaciju parametara kod *LinearSVC*-a, KNN-a, *Decision Tree* klasifikatora i LSTM-a korištena je unakrsna validacija (eng. *cross-validation*), dok je za *XGBoost* 20% trening skupa upotrebljeno za optimizaciju parametara, koristeći *PredefinedSplit* validator (princip indeksiranja – podatak koji spada u trening set ima indeks -1, dok podatak koji spada u validacioni set ima indeks 0). Razlog tome su hardverska ograničenje (treniranje modela je vremenski zahtevno). Unakrsna validacija se često primenjuje kada je skup podataka nad kojim se model trenira i testira relativno mali. Sastoji se iz sledeći koraka:

- Promešati trening skup;
- Podeliti trening skup u *k* grupa;
- Svaku pojedinačnu grupu posmatrati kao testni skup, a ostatak podataka kao trening skup, te evaluirati model nad testnim skupom;
- Izabrati parametre koji su dali najbolje rezultate.

Potrebno je unapred odrediti hiper-parametar *k* koji određuje na koliko grupa će se trening skup podeliti, što je u našem slučaju 5.

U tabeli 1 su prikazane ostvarene performanse isprobanih modela dobijene nad testnim skupom podataka.

Tabela 1. *f1 micro score* nad testnim skupom

Model	F1 micro score
TF-IDF + LinearSVC	0.75
TF-IDF + oversampling + LinearSVC	0.71
TF-IDF + KNN	0.59
TF-IDF + Decision Tree	0.65
TF-IDF + XGBoost	0.74
BERT + XGBoost	0.5
GloVe + LSTM	0.61

Mera evaluacije koje je korištena za sve navedene modele, kao i za njihovo međusobno poređenje je *f1_micro* mera. Ona se najčešće koristi za proveru kvaliteta *multi-label* binarnih problema, što je i ovde slučaj. Računa se kao srednja vrednost *f1_score* mere svih klasa. *F1_score* se računa po formuli:

$$2 \times ((Precision \times Recall) \div (Precision + Recall)).$$

Precision predstavlja odnos tačno pozitivnih predikcija (eng. *True positives*) i svih pozitivnih predikcija. Ako je broj netačno pozitivnih predikcija veći od broja tačno pozitivnih predikcija model će imati manji *precision*. Sa druge strane *Recall* predstavlja meru tačno pozitivnih predikcija u odnosu na sve tačno pozitivne primere. Idealna f1 micro mera iznosi 1, što bi značilo da model savršeno klasifikuje klauzule.

VI. Zaključak

U ovom radu razmatrali smo i isprobali različita rešenja za problem *multi-label* klasifikacije klauzula iz pravnih ugovora. Motivacija za izradu ovog sistema predstavlja to što su komercijalni ugovori često veoma dugi i kompleksni tekstovi, te bi automatizacija procesa praćenja i pregleda ugovora bila od velikog značaja za pravnike.

Skup podataka čini preko 4000 labeliranih pravnih klauzula u jednu ili više kategorija. Ukupan broj kategorija je 40. Nad podacima smo najpre primenili pretprocesiranje i *word embedding*. Za *word embedding* smo isprobali tri različite tehnike: TF-IDF, *GloVe* i BERT. Nakon toga smo za postupak klasifikacije klauzula trenirali i testirali više ralzičitih modela: *LinearSVC* u kombinaciji sa TF-IDF-om, KNN u kombinaciji sa TF-IDF-om, *Decision Tree* klasifikator u kombinaciji sa TF-IDF-om, *XGBoost* u kombinaciji sa TF-IDF-om i BERT-om i LSTM u kombinaciji sa GloVe-om. Model smo evaluirali nad testnim skupom. Mera evaluacije koja je korištena je *f1 micro* mera. Najbolji rezultat su ostvarili TF-IDF + LinearSVC (0.75) i TF-IDF + XGBoost (0.74).

Potencijalno poboljšanje performansi u odnosu na one koje postižu isprobani modeli bi se moglo postići najpre proširenjem skupa podataka: dodavanjem novih pravnih ugovora, klauzula,

kao i balansiranjem skupa podataka. Također, kompleksnije arhitekture neuronskih mreža uz optimizaciju više parametara bi potencijalno mogle ostvariti bolje performanse, kao i korištenje različitih transformer modela za klasifikaciju.

REFERENCES

[1] Dale, Robert. (2019). Law and word order: NLP in legal tech. Natural Language Engineering. 25. 211-217. 10.1017/S1351324918000475.

[2] How AI Is Being Used In The Legal Industry, Forbes, poslednji pristup: 30.03.2021.,
url:
https://www.forbes.com/sites/forbesbusinesscouncil/2021/01/19/how-ai-is-being-used-in-the-legal-industry/?sh=634b6e1550c6

[3] Hendrycks, Dan & Burns, Collin & Chen, Anya & Ball, Spencer. (2021). CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review.

[4] Chalkidis, Ilias & Fergadiotis, Manos & Malakasiotis, Prodromos & Aletras, Nikolaos & Androutsopoulos, Ion. (2020). LEGAL-BERT: The Muppets straight out of Law School. 2898-2904. 10.18653/v1/2020.findings-emnlp.261.

[5] G. Endel, F. and Piringer, H. (2015) 'Data Wrangling: Making data useful again', IFAC-PapersOnLine, 48(1), pp. 111–112.

[6] Ms. Anjali Ganesh Jivani, A Comparative Study of Stemming Algorithms, Anjali Ganesh Jivani et al, Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938, ISSN:2229-6093.

[7] Tao, Z.Y., Ling, G., & Chang, W.Y. (2005). An Improved TF-IDF approach for text classification. Journal of Zhejiang University Science, 6A(1), 49-55.I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[8] Raghavan A K, Venkatesh Umaashankar, and Gautham Krishna Gudur. 2019. Label frequency transformation for multi-label multi-class text classification. In GermEval 2019, 15th Conference on Natural Language Processing (KONVENS 2019), Erlangen, Germany.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[9] Tsoumakas G, Katakis I. Multi-label classification: An overview. Dept of Informatics, Aristotle University of Thessaloniki; Greece: 2006.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[10] Priyam, A., Gupta, R., Rathee, A., Srivastava, S., 2013. Comparative Analysis of Decision Tree Classification Algorithms. International Journal of Current Engineering and Technology Vol.3, 334–337. doi:JUNE 2013, arXiv:ISSN 2277 - 4106.

[11] Xiaofei Ma, Zhiguo Wang, Patrick Ng, RameshNallapati, and Bing Xiang. 2019. Universal Text Representation from BERT: An Empirical Study. arXiv:1910.07973 [cs].

[12] Li W, Yin Y, Quan X, Zhang H. Gene Expression Value Prediction Based on XGBoost Algorithm. Frontiers in Genetics. 2019;10. pmid:31781160