SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

POSLIJEDIPLOMSKI DOKTORSKI STUDIJ
ELEKTROTEHNIKE I INFORMACIJSKE TEHNOLOGIJE

KVALIFIKACIJSKI ISPIT

ANALIZA I PRIMJENA METODA AUTOMATSKOG SEMANTIČKOG OZNAČAVANJA TEKSTA

Daniel Vasić

Split, listopada 2018.

SADRŽAJ

1. UVOD

Proces pretvaranja nestrukturiranih podataka u strukturirane "strojno čitljive" podatke je složen proces, a uključuje pronalazak bitnih pojmova i njihovo sustavno uređivanje iz teksta napisanog na prirodnom jeziku. Budući da je proces ručnog strukturiranja podataka vrlo intenzivan i vremenski zahtjevan zadatak, postoji motivacija za automatizaciju ovog procesa. Pronalazak pojmova, koji igraju važnu ulogu u određenoj domeni, je vrlo važan prvi korak. Identifikacija pojmova, ali i definiranje odnosa između tih pojmova na semantičkoj razini izgleda kao gotovo nemoguć zadatak, no ipak postoje razni pristupi.

Pristupi temeljeni na pravilima koriste unaprijed zadana pravila ili heurističke obrasce za izvlačenje pojmova i njihovih međusobnih odnosa. Tipično se temelje na leksičko-sintaktičkim uzorcima. Na ovaj način možemo postići ograničene rezultate jer se pristupi temeljeni na pravilima oslanjaju na ekspertno znanje, a niti jedan stručnjak ne može definirati univerzalni skup pravila za ovako složen proces. Drugi pristup je temeljen na raspodjeli (*engl. distribution*) te modelira ovaj zadatak kao zadatak učenja, grupiranja i klasifikacije. Takvi pristupi se temelje na pretpostavci da se paradigmatski vezani pojmovi pojavljuju u sličnim kontekstima. Glavna prednost je u tome što su u mogućnosti otkriti odnose koji se ne pojavljuju u tekstu ili leksikonima na kojima su trenirani.

Danas se češće koriste metode temeljene na raspodjeli, koje su temeljene na strojnom učenju. Globalno pristupi temeljeni na raspodjeli se mogu podijeliti u dvije kategorije. Prva kategorija obuhvaća metode koja na osnovu ručno označenih leksičkih resursa pokušavaju strukturirati tekst na semantičkoj i sintaktičkoj razini i nazivaju se nadzirane metode. Druga kategorija ovaj problem promatra kao problem grupiranja elemenata sa sličnim obilježjima i zovu se nenadzirane metode. Oba ova pristupa daju dobre rezultate, ali u problemima sintaktičkog i semantičkog označavanja rečenica pristupi temeljeni na nadziranim metodama daju bolje rezultate. Osnovni nedostatak metoda nadziranog učenja je u tome što koriste ručno označene leksičke resurse koje je teško razviti. Takvi resursi zahtijevaju dobro poznavanje određenog jezika. Također, razvijanje takvog resursa je mukotrpan i dugotrajan proces. Nenadzirane metode ne zahtijevaju takve resurse već uče na osnovu neoznačenih podataka.

U ovom radu naglasak je na analizi metoda i primjeni strojne obrade prirodnog jezika za probleme semantičkog označavanja teksta ili u literaturi zadatka bolje poznatog kao

semantičko označavanje uloga (engl. *Semantic role labeling SRL*). Ovaj zadatak obuhvaća napredne metode shvaćanja smisla rečenice, te ima primjenu u brojnim zadacima obrade prirodnog jezika. Neki od zadataka su: automatsko generiranje pitanja u sustavima koji se temelje na pronalasku odgovora, strojnom prevođenju, ekstrakciji znanja iz teksta, pa čak i sažimanju teksta i strojnom prevođenju. Semantičko označavanje uloga je jedan od bitnijih problema u obradi prirodnog jezika, a može se primijeniti u svim zadacima gdje je potrebno razumijevanje smisla rečenice. Jedan od problema u semantičkom označavanju uloga na više jezika je nedostupnost leksičkih resursa na drugim jezicima osim na engleskom jeziku. Primjenom metoda nenadziranog strojnog učenja, ovaj problem možemo djelomično riješiti, no takvi pristupi ne daju rezultate koji se mogu dobiti primjenom nadziranih metoda strojnog učenja. Osim po metodi, semantičko označavanje uloga možemo podijeliti i prema vrsti leksičkog resursa, koje ćemo detaljno opisati u teorijskom pregledu.

Struktura ovog rada sadrži teorijski pregled semantičkih uloga u kojem smo naveli motivaciju za definiranje ovog zadatka. Glavnu ulogu u automatizaciji zadatka semantičkog označavanja teksta igraju leksički resursi. Nad ovim resursima se primjenjuju statističke metode koje modeliraju ovaj zadatak kao problem učenja i grupiranja. U teorijskom pregledu obuhvatili smo sve dostupne leksičke resurse na engleskom jeziku, s time da je jedno poglavlje iskorišteno za opis leksičkih resursa koji su dostupni na hrvatskom standardnom jeziku.

Poseban osvrt u teorijskoj osnovi smo pružili analizi metoda nadziranog strojnog učenja koje se često primjenjuju u obradi prirodnog jezika. U ovom poglavlju smo detaljno opisali primjene tehnika strojnog učenja na zadacima koji uključuju označavanje teksta. Nadzirane metode možemo podijeliti u dvije kategorije: metode koje se temeljene na ručno definiranim značajkama i metode temeljene na učenju značajki iz podataka. Proces ručnog definiranja značajki zahtijeva dobro poznavanje gramatike jezika kako bi odredili skup značajki koje služe za precizno semantičko označavanje uloga. Pristupi koji se temelje na učenju značajki iz teksta zahtijevaju jako velike količine teksta, te tipično koriste neuronske mreže za učenje semantičkih uloga. Oba ova pristupa zahtijevaju ručno označene leksičke resurse. Poseban naglasak u teorijskoj podlozi smo stavili na tehnike nadziranih metoda strojnog učenja, jer se mogu primijeniti u jezicima gdje postoje razvijeni leksički resursi. U ovom poglavlju dajemo osvrt na posebne vrste neuronskih mreža koje se koriste u zadacima obrade teksta. Posebno poglavlje smo odvojili za modele vektorskog prostora koji su danas popularni, a koriste se za smislen vektorski prikaz teksta.

Nakon analize postojećih metoda za obradu prirodnog teksta u sljedećem poglavlju dajemo osvrt na primjenu i analizu svih navedenih metoda u zadatku semantičkog označavanja teksta. Poglavlje dijelimo na pristupe koje se temelje na nadziranim i ne nadziranim metodama strojnog učenja. Poseban naglasak je na metodama nadziranog strojnog učenja jer ove metode u zadatku semantičkog označavanja uloga daju nemjerljivo bolje rezultate. Nenadzirane metode se često koriste u višejezičnim pristupima i jezicima gdje ne postoje velike skupine označenih podataka. Također u analizi rezultata smo dali usporedbe svih rezultata postignutih na nadziranim i nenadziranim metodama strojnog učenja. U poglavlju implementacije smo implementirali i vrednovali alat za semantičku obradu teksta na hrvatskom književnom jeziku. Prikazana je preciznost implementiranog alata i analiza modela. U procesu implementacije korišten je leksički resurs razvijen posebno za označavanje semantičkih uloga. Ovaj leksički resurs smo detaljno opisali u teorijskom dijelu.

Alat za semantičko označavanje uloga se može primijeniti u bilo kojem sustavu koji zahtjeva prepoznavanje značenja teksta i strukturiranja informacija. U zadnjem poglavlju opisujemo primjenu u inteligentnim tutorskim sustavima. Primjena alata za semantičku obradu teksta može uvelike doprinijeti razvoju inteligentnih tutorskih sustava zasnovanih na prirodnom jeziku. Ovo je posebna vrsta inteligentnih tutorskih sustava u kojemu se razvija komunikacija s učenikom na prirodnom jeziku. Ova vrsta komunikacije unutar inteligentnog tutorskog sustava je njegov najveći nedostatak [1]. Postoji nekoliko sustava koji pokušavaju riješiti problem komunikacije u inteligentnim tutorskim sustavima, a daleko najpoznatiji sustav je AutoTutor [2]. U okruženju hrvatskog standardnog jezika razvijen je, implementiran i primijenjen CoLaB Tutor (*engl. Controlled Language Based Tutor*) [3]. Obrada prirodnog jezika je područje koje je zadnjih godina napredovalo razvojem i primjenom neuronskih modela. Ovakvi modeli mogu se iskoristiti u komponentama inteligentnih tutorskih sustava kako bi poboljšali i olakšali proces komunikacije na prirodnom jeziku. U zadnjem poglavlju opisujemo moguću primjenu alata za semantičku obradu teksta u razvoju inteligentnog tutorskog sustava koji u procesu razumijevanja teksta koristi semantičko označavanje uloga.

2. TEORIJSKA OSNOVA SEMANTIČKOG OZNAČAVANJA TEKSTA

U obradi prirodnog jezika jedan od najbitnijih zadataka je određivanje značenja rečenica. Semantika se odnosi na aspekte značenja koji su izraženi u jeziku. Semantika se suprostavlja sa sintaksom tj. načinom na koji je nešto napisano i pragmatikom tj. primjenom znakova i njihovim međusobnim odnosom. Postoje brojni aspekti što je zapravo značenje kojega semantika izučava i kako ga opisati. U ovom radu obraditi ćemo način predstavljanja značenja unutar rečenica prirodnog jezika. Jedan od načina predstavljanja semantike unutar rečenice je korištenjem semantičkih uloga. U sljedećim poglavljima opisana je teorijska podloga o semantičkim ulogama, leksičkim resursima na engleskom jeziku i hrvatskom standardnom jeziku i primjena metoda strojnog učenja nad zadacima obrade teksta. Ova teorijska podloga služi kao temelj za analizu i primjenu raznih metoda u automatskom prepoznavanju semantičkih uloga.

2.1. Semantičke uloge

Pojam semantička uloga (engl. semantic role)[1] predstavlja odnose koji glagoli u rečenici imaju s ostalim riječima. Glagoli izražavaju semantiku događaja koji se opisuje kao relacijska informacija među sudionicima tog događaja, a projicira sintaktičku strukturu koja kodira tu informaciju. Glagoli su također vrlo promjenjivi, te prikazuju bogatu paletu semantičkog i sintaktičkog ponašanja. Glagolske klasifikacije pomažu sustavima za obradu teksta na prirodnom jeziku u organizaciji glagola u skupine koje dijele temeljna semantička i sintaktička obilježja. Semantičke uloge u osnovi opisuju konceptualne relacije između učesnika u određenoj rečenici. One ilustriraju osnovne „**Tko**, **Šta**, **Gdje**, **Kako** i **Kada**“ informacije unutar rečenice. Semantičke uloge otkrivaju semantiku učesnika unutar rečenice, tako da promjene u sintaksi ne utječu na njihove uloge. U cilju što boljeg razumijevanje smisla semantičkih uloga, promotrimo sljedeći primjer:

1. Sandy je razbila staklo (engl. Sandy shattered the glass).
2. Staklo je razbijeno od Sandy (engl. The glass was shattered by Sandy).

Obje rečenice imaju isti smisao, osoba pod imenom Sandy je izvršila radnju, razbila prozor. Prva rečenica prikazuje Sandi kao subjekt radnje i staklo (engl. glass) kao objekt radnje dok

[1]Naziv “semantička uloge” jedan od ustaljenih pojmova koji se često pojavljuje u suvremenoj lingvistici. Osim ovog naziva ustaljeni su nazivi “tematske uloge” (engl. thematic roles) i “dubinski padeži” (engl. deep cases).

druga rečenica sadrži pasivni oblik glagola razbiti te tako staklo postaje subjekt radnje. U ovoj opisanoj radnji semantičke uloge su vršitelj radnje (engl. Agent) i trpitelj radnje (*engl*. *Patient*). Sandi je vršitelj radnje, a staklo trpitelj radnje (onaj/ono nad kim/čim se radnja izvršava). Semantičke uloge su jednake unatoč promjeni sintakse. Ovaj koncept semantičkih uloga po prvi put u suvremenoj lingvistici pojavljuje se sredinom šezdesetih godina dvadesetog stoljeća pod nazivom padežna gramatika (engl. case grammar) [4]. Padežna gramatika je sustav lingvističke analize, a fokusira se na pronalazak veze između valencije glagola[2] i konteksta u kojem se glagol nalazi. Ova gramatika počiva na tvrdnji kako se morfološke i sintaktičke strukture svih jezika izvode iz "skrivenih" semantičkih kategorija, a ne iz sintaktičkih kategorija kako tvrdi generativna gramatika (engl. generative grammar) [5]. Utemeljitelj padežne gramatike je Charles Fillmore koji tvrdi da se dubinski padeži sastoje od grupe univerzalnih oznaka koji identificiraju tipove predrasuda koje ljudi identificiraju u događajima oko njih. U početku Charles Fillmore definira sljedeće padeže:

- *Agens (A)* - pokretač glagolske radnje koji u većini slučajeva označava osobu
- *Instrumental (I)* - predmet s kojim se izvršava glagolska radnja
- *Dativ (D)* - osoba ili živo biće koje je zahvaćeno glagolskom radnjom
- *Faktiv (F)* - predmet koji proizlazi kao produkt glagolske radnje
- *Lokativ (L)* - mjesto glagolske radnje
- Objektiv (O) - neživi entitet koji je zahvaćen radnjom

Ovim je definiran osnovni skup padeža, ali Filmore u izvornom članku navodi kako će "dodatni padeži biti sigurno potrebni", već uvidjevši da ovih šest padeža nije dovoljno. U svom radu [6], Fillmore proširuje na devet dubinskih padeža: *Agens, Doživljavač (Experiencer), Instrument, Objekt, Izvor, Cilj, Mjesto, Vrijeme, Put*. S konceptom dubinskih padeža razvija se koncept padežnog okvira, koji označava pravila prema kojima se padeži mogu kombinirati s glagolima. Ovim se definira nastanak rečenice koja se obavezno sastoji od modalnosti i propozicije, a propozicija je uvijek sastavljena od glagola i njegovih obaveznih dubinskih padeža [7]. Osnovni nedostatak ove teorije je nedovoljna nijansiranost uloga. S ovim se postavlja pitanje koja je granica između pojedinih uloga. David Dowty predložio je pojmove *Proto-Agent* i *Proto-Patient* koje se temelje na posljedicama koje se mogu ispitati pitanjima "*Je li argument promijenio stanje?*" ili "*Je li argument imao dobrovoljnu uključenost u radnju?*". Dowty u radu [8] tvrdi da ova svojstva razdvajaju argumente u leksikonu gdje se pridružuju klasičnom poimanju vršitelja i trpitelja radnje. Na primjer Proto-Patient često mijenja stanje i često na

[2]Valencija glagola je termin koji označava broj argumenata glagola

njega utječe drugi sudionik. Razni računalni resursi su razvijeni upravo na ovom poimanju semantičke uloge. Na te leksičke resurse tipično se primjenjuju statističke metode za izgradnju prediktivnih modela koji pokušavaju automatski odrediti semantičku ulogu riječi u tekstu. Takvi sustavi razvijeni su pomoću nadziranih algoritama strojnog učenja koji uče na temelju značajki koje su izvađene leksičkih baza podataka. Leksičke baze podataka su računalni resursi koji sadrže tekst napisan na prirodnom jeziku, a kojega su ručno označili stručnjaci iz područja lingvistike. Općenito ovakve baze podataka se zovu leksički resursi jer na različite načine modeliraju problem koji sa tim resursom pokušavaju riješiti. U nastavku opisati ćemo leksičke resurse koji se upotrebljavaju za prepoznavanje semantičkih uloga.

2.2. Leksički resursi

U ovom poglavlju prikazati ćemo leksičke resurse koji se koriste za semantičko označavanje uloga na engleskom jeziku. Na kraju prikazati ćemo leksički resurs koji se koristi za semantičko označavanje uloga na hrvatskom standardnom jeziku. Leksički resursi spomenuti u ovom poglavlju razliku se po namjeni i načinu označavanja. FrameNet i PropBank su dva leksička resursa koji se razlikuju po strukturi i načinu označavanja te iako služe za izvršavanje istog zadatka ne mogu se uspoređivati. U nastavku ćemo detaljnije obrazložiti smisao ovih leksičkih resursa te objasniti njihove primjene.

2.2.1. FrameNet

FrameNet [9][10] je projekt izgradnje leksičkog resursa na engleskom jeziku koji je jednako čitljiv ljudima i računalima. FrameNet korpus za jednog "običnog" čovjeka je rječnik koji sadrži više od 13,000 riječi i opisa njihovih značenja. Za istraživača u području obrade prirodnog jezika to je resurs sa preko 200,000 rečenica koje su ručno označene i povezane sa više od 1,200 semantičkih okvira. Ovaj resurs je jedinstven skup podataka za treniranje alata za semantičko označavanje uloga na engleskom jeziku. Općenito FrameNet korpus možemo kategorizirati kao valencijski rječnik. FrameNet se bazira na padežnoj gramatici Charlesa Fillmorea, a sastoji se od okvira, elemenata okvira i leksičkih jedinica. U smislu padežne gramatike okvir predstavlja predikat za koji su vezani elementi okvira. Elementi okvira su semantičke uloge, a leksičke jedinice su riječi koje se nalaze unutar elemenata okvira. Okvir je shematska reprezentacija situacije koja uključuje razne učesnike i druge konceptualne uloge. Elementi okvira pružaju dodatnu informaciju o semantičkoj strukturi rečenice. Razlikujemo ključne i ne ključne elemente okvira. Ključni elementi okvira su vrlo važni i najviše pridonose značenju cijelog okvira, a ne ključni elementi okvira su više deskriptivni (kao što je vrijeme,

mjesto, način itd.). FrameNet uključuje i informacije kako ovi elementi mogu biti korišteni u različitim kontekstima, što je jako važno za moguće alternacije dijateze[3]. Leksičke jedinice su osnovni oblici riječi koje također sadrže i govornu oznaku riječi (engl. part of speech). Za jedan okvir se može vezati više leksičkih jedinica, a jedna leksička jedinica može biti podijeljena između više okvira. Leksičke jedinice su sastavni dio rečenice. FrameNet također uključuje i odnose (engl. relations) između različitih okvira. Primjer rečenice iz FrameNet korpusa prikazan je na slici 2.1.

Slika 2.1. Shematski prikaz leksičkih elemenata, okvira i elemenata okvira označenih uz pomoć FrameNet anotacija

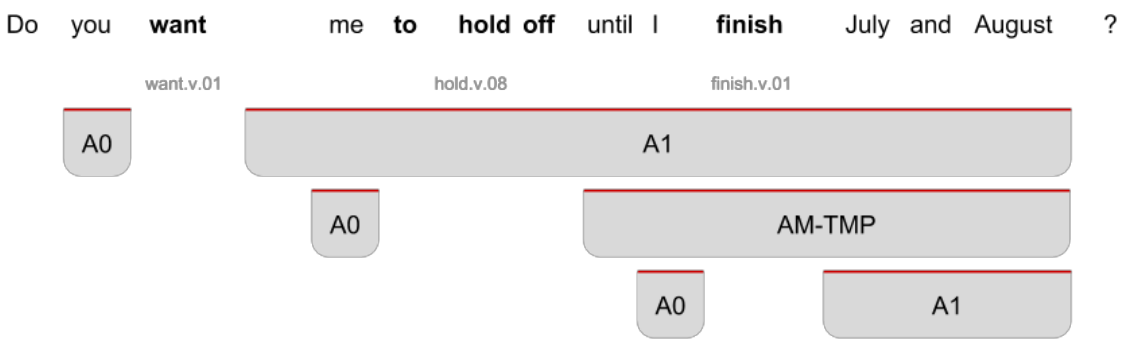
Na slici su prikazani sljedeći okviri: *INTENTIONALLY_ACT, DESIRING, HOLD_OFF_ON, FINISH i CALENDARIC_UNIT*. Sa svim ovim okvirima povezani su elementi okvira koji se mogu vezati za taj okvir. Recimo za leksičku jedinicu *Do* koja je prepoznata kao okvir *INTENTIONALLY_ACT* imamo jedan element okvira *ACT* koja prepoznaje vršitelja radnje u ovom okviru. Za leksičku jedinicu **want** koja je prepoznata kao okvir *DESIRING* imamo dva elementa, a oni su EXPERIENCER tj. vršitelj radnje, onaj koji izvršava radnju **wanting** i događaj koji se želi (**want**), a to je **to hold off**.

2.2.2. PropBank

PropBank [11], [12] je leksički resurs kojem je cilj pružiti široko rasprostranjen ručno označen korpus za semantičku obradu teksta. PropBank sadrži glagolske propozicije i njihove argumente, koje su označene ručno na rečenicama iz originalnog Penn Treebank korpusa [13]. Svaka uloga glagola je numerirana i opisana. PropBank je korpus koji je primarno orijentiran prema glagolima, a FrameNet je organiziran prema okvirima koji generaliziraju opise između sličnih glagola, ali i drugih vrsta riječi. PropBank ne uključuje oznake događaja koji su opisani uz pomoć imenica. Osnovna razlika između PropBank korpusa i FrameNet-a je u tome to je

[3]Alternacija dijateze se javlja kada smisao rečenice ostaje isti iako rečenica je napisana na različite načine

PropBank označen na uzastopnom tekstu, a FrameNet je označen na nasumičnim rečenicama. Samo u nekoliko primjeraka FrameNet sadrži označene kontinuirane dijelove teksta. PropBank sadrži oznake koje često su bliže sintaksnoj razini, a FrameNet je više semantički orijentiran. Od samog početka PropBank je zamišljen kao korpus za treniranje sustava za semantičko označavanje argumenata uz pomoć metoda strojnog učenja. Semantičke uloge u PropBank korpusu su numerirane započevši od 0. Za određeni glagol ARG0 je *Proto-Agent* dok je ARG1 Proto-Patient ili Tema. Uz ovo u PropBank-u su definirane općenite oznake koje se mogu primjeniti na bilo koji glagol. Skup uloga (*engl. roleset*) je grubi smisao predikata, koji uključuje niz uloga povezanih uz taj predikat, a ti argumenti su generalno brojčano označeni. Ove oznake imaju normaliziranu strukturu (Arg0, Arg1, Arg2, ArgM, ...) te ovakva inovacija omogućava pronalaženja između okvira [11].



Slika 2.2. Shematski prikaz leksičkih elemenata, okvira i elemenata okvira označenih uz pomoć PropBank anotacija

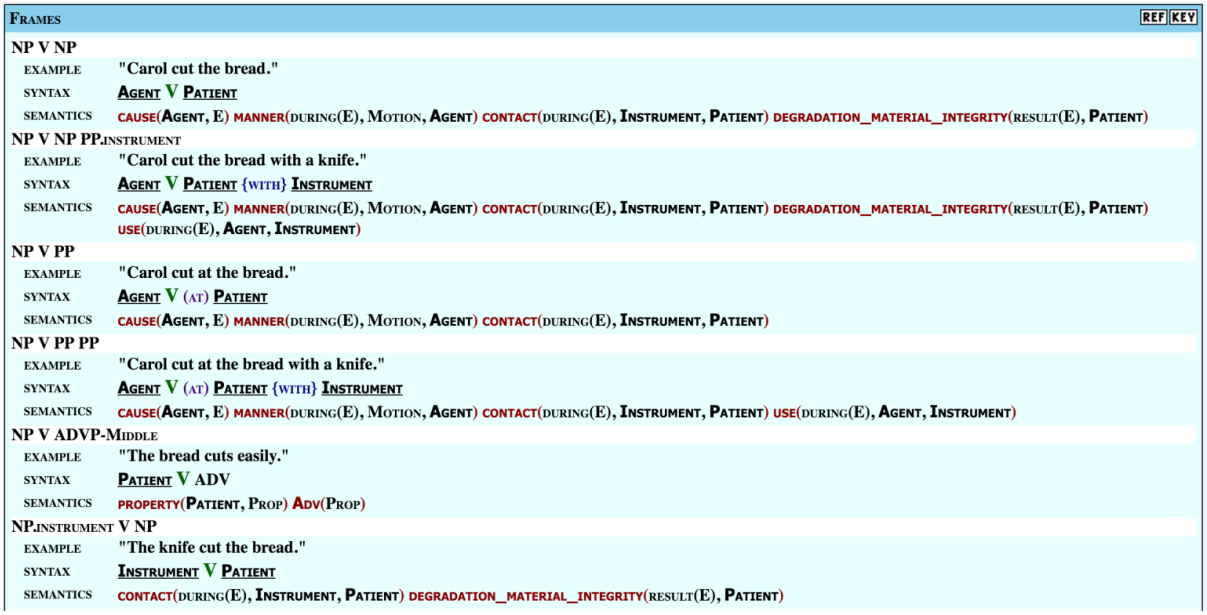
Na slici prikazani su predikati zajedno sa njihovim označenim značenjem nadalje za svaki predikat su označene semantičke uloge. U primjeru predikata **want (want.v.01)** koji se sastoji od dva semantička argumenta vršitelja radnje u ovom slučaju zamjenicu *you* i trpitelja radnje koji opisuje što osoba zapravo želi. Budući da je rečenica složena, sastoji se od više predikata. Drugi predikat označen u rečenici je **hold (hold.v.08)** i on se sastoji od dva argumenta, vršitelja radnje koji obustavlja/zadržava radnju te priložne oznake vremena koja opisuje do kada će radnja biti zadržana.

2.2.3. VerbNet

Glagolske klasifikacije pomažu sustavima za obradu prirodnog jezika da uspješno obavljaju organizaciju glagola u skupine koje dijele temeljna semantička i sintaktička obilježja. VerbNet

[14], [15] je najveći on-line glagolski leksikon trenutno dostupan za engleski jezik. To je hijerarhijski, domenski neovisan, široko pokriven glagolski leksikon s preslikavanjem prema drugim leksičkim resursima kao što su WordNet, FrameNet i PropBank. Ovaj leksikon je organiziran u glagolske klase koje proširuju Levin-ove [16] razrede sa dodatnim podrazredima. VerbNet se sastoji od oko 5800 engleskih glagola koji su grupirani u 270 klasa prema zajedničkim semantičkim ponašanjima.

Računalni glagolski leksikoni ključni su za izgradnju sustava za obradu prirodnog jezika usmjerenih prema semantičkoj obradi. Glagoli izražavaju semantiku događaja koji se opisuje kao relacijska informacija među sudionicima tog događaja, a projektira sintaktičku strukturu koja kodira tu informaciju. Glagoli su također vrlo promjenjivi, te prikazuju bogatu paletu semantičkog i sintaktičkog ponašanja. Razlika između PropBank-a i VerbNeta je u tome što je PropBank leksikon stvoren za zadatke treniranja alata za automatsko označavanje semantičkih uloga dok je VerbNet više organiziran oko opisa glagola i njegovih argumenata.


FRAMES REF KEY

NP V NP
EXAMPLE "Carol cut the bread."
SYNTAX AGENT V PATIENT
SEMANTICS CAUSE(AGENT, E) MANNER(DURING(E), MOTION, AGENT) CONTACT(DURING(E), INSTRUMENT, PATIENT) DEGRADATION_MATERIAL_INTEGRITY(RESULT(E), PATIENT)

NP V NP PP.INSTRUMENT
EXAMPLE "Carol cut the bread with a knife."
SYNTAX AGENT V PATIENT {WITH} INSTRUMENT
SEMANTICS CAUSE(AGENT, E) MANNER(DURING(E), MOTION, AGENT) CONTACT(DURING(E), INSTRUMENT, PATIENT) DEGRADATION_MATERIAL_INTEGRITY(RESULT(E), PATIENT) USE(DURING(E), AGENT, INSTRUMENT)

NP V PP
EXAMPLE "Carol cut at the bread."
SYNTAX AGENT V (AT) PATIENT
SEMANTICS CAUSE(AGENT, E) MANNER(DURING(E), MOTION, AGENT) CONTACT(DURING(E), INSTRUMENT, PATIENT)

NP V PP PP
EXAMPLE "Carol cut at the bread with a knife."
SYNTAX AGENT V (AT) PATIENT {WITH} INSTRUMENT
SEMANTICS CAUSE(AGENT, E) MANNER(DURING(E), MOTION, AGENT) CONTACT(DURING(E), INSTRUMENT, PATIENT) USE(DURING(E), AGENT, INSTRUMENT)

NP V ADVP-MIDDLE
EXAMPLE "The bread cuts easily."
SYNTAX PATIENT V ADV
SEMANTICS PROPERTY(PATIENT, PROP) ADV(PROP)

NP.INSTRUMENT V NP
EXAMPLE "The knife cut the bread."
SYNTAX INSTRUMENT V PATIENT
SEMANTICS CONTACT(DURING(E), INSTRUMENT, PATIENT) DEGRADATION_MATERIAL_INTEGRITY(RESULT(E), PATIENT)

*Slika 2.3. Shematski okvir glagola **rezati** iz VerbNet korpusa*

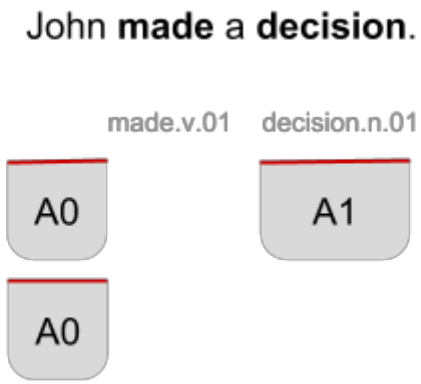
Na slici 2.3. je opisan okvir glagola **rezati (engl. cut)**. Okviri uključuju opis sintaktičkih struktura za razne oblike glagola, za svaku primjenu glagola dan je sintaksni prikaz, ali i semantički opis upotrebe pojedinog glagola. Semantički okvir detaljno opisuje radnju, u rečenici "Carol cut the bread" semantički prikaz ove radnje je:

- vršitelj radnje (engl. Agent) je uzrok ove akcije - **CAUSE(AGENT, E)**,

- način na koji vrši radnju je da uzrokuje gibanje tijekom događaja, a uzrok gibanja je vršitelj radnje - **MANNER(DURING(E), MOTION, AGENT),**
- ova radnja također zahtjeva kontakt između instrumenta s kojim se vrši radnja i trpitelja radnje tijekom događaja - **CONTACT(DURING(E), INSTRUMENT, PATIENT),**
- trpitelj radnje tijekom radnje se rastavlja na više cjelina što je opisano okvirom **DEGRADATION_MATERIAL_INTEGRITY(RESULT(E), PATIENT)**

2.2.4. NomBank

NomBank [17], [18] je proširenje PropBank corpusa na imenske predikate. Budući da u PropBank korpusu predikat je uvijek glagol u obzir nisu uzeti slučajevi koji se sastoje od imenice koja igra ulogu predikata. NomBank je projekt sveučilišta u New Yorku, a osnovni cilj ovog projekta je označavanje PropBank korpusa predikatima koji se javljaju u obliku imenice. Ovaj projekt je nastavljen nakon Nomlex [19] projekta te ga je dodatno nadogradio. PropBank, NomBank i ostali zabilježeni projekti trebali bi dovesti do stvaranja boljih alata za automatsku analizu teksta. NomBank argumenti su uključeni u CoNLL 2009 zadatak te predstavljaju dodatna ograničenja na proces automatskog označavanja semantičkih uloga.
Prvenstveno jedan od osnovnih problema je identifikacija predikata koja se sada proširuje i na imenske fraze. Primjer semantički označene rečenice sa dodanim NomBank anotacijama je prikazana na slici.



Slika 2.4. Shematski prikaz leksičkih elemenata, okvira i elemenata okvira označenih uz pomoć i NomBank anotacija

Na slici je vidljivo da pored glagola make koji je označen kao predikat, postoji i još jedan imenski predikat **decision.n.01** odluka kao imenica sa sobom povlači argument osobe koja je

donijela odluku. Nombank u svojim anotacijama pokušava obuhvatiti upravo ovakve situacije. Smisao predikata se dohvaća iz WordNet [20] semantičke baze podataka, a NomBank nad tim predikatima definira koje vrste argumenata može obuhvatiti.

2.2.5. SemLink

Cilj projekta SemLink je izgradnja poveznice između svih leksičkih resursa koji se preklapaju. Svaki od navedenih leksičkih resursa varira u razini i prirodi semantičkih detalja jer su neovisno stvoreni s različitim ciljevima. Ipak, svi ti resursi se mogu koristiti za povezivanje semantičkih informacija s propozicijama prirodnog jezika. SemLink služi kao platforma za objedinjavanje tih resursa i stoga kombinira finu granularnost i bogatu semantiku FrameNet-a, sintaktički utemeljene generalizacije VerbNet-a i relativno grubo zrnate semantike PropBank-a, za koje se pokazalo da su učinkoviti podaci u treniranju uz pomoć nadziranih tehnika strojnog učenja.

Način na koji SemLink povezuje leksičke resurse je uz pomoć mapiranja kojim se omogućava kombiniranje različitih vrsta informacija. Ova mapiranja mogu se koristiti za različite zadatke koji zahtijevaju zaključivanje i višu semantiku.

2.2.6. Leksički resursi na hrvatskom standardnom jeziku

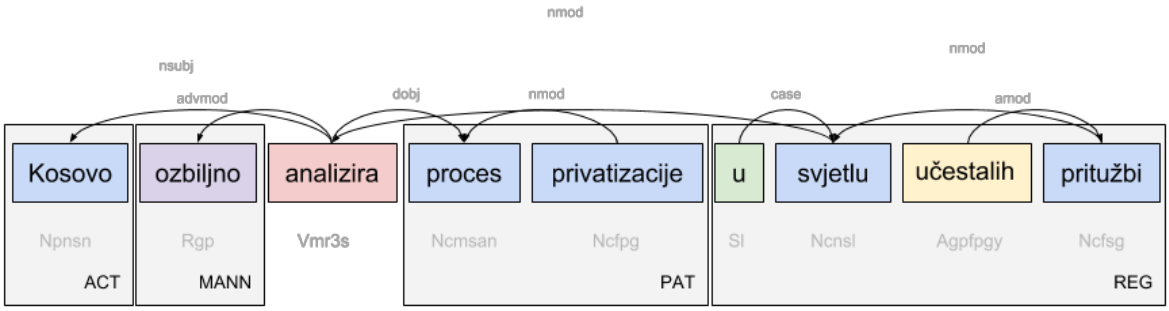
Osnovni nedostatak navedenih leksičkih resursa je dostupnost samo na engleskom jeziku jer je proces izgradnje takvog korpusa vrlo zahtjevan posao. Iako FrameNet je napravljen na njemačkom, španjolskom i japanskom jeziku, ovi resursi su znatno manji od originalnog FrameNet-a. Isto vrijedi i za PropBank korpuse koji su razvijeni na korejskom, kineskom, španjolskom i katalonskom jeziku. Kada usporedimo ove resurse s PropBank korpusom koji sadrži oko 113,000 struktura, resursi za druge jezike su dva do tri puta manji (npr. korpus na korejskom jeziku sastoji se od 33,000 semantičkih oznaka).

Korpus označen semantičkim ulogama na hrvatskom jeziku je trenutno u izgradnji. Iako postoji mnogo pristupa koji koriste paralelne korpuse kako bi trenirali višejezične alate za semantičko prepoznavanje uloga [21][22] ne postoji ni jedan sustav koji to radi automatski za hrvatski jezik. Razvoj ovakvog alata bi omogućio razvoj raznih alata za ekstrakciju informacija (engl. Information Extraction), sustave odgovaranja na pitanja (*engl. Question Answering Systems*), strojnog prevođenja (engl. Machine Translation) i brojnih drugih. Iako postoje brojni jezični resursi na hrvatskom jeziku, ipak ne postoji puno resursa koji bi se mogli iskoristiti za razvoj robusnih sustava za identifikaciju semantičkih uloga.

Jedan od najpoznatijih korpusa za hrvatski jezik je HrWaC [23] koji sadrži morfosintaktičke informacije o 1.9 milijardi pojavnica prikupljenih sa .hr domena. Ovaj korpus omogućava razvoj alata koji će izvršavati automatsko sintaktičko označavanje teksta no ne pruža informaciju o semantičkim strukturama unutar tog korpusa.

Također postoji i razvijeni valencijski rječnik glagola na hrvatskom jeziku pod nazivom CROVALLEX [24]. CROVALLEX je rječnik koji sadrži 1,739 glagola zajedno sa 5,118 valencijskih okvira. Osnovni nedostatak ovog leksikona je u tome što ne sadrži označene rečenice već "samo" primjere rečenica, no kao takav se može koristiti za određivanje smisla glagola u rečenici.

Korpus označen semantičkim ulogama na hrvatskom standardnom jeziku i paralelno slovenskom jeziku je razvijen u okviru projekta Instituta za jezikoslovlje i lingvistiku u Zagrebu i Jožef Stefan instituta u Ljubljani. Ovaj leksikon [25] sadrži 3,003 rečenice u korpusu za treniranje i 754 rečenice za testiranje. Sve rečenice su označene morfološko sintaktičkim ali i semantičkim oznakama kao što je prikazano na slici. Korpus ukupno sadrži 87,387 označenih tokena i predstavlja trenutno korpus za treniranje alata za prepoznavanje semantičkih uloga.



Slika 2.5. Shematski prikaz predikata, semantičkih okvira i sintaktičkih informacija kao što su stablo ovisnosti i govorne oznake za riječ na hrvatskom jeziku

U rečenici "Kosovo ozbiljno analizira proces privatizacije u svjetlu učestalih pritužbi" sve riječi su označene govornim oznakama prema MULTEXT-East [26][27] specifikaciji, također je stablo ovisnosti označeno preko univerzalnih ovisnosti (engl. universal dependencies) [28][29]. U rečenici su prikazani semantički okviri predikata **analizirati**. Semantički argumenti su vršitelj radnje (ACT), trpitelj radnje (PAT) te pogled na radnju (REG).

2.3. Primjena metoda strojnog učenja u zadacima obrade prirodnog jezika

Metode strojnog učenja danas se primjenjuju u raznim poljima, a posebno u obradi prirodnog jezika. Ove metode koriste podatke kako bi razvile statističke modele bez da su ručno programirane. One nadilaze programske instrukcije koristeći podatke kako bi donosile odluke i predikcije. Metode strojnog učenja se mogu podijeliti u dvije kategorije:

- metode koje zahtijevaju označene podatke ili nadzirane metode
- metode koje ne zahtijevaju označene podatke ili nenadzirane metode

Navedena je vrsta klasifikacije prema vrsti resursa za treniranje, ali metode strojnog učenja mogu se podijeliti i prema željenom izlazu sustava i to na:
metode klasifikacije koje dijele ulaz na dvije ili više klasa, a nakon treniranja sustav treba predvidjeti klasu neviđenih podataka,

- metode regresije gdje je izlaz iz modela kontinuiran,
- metode grupiranja dijele podatke u slične grupe, grupe prethodno nisu poznate,
- metode procjene gustoće pronalaze distribuciju na osnovu ulaznih podataka
- metode smanjenja dimenzija pretvara ulazne podatke u vektorski prostor nižih dimenzija.

Nadzirane metode strojnog učenja mogu se zasnivati na ručno definiranim značajkama ili mogu učiti iz teksta. Posebno su zanimljive metode temeljene na neuronskim mrežama koje u zadacima strojne obrade teksta daju impresivne rezultate. Još jedna prednost neuronskih mreža je što ne zahtjevaju ručno definirane značajke da bi postigle dobre rezultate. Neuronska mreža samostalno iz ulaznog skupa podataka "uči" najbolju reprezentaciju za zadatak koji obavlja.
Nadzirane metode strojnog učenja zahtijevaju ručno označene skupove podataka. Uglavnom se koriste leksički resursi koji sadrže veliki broj ručno označenih podataka. U većini zadataka potrebno je iz podataka pronaći informacije kojima se pronalazi najveća korelacija između ulaznih podataka i oznaka tih podataka. Ove informacije se zovu značajke (*engl. features*), a proces se zove izvlačenje značajki. Proces izvlačenja značajki omogućava ugradnju intuitivnih pravila koja omogućuju bolje rezultate klasifikacije, a također smanjuju dimenzionalnost ulaznih podataka. Nadzirane metode u obradi prirodnog jezika možemo podijeliti u dvije kategorije:

- metode temeljene na ručnom definiranju značajki
- metode koje uče značajke iz podataka.

Metode temeljene na značajkama zahtijevaju ugrađivanje stručnjakovog znanja unutar podataka za treniranje kako bi se stvorili složeni i precizni modeli. Proces definiranja značajki

često se zasniva na intuiciji stručnjaka i zahtjevan je posao. Metode koje uče iz teksta su najčešće neuronski modeli, ovi modeli pokušavaju na osnovu velikog broja parametara riješiti problem. U nastavku ćemo opisati metode temeljene na ručnom definiranju značajki, a posebno ćemo obraditi metode zasnovane na neuronskim mrežama.

2.3.1. Metode temeljene na ručnom definiranju značajki

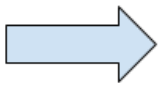
Za jezike koji imaju bogat skup ručno označenih podataka logičan je izbor nadziranih metoda strojnog učenja koji će iz podataka za treniranje "naučiti" razlikovati pojedine klase. Većina zadataka u obradi prirodnog jezika se mogu svesti na označavanje riječi u rečenici, bilo da se radi o sintaktičkoj ili semantičkoj obradi.

Ukoliko je $W = \{w_1, w_2, \dots, w_n\}$skup svih riječi u nekom jeziku, a $S = \{s_1^T, s_2^T, \dots, s_m^T\}, w_i \in s_j^T, w_i \in W, j = 1 \dots m$skup svih rečenica. Neka je $T = \{t_1, t_2, \dots, t_o\}$ skup svih oznaka riječi, a $O = \{o_1^T, o_2^T, \dots, o_m^T\}, t_i \in o_j^T, t_i \in T, j = 1 \dots m$skup svih vektora oznaka svake rečenice. Tada je potrebno pronaći funkciju koja će svakoj rečenici iz skupa *S*pridijeliti vektor iz skupa O.

Dobri rezultati se mogu postići definiranjem funkcije koja iz svake riječi izvlači niz značajki $\phi: W \to F, F = \{f_1^T, f_2^T, \dots, f_k^T\}, \forall f_\square = [\phi_1(w_\square), \phi_2(w_\square), \dots, \phi_l(w_\square)], w \in W$.

Funkcija značajki se sastoji od niza transformacija riječi koje izvlače informacije specifične za zadatak strojne obrade koji se izvršava. Skup svih značajki se enkodira u numeričke reprezentacije koje su pogodne za algoritme klasifikacije. Najjednostavniji način enkodiranja riječi numerički je zamjenom numeričkim identifikatorom koji će se koristiti umjesto te riječi. Zamjenom numeričkim identifikatorom javlja se problem jer numerički podaci nisu zapravo kategorički te na osnovu samih brojeva razni modeli mogu donositi pretpostavke koje nisu istinite. Na primjer, ukoliko riječi ***mačka, lav i pas*** predstavimo brojevima ***1, 2, 3,*** te ukoliko model interno u procesu treniranja i predikcije računa prosjek tada bi na osnovu zbroja vektora **`mačka`** i vektora **`pas`** dobili vektor **`lav`** što je očito pogrešno. Jedan način izbjegavanja ovih problema je binarizacijom kategorija u takozvane "one-hot" vektore. One-hot vektori predstavljaju kategoriju n dimenzionalnim jediničnim vektorom. Na slici 2.6. je prikazan proces kodiranja riječi iz prethodnog primjera uz pomoć one-hot vektora.

Naziv riječi	Kategorička vrijednost
Mačka	1
Lav	2
Pas	3



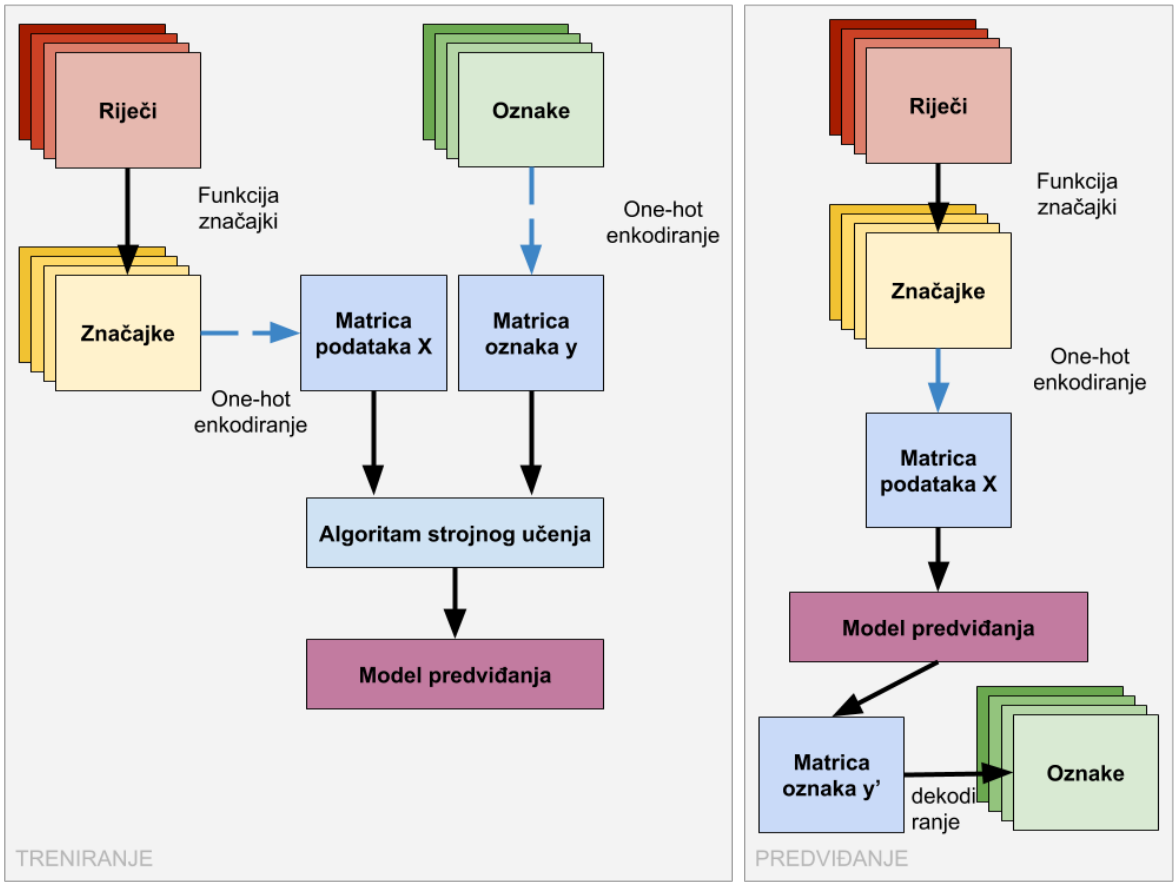
	Mačka	Lav	Pas
Mačka	1	0	0
Lav	0	1	0
Pas	0	0	1

Slika 2.6. Shematski one-hot predstavljanja riječi u rečenici

Skup značajki se uz pomoć upravo ovog pristupa transformira u matricu Xkoja se koristi za klasifikaciju. Na isti način se skup svih oznaka transformira u matricu y. U ovisnosti o zadatku, $y$može biti matrica ukoliko je zadatak predviđanja više klasa ili vektor ukoliko je je potrebno predvidjeti samo jednu klasu (svaki red matrice Xoznačen sa 0 ili 1) .

Općenito metode nadziranog strojnog učenja nastoje za ulaznu varijablu Xuz pomoć algoritma za klasifikaciju pronaći funkciju $f: X \rightarrow y$koja će ispravno mapirati ulaznu varijablu Xu izlaznu varijablu y. Ovaj problem se rješava primjenom statističkih modela na ručno definiranim značajkama. Time se omogućava uključivanje lingvističkog znanja kroz ručno definirane značajke i tako se postižu jako dobri rezultati.

Potrebno je pronaći značajke koji će dobro odvojiti podatke (*engl. discriminate*), ali i generalizirati model za uspješniju klasifikaciju neviđenih podataka. Trenirani model se primjenjuje na neoznačenim podacima kako bi dao predviđenu oznaku $y' \subseteq y$. Na slici 2.7. je prikazan proces izvlačenja značajki i treniranja modela te primjena modela na neoznačenom skupu podataka.

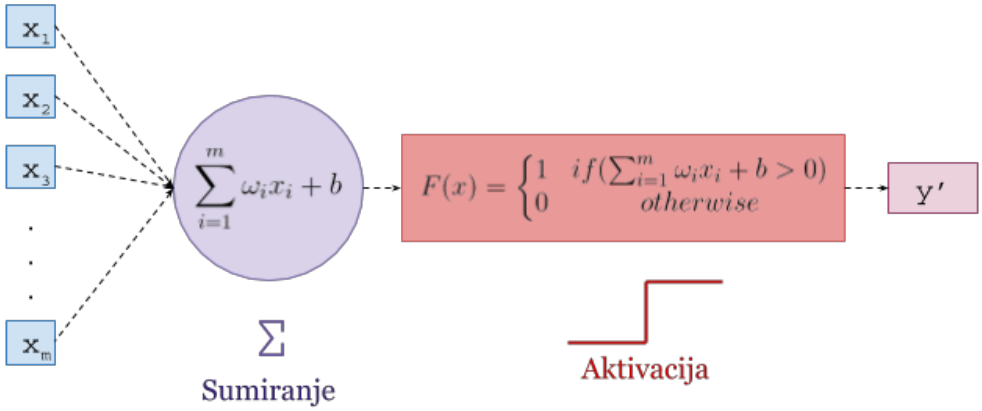


Slika 2.7. Shematski prikaz procesa treniranja i predviđanja metodama nadziranog strojnog učenja

Neke od često korištenih metoda za klasifikaciju i regresiju, čime se inače bave metode nadziranog strojnog učenja, su linearna i logistička regresija, strojevi s potpornim vektorima (*engl. Support Vector Machine*), naivni Bayes-ov klasifikator (*engl. Naive Bayes*), neuronske mreže (*engl. Neural Networks*) i brojni drugi pristupi.

2.3.2. Metode temeljene na neuronskim mrežama

Različite vrste neuronskih mreža se danas koriste u gotovo svim primjenama obrade prirodnog jezika. U nastavku objasniti ćemo neke od najvažnijih arhitektura u ovim zadacima, a posebno u prepoznavanju semantičkih uloga. U ovom poglavlju opisati ćemo kako funkcionira neuronska mreža, a početi ćemo od osnovne jedinice neuronske mreže, a to je neuron. Najjednostavnija vrsta neurona je perceptron [30], [31], razvijen je 1950-tih godina. Perceptron, u najjednostavnijem obliku, iz više ulaza stvara jedan jedinstveni izlaz.



Slika 2.8. Shematski prikaz perceptron neurona

Na slici 2.8. je prikazan jednoslojni perceptron sa ulazom $X = [x_1, x_2 \dots, x_m]$, i izlazom koji ovisi o parametrima neuronske mreže. Parametri $\Omega = [\omega_1, \omega_2, \dots \omega_m]$ predstavljaju relevantnu važnost ulaza u odnosu na izlaz, a $b \in B$određuju "pristranost" (*engl. bias*) određenoj klasi. Iz funkcije se vidi da veća vrijednost parametra $b$povećava mogućnost da će vrijednost funkcije pristranosti biti 1. Izlaz iz neurona y'određuje se na način da se ispita je li suma $\sum_j^{\square} x_j \omega_j$iznad određene granične vrijednosti kao što je prikazano u jednadžbi.

$$y' = \begin{cases} 0 & if \ \sum_{i=0}^{m} \omega_i x_i \leq granica \\ 1 & if \ \sum_{i=0}^{m} \omega_i x_i > granica \end{cases} \tag{2.1}$$

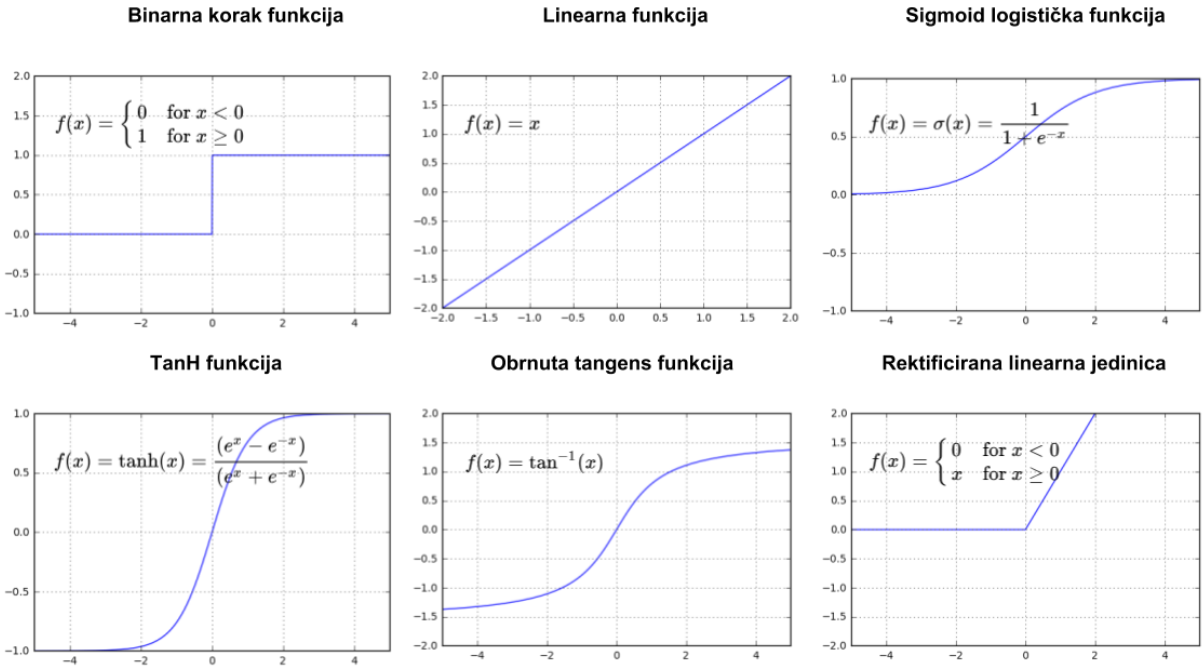
Granična vrijednost u ovoj jednadžbi označava koliko je sustav pristran pojedinoj klasi u procesu donošenja odluke. Ova vrijednost često se naziva vrijednost pristranosti. Težinske vrijednosti Ωsu podesivi parametri koji uz vrijednost pristranosti omogućavaju neuronskoj mreži da bude sustav koji se može trenirati. Navedeni opis se još dodatno može pojednostavniti, naime izraz $\sum_j x_j \omega_j$ se može zapisati kao umnožak dvaju matrica $\Omega \cdot X = \sum_j x_j \omega_j$. a graničnu vrijednost možemo prebaciti na drugu stranu jednadžbe i označiti sa vektorom $B = [b_1, b_2, \dots, b_l]$pri čemu je $l$broj slojeva neuronske mreže.

$$Y' = \begin{cases} 0 & if \ X \cdot \Omega + \mathrm{B} \leq 0 \\ 1 & if \ X \cdot \Omega + \mathrm{B} > 0 \end{cases} \tag{2.2}$$

Težinske vrijednosti i granična vrijednost se mogu podesiti kako bi se neuronska mreža ponašala na način koji želimo.

U ovom jednostavnom primjeru koristili smo jednostavnu korak funkciju (engl. step function) koja se u literaturi općenito zove prijenosna funkcija (engl. activation function). Težinska suma ulaznih vrijednosti proizvodi aktivacijski signal koji nastaje primjenom prijenosne funkcije. Često korištene prijenosne funkcije su linearna funkcija, korak funkcija, logistička sigmoid

funkcija, obrnuta tangens funkcija, rektificirana linearna jedinica *(engl. Rectified linear unit ReLu),* softmax funkcija te brojne druge. Na slici 2.9. su prikazani grafovi pojedinih najčešće korištenih prijenosnih funkcija.

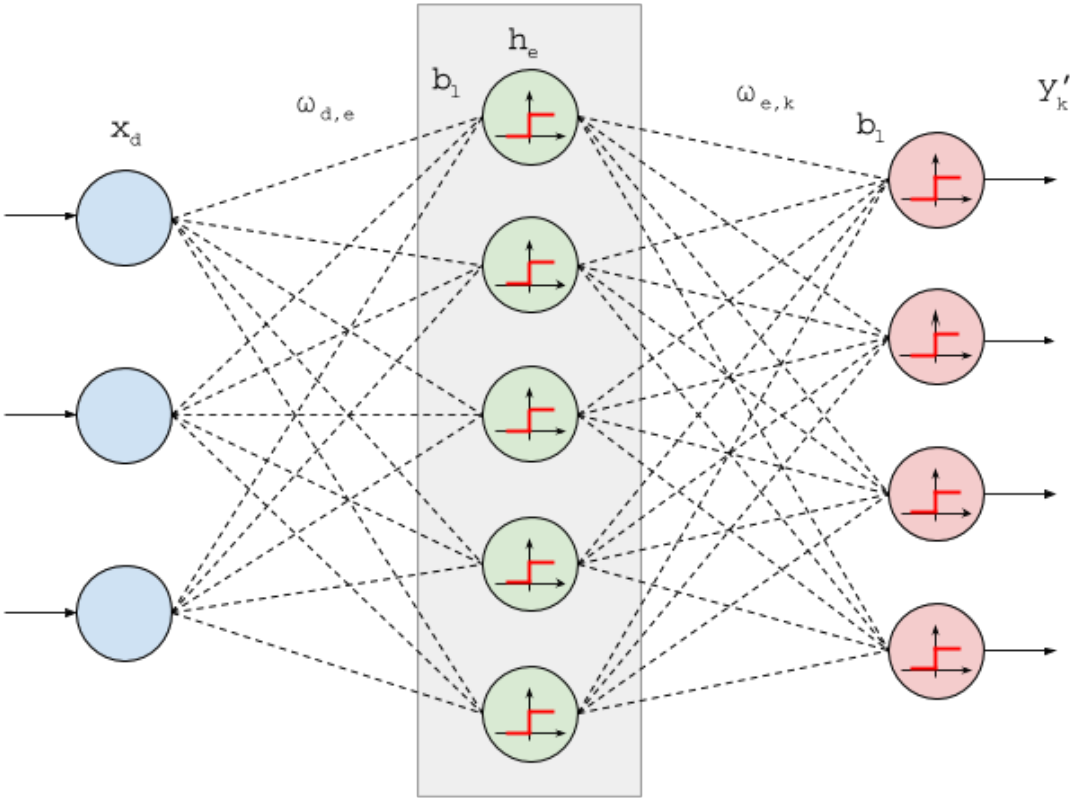


Slika 2.9. Grafovi prijenosnih funkcija korištenih u arhitekturama neuronskih mreža

Umjetna neuronska mreža (*engl. artificial neural networks ANN*) je nelinearni model neuronske mreže baziran na strukturi ljudskog mozga. Ovaj pristup se sastoji od umjetnih neurona i organiziran je oko tri međusobno povezana sloja:

- ulazni sloj (engl. input layer),
- skriveni sloj (engl. hidden layer) i
- izlazni sloj (engl. output layer).

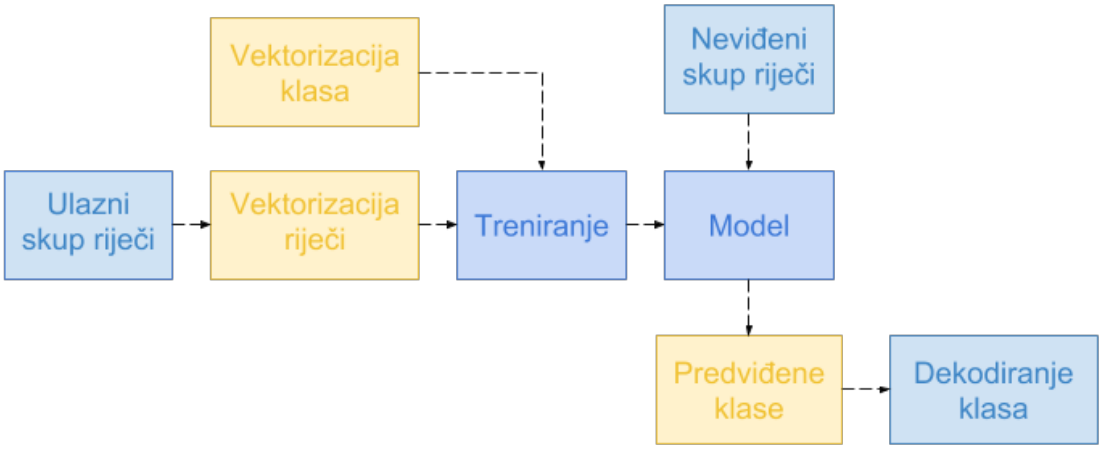
Ulazni sloj se sastoji od ulaznih neurona koji su povezani sa skrivenim slojem i podaci se šalju iz ulaznog sloja u skriveni sloj. Nadalje, podaci se iz skrivenog sloja šalju u izlazni sloj. Svaki neuron sastoji se od težinske vrijednosti ulaza i prijenosne funkcije te jednog izlaza. Treniranje je proces optimizacije težinskih vrijednosti koji se svodi na minimizaciju pogreške predviđanja neuronske mreže i istinitih vrijednosti. Razlika između predviđanja i istinitih vrijednosti se računa uz pomoć posebne funkcije koja se zove funkcija gubitka. Metoda koja se koristi kako bi se odredila pogreška između svakog neurona se zove propagacija unazad (engl. backpropagation), preko kojeg se računa spust funkcije gubitka. Na slici 2.10. je prikazana arhitektura višeslojnog perceptrona (engl. Multilayer perceptron MLP) [32][33].



Slika 2.10. Prikaz višeslojnog perceptrona sa tri sloja

Višeslojni perceptron je neuronska mreža temeljena na ANN koja se sastoji od tri ili više slojeva. Svaki neuron u prethodnom sloju je povezan sa svim neuronima u sljedećem sloju, stoga je ova arhitektura neuronske mreže potpuno povezana (*engl. fully connected*).

MLP se često koristi u obradi prirodnog jezika u zadacima kao što je označavanje dijelova rečenice (*engl. part of speech tagging*). Na slici je prikazana arhitektura MLP-a gdje je $x_j \in X$ ulazni vektor pri čemu je $d$dimenzija ulaznog vektora, a $k$je dimenzija izlaza iz neuronske mreže. Težinske vrijednosti $\omega_{d,e}, \omega_{e,k} \in \Omega$ i granične vrijednosti $b_l \in B$ su podesivi parametri neuronske mreže. Vrijednosti skrivenih slojeva $h_e \in H$su stanja neuronske mreže nastalih primjenom prijenosnih funkcija nad podesivim parametrima. Funkcioniranje MLP-a se objašnjava na primjeru klasifikacije teksta gdje se ulazni skup riječi preslikava u klasu. Ulaz u neuronsku mrežu je rijetka matrica Xkoja se dobiva vektorizacijom ulaznih podataka, a mreža se trenira na osnovu matrice y. Ova matrica se dobiva vektorizacijom klasa. Postupak vektorizacije ulaznih podataka je identičan za bilo koji tip ulaznih podataka. bilo da se radi o značajkama koje sustav smatra kategoričkim vrijednostima ili sirovim tekstualnim podacima.



Slika 2.11. Način klasifikacije teksta putem neuronskih mreža

Kao što je prikazano na slici, za klasifikaciju teksta putem neuronske mreže potrebno je pretvoriti podatke u odgovarajući oblik. Tekstualne podatke koje želimo klasificirati potrebno je pretvoriti u numeričku matricu. Vreća riječi (engl. *Bag of Words*) je poznata metoda koja se često koristi za pretvaranje rečenica u vektorski oblik. Ukoliko imamo skup svih riječi Wi ukoliko imamo skup svih rečenica S, tada matrica Xprikazana u jednadžbi x predstavlja matrični zapis svih rečenica. Matrica ima redova koliko ima rečenica u korpusu, a broj stupaca je određen brojem jedinstvenih riječi u skupu W. Ukoliko riječ $w_j \in W$se ne nalazi u rečenici $s_i \in W$tada vrijednost $x_{i,j} = 0$inače je vrijednost jednaka jedan $x_{i,j} = 1$.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,d} \end{bmatrix}, x_{i,j} = \begin{cases} 0 & if\ w_j \notin s_i^T \\ 1 & else \end{cases}, i = [1, \cdots, m], j = [1, \cdots, d] \qquad (2.3)$$

Ovakva vektorska reprezentacija rečenice sastoji se od jako puno nula te nekoliko jedinica u ovisnosti o duljini rečenice. Učenjem kroz ovakve vektorske reprezentacije neuronska mreža daje relativno dobre rezultate na zadacima klasifikacije dokumenta i rečenica, ali ne i pojedinih riječi. Za klasifikaciju riječi čest pristup je da se enkodira riječ kao "one-hot" vektor i njen kontekst i tako za svaku riječ u rečenici. Svaka riječ je označena oznakom iz skupa $T = [t_1, t_2, \ldots, t_k]$. Za svaku oznaku riječi formira se matrica Y koja se sastoji od redaka čiji broj ovisi o broju riječi/rečenica, a broj stupaca ovisi o broju oznaka koje se kodiraju.

$$X = \begin{bmatrix} y_{1,1} & \cdots & y_{1,d} \\ \vdots & \ddots & \vdots \\ y_{m,1} & \cdots & y_{m,k} \end{bmatrix}, y_{i,j} = \begin{cases} 0 & if\ t_j \notin o_i^T \\ 1 & else \end{cases}, i = [1, \cdots, m], j = [1, \cdots, d] \qquad (2.3)$$

Matricu X množimo sa težinskim vrijednostima Ω i dodajemo vektor B, nakon čega primjenjujemo sigmoid prijenosnu funkciju σ. Ovaj postupak možemo ponavljati više puta, na izlazu iz prethodnog sloja se primjenjuje identičan proces kao što je prikazano jednadžbom.

$$Y' = \sigma_l(\dots(\sigma_2(\sigma_1(X \cdot \Omega + b_1) + b_2) + \cdots + b_l), \sigma = \frac{1}{1+e^{-x}} \quad (2.4)$$

Matrica Y'je vrijednost koja se pojavljuje na izlazu iz neuronske mreže. Potrebno je izračunati koliko dobro neuronska mreža predviđa vrijednosti matrice Y. Funkcija koja računa koliko predviđene vrijednosti Y'odstupaju od Yzove se funkcija gubitka i često se označava sa J. Ona je u najjednostavnijem obliku jednaka kvadratnoj razlici dvaju vrijednosti. Ova kvadratna razlika se zove srednja kvadratna greška (*engl. mean square error MSE*).

$$J_{MSE} = \frac{1}{2}\left|\left|Y - Y'\right|\right|^2 \quad (2.5)$$

Ovom jednadžbom definiramo mjeru koliko predviđanja neuronske mreže odstupaju od traženih vrijednosti. Sada želimo vidjeti koliko promjene težinskih vrijednosti utječu na promjenu vrijednosti funkcije. Cilj algoritma propagacije unazad (*engl. backpropagation*) je ažuriranje težinskih vrijednosti tako da izlaz neuronske mreže bude što sličniji traženom izlazu. Time se smanjuje greška izlaza iz svakog neurona, a time i cijele neuronske mreže. Drugim riječima zanima nas koliko težinska vrijednost Ω utječe na rezultat J_{MSE} funkcije. Ovo bi se moglo zapisati kao parcijalna derivacija J_{MSE} prema Ω. Ova parcijalna derivacija primjenom lančanog pravila može se zapisati kao umnožak parcijalnih derivacija funkcije gubitka J_{MSE} prema parametrima izlaza neuronske mreže O, parcijalne derivacije funkcije izlaza iz neuronske mreže prema izlazu iz prijenosne funkcije ∂Net.

$$\frac{\partial J_{MSE}}{\partial \Omega} = \frac{\partial J_{MSE}}{\partial \mathrm{O}} * \frac{\partial O}{\partial \mathrm{Net}} * \frac{\partial Net}{\partial \Omega} \quad (2.6)$$

Za izračun ovih derivacija računaju se parcijalne derivacije prema sljedećim jednadžbama:

$$\frac{\partial Net}{\partial \Omega} = \frac{\partial}{\partial \Omega}(X \cdot \Omega) \quad (2.7)$$

$$\frac{\partial O}{\partial \mathrm{Net}} = \frac{\partial}{\partial \mathrm{Net}}\sigma(X \cdot \Omega + B) = \sigma(X \cdot \Omega + B)(1 - \sigma(X \cdot \Omega + B)) \quad (2.8)$$

$$\frac{\partial J_{MSE}}{\partial \Omega} = \frac{\partial}{\partial \Omega} * \frac{1}{2} * (Y - Y') = Y' - Y \quad (2.9)$$

Određena težinska vrijednost se ažurira na način da se vrijednost parcijalne derivacije oduzme od trenutne vrijednosti pri čemu se parcijalna derivacija množi s parametrom η koji se zove stopa učenja (*engl. learning rate*).

$$\Omega = \Omega - \eta * \frac{\partial J_{MSE}}{\partial \Omega} \quad (2.10)$$

Računanje parcijalnih derivacija za svaki neuron, težinsku vrijednost i ulaz u neuronskoj mreži je vrlo zahtjevan posao. Štoviše porastom broja podataka raste broj parametara koje neuronska mreža mora izračunati. Dobar način aproksimacije parametara je putem algoritma pod nazivom stohastičko spuštanje gradijenta (*engl. Stochastic Gradient Descent SGD*) [34]. Ovaj algoritam nasumično uzima primjerke za treniranje te ažurira parametre neuronske mreže na osnovu samo jednog primjerka iz skupa za treniranje. Broj primjeraka koji se koriste za izračun

parcijalnih derivacija naziva se hrpa (*engl. batch*). SGD je algoritam koji koristi hrpu veličine jedan za izračunavanje spusta funkcije i nasumično miješa podatke tijekom treniranja kako ne bi ažurirao parametre prema uvijek istim trening podacima.

2.3.3. Modeli vektorskih prostora

Računalo u većini zadataka obrade pa čak i prirodnog jezika zahtjeva numeričke podatke, a ne tekstualne podatke. Tipično se koriste "one-hot" vektori koji kodiraju kategoričke informacije u numerički vektorski prostor. Ovaj pristup se nije pokazao učinkovitim jer su ovi kategorički identifikatori proizvoljni i ne pružaju nikakvu dodatnu informaciju o povezanosti pojedinih riječi. Na primjer riječ **`mačka i pas`** mogu imati proizvoljne identifikatore, no ti identifikatori ne obuhvaćaju informacije o tome da su mačke i psi sisavci, životinje sa četiri noge te bojne druge zajedničke osobine. Problem koji se javlja sa korištenjem "one-hot" vektora je ukoliko imamo jako puno riječi, matrice će biti rijetke (*engl. sparse*) te dobivamo podatke koji nisu najpogodniji za treniranje statističkih modela.

Kako bismo nadišli ovaj problem koriste se modeli vektorskih prostora (*engl. Vector space models VSM*). Ovi modeli riječi predstavljaju u kontinuiranom vektorskom prostoru gdje se riječi sa sličnim semantičkim značenjem nalaze blizu. VSM-ovi imaju dugu i bogatu povijest u obradi prirodnog jezika i svi se temelje na distribucijskoj hipotezi[4]. Metode koje se temelje na ovoj hipotezi mogu se podijeliti u dvije kategorije:

- metode temeljene na prebrojavanju i
- metode temeljene na predviđanju [35].

Najpoznatiji pristupi temeljeni na prebrojavanju su:

- latentna semantička analiza (*engl. Latent Semantic Analysis LSA*) i
- GloVe (*engl. Global Vector representations*).

Pristupi temeljeni na predviđanju su neuronski modeli, a najpoznatiji su:

- Word2Vec
- FastText.

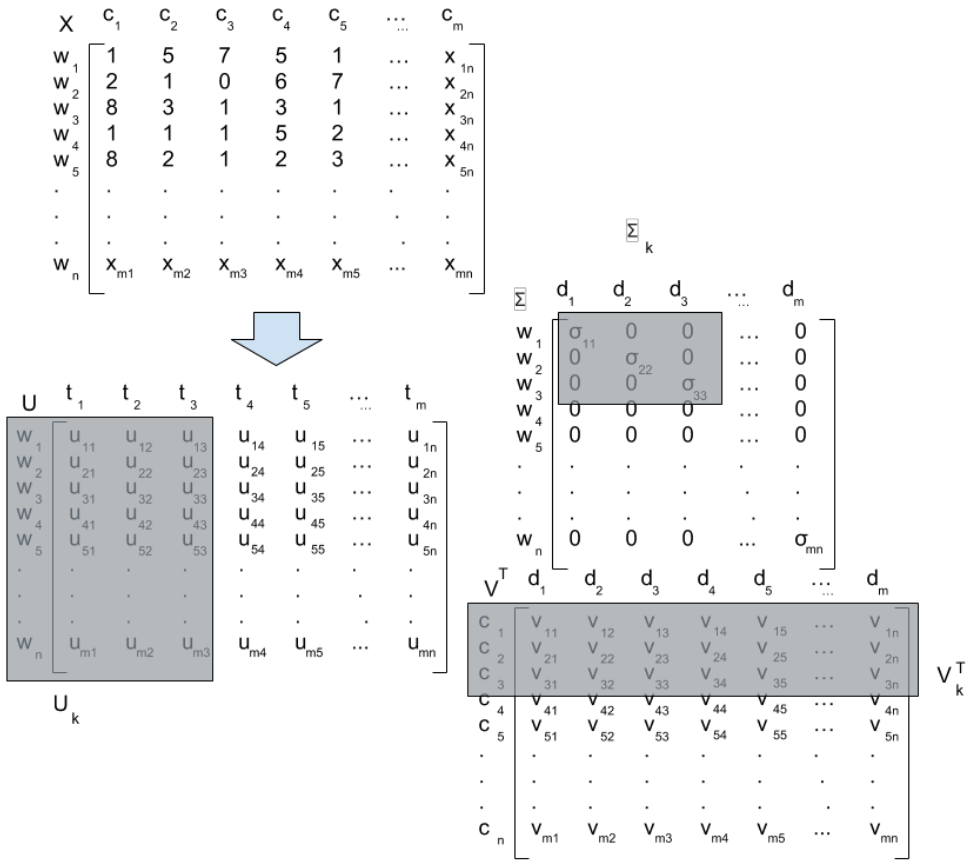
U nastavku ćemo opisati navedene metode predstavljanja riječi vektorskom reprezentacijom.

2.3.3.1 Latentna semantička analiza (LSA)

Latentna semantička analiza [36] je tehnika u obradi prirodnog jezika koja se često koristi u analizi povezanosti dokumenta sa terminima. Neka je skup svih riječi označen sa $W =$

[4]Distribucijska hipoteza je hipoteza koja kaže da riječi koje se nalaze u sličnim kontekstima dijele značenje. Poznat je citat Johna Ruperta Firtha koji kaže da se riječ prepoznaje prema društvu u kojem se nalazi.

$\{w_1, w_2, \dots, w_n\}$i skup svih dokumenata označen sa $C = \{c_1, c_2, \dots, c_m\}, w_i \in c_j, w_i \in W, j = 1 \dots m$. Nad ovim podacima definiramo matricu X koja sadrži brojeve $x_{i,j}$koji predstavljaju broj pojavljivanja riječi $w_i$unutar dokumenata c_j. Tada je $t_i^T = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]$ redak matrice X koja označava skup vrijednosti pojavljivanja termina w_iu svim dokumentima iz skupa c. Slično je $d_j = [x_{1,j}, x_{2,j}, \dots, x_{n,j}]$ stupac matrice koji označava broj pojavljivanja riječi w_iu dokumentu c_j. Skalarni produkt dvaju vektora t_i^Ti t_p daje korelaciju dvaju riječi w_i i w_p u odnosu na sve dokumente. Tada matrični produkt XX^T sadrži sve te skalarne produkte. Može se pokazati da se matrica X može razložiti na umnožak ortogonalnih matrica U, V i dijagonalne matrice Σ.



Slika 2.12. Prikaz primjene dekompozicije jedinstvenih vrijednosti trećeg reda

Vrijednosti iz matrice Σnazivamo jedinstvenim vrijednostima, dok su vektori $[t_1, t_2, \dots, t_m]$ i $[c_1, c_2, \dots, c_n]$ lijevi i desni jedinstveni vektori. Ukoliko uzmemo $k$vrijednosti iz matrica U_k, $V_k^T \Sigma_k$ dobivamo aproksimaciju matrice Xsa najmanjom mogućom greškom. Ukoliko želimo usporediti riječi w_ii $w_p$tada vektore $\widehat{t_\iota} = [u_{i1}, u_{i2}, \dots, u_{ik}]$ i $\widehat{t_p} = [u_{p1}, u_{p2}, \dots, u_{pk}]$pomnožimo sa matricom jedinstvenih vrijednosti $\Sigma_k \widehat{t_\iota}$ i $\Sigma_k \widehat{t_p}$te mjerimo njihovu međusobnu udaljenost.

Ove reprezentacije riječi i dokumenata se mogu iskoristiti u raznim zadacima za predstavljanje riječi vektorima. SVD se koristi kako bismo smanjili dimenzionalnost podataka, a pri tome zadržavajući strukturu sličnosti. LSA koristi rijetku matricu pojavljivljanja riječi[5] pri čemu riječi su retci matrice, a stupci su dokumenti ili paragrafi. Nakon izgradnje matrice radi se aproksimacija matricom nižeg ranga. Brojni su razlozi ovoj aproksimaciji. Jedan od njih je što veće matrice zahtijevaju više resursa dok je drugi razlog što matrice u obradi prirodnog jezika često znaju biti rijetke, te upravo SVD zadržava glavne značajke matrice. Brojne su primjene LSA od usporedbe dokumenata te pronalaska sličnih dokumenata na više jezika do proširivanja prostora značajki za sustave strojnog učenja.

2.3.3.2 Word2Vec

Modeli i predstavljanje riječi u rečenici uz pomoć neuronskih mreža su postali popularniji pristupi od klasičnih metoda kao što je LSA [37], [38]. Word2Vec [39] se koristi u generiranju vektorske reprezentacije riječi (engl. embeddings). Ovaj model se sastoji od dva pristupa:

- uzastopna vreća riječi (engl. Continuous Bag of Words CBOW) i
- preskok grupe znakova (engl. Skip-gram).

U oba modela prozor preddefinirane širine se pomiče duž cijelog korpusa, te se ti podaci koriste kako bi se istrenirala plitka neuronska mreža. Razlika je u tome što se pristupom CBOW trenira neuronska mreža kako bi na osnovu riječi iz okoline (*engl. context*) prepoznala određena riječ, dok Skip-gram radi obrnuti proces (na osnovi riječi pokušava predvidjeti kontekst). U oba pristupa nije bitan model, već linearne transformacije iz skupa Ωkoje su naučene u skrivenim slojevima neuronske mreže. Upravo se te vrijednosti koriste za vektorsku reprezentaciju riječi. U neuronskim modelima maksimizira se vjerojatnost riječi w_t na osnovu riječi koje se nalaze u okolini $h = \{w_{t-l}, \dots, w_{t-1}, w_t, w_{t+1}, \dots, w_{t+l}\}$ gdje je l širina konteksta. Vjerojatnost $P_\Omega(w_t|h)$ se računa na osnovu kompatibilnosti riječi $w_t$sa kontekstom h koja se računa uz pomoć $score$ funkcije, nad čime se primjenjuje softmax prijenosna funkcija.

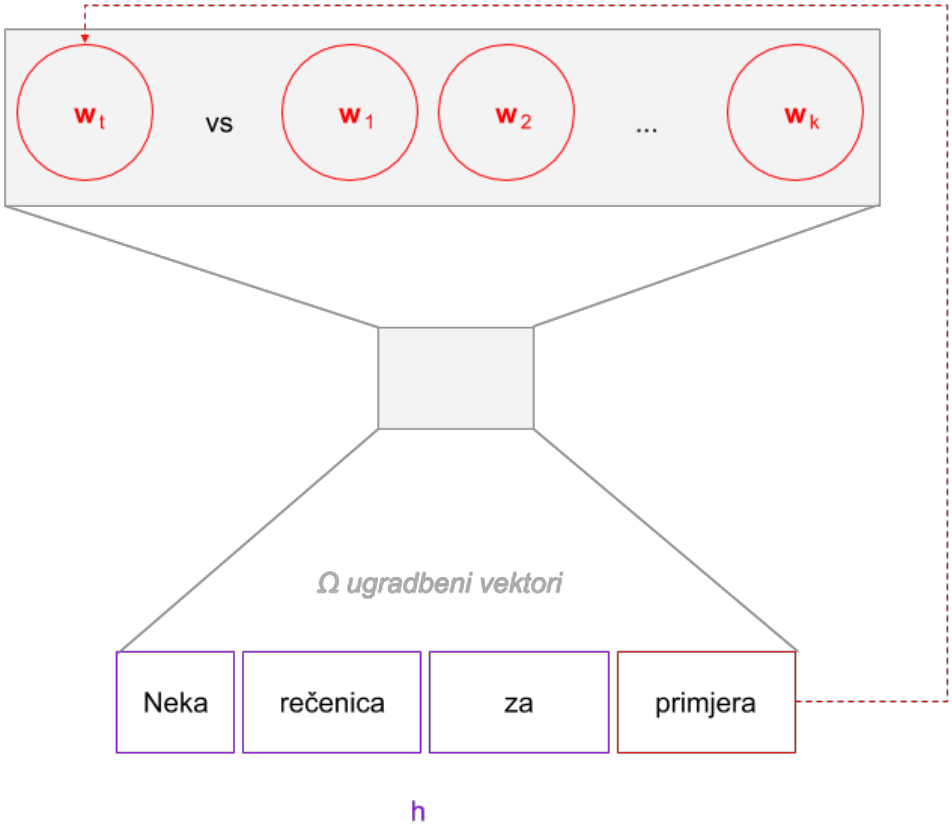
$$P(w_t|h) = softmax\big(score(w_t|h)\big) = \frac{\exp\big(score(w_t,h)\big)}{\sum_{w' \subset W} \exp\big(score(w',h)\big)} \tag{2.11}$$

Međutim ovaj proces je resursno zahtjevan jer se normalizira rezultat za svaku riječ w'iz korpusa na osnovu konteksta hi to u svakom koraku treniranja. U CBOW i Skip-gram pristupima umjesto softmax funkcije koja bi trebala pronaći riječ s maksimalnom vjerojatnosti pojavljivanja koriste se binarni klasifikatori.

[5]Tipično je to mjera tf-idf (*engl. Term frequency inverse document frequency*)

$$J_{NEG} = \log P_{\Omega}(D = 1|w_t, h) + k\, E_{w\sim V_{\Sigma}}[\log P_{\Omega}(D = 0|\mathrm{w}, h)] \tag{2.12}$$

Naziv ove funkcije gubitka je negativno uzorkovanje (*engl. negative sampling*) jer se u skupu $k\, E_{w\sim V_{\Sigma}}$generira niz riječi koje ne odgovaraju kontekstu h, a koje su označene negativnom klasom. Korištenje ovakve funkcije gubitka svodi problem aproksimacije nad određenim brojem riječi koji sami odredimo, a ne nad svim riječima u rječniku Včime se proces znatno ubrzava. Nad funkcijom gubitka J_{NEG} primjenjuje se algoritam propagacije unazad prikazan jednadžbama 2.6-2.9 i daljnji proces se svodi na minimiziranje vrijednosti funkcije gubitka podešavajući parametre Ω.



Slika 2.13. Prikaz Word2Vec CBOW metode sa negativnim uzorkovanjem

Na slici 2.13. je prikazan CBOW model Word2Vec pristupa, ulaz u neuronsku mrežu je kontekst h, a neuronska mreža se optimizira da prepozna riječ w_t koristeći negativno uzorkovanje. Ovim načinom se smanjuje računalna snaga koja je potrebna da se primjeni funkcija gubitka nad cijelim rječnikom.

2.3.3.3 GloVe

GloVe pristup za razliku od Word2Vec metode izvodi model na osnovu matrice pojavljivanja riječi (*engl. co-occurrence matrix*). Prije treniranja modela potrebno je napraviti matricu pojavljivanja X. Matrica Xse generira uz pomoć svih riječi u korpusu što čini ovaj proces jako resursno zahtjevnim zadatkom. Vrijednosti matrice $x_{i,j}$predstavljaju vjerojatnost pojavljivanja riječi w_iu kontekstu riječi w_j. Cilj pristupa je pronaći parametre neuronske mreže Ωi Btakve da dobro aproksimiraju logaritam matrice X.

$$\omega_i^T \omega_j + b_i + b_j = \log\left(X_{ij}\right) \tag{2.12}$$

Proces treniranja koristi plitku neuronsku mrežu sa jednim slojem nad matricom pojavljivanja, gdje se minimizira funkcija gubitka koja je prikazana u jednadžbi.

$$J_{Glove} = \sum_{i=1}^{V} \sum_{j=1}^{V} f\left(X_{ij}\right)\left(\omega_i^T \omega_j + b_i + b_j - \log\left(X_{ij}\right)\right)^2 \tag{2.13}$$

$$f\left(X_{i,j}\right) = \begin{cases} \left(\frac{X_{i,j}}{X_{max}}\right)^{\alpha} & if\ (X_{i,j} < X_{max}) \\ 0 & else \end{cases} \tag{2.14}$$

Funkcija fu ovoj jednadžbi pomaže riječima koje se prečesto pojavljuju zajedno da ne zakrive funkciju cilja previše. U daljnjem procesu ova funkcija cilja se minimizira korištenjem AdaGrad algoritma [40]. U suštini se koristi algoritam propagacije unazad nad funkcijom gubitka J_{Glove}, jedina razlika je u načinu ažuriranja težinskih vrijednosti.

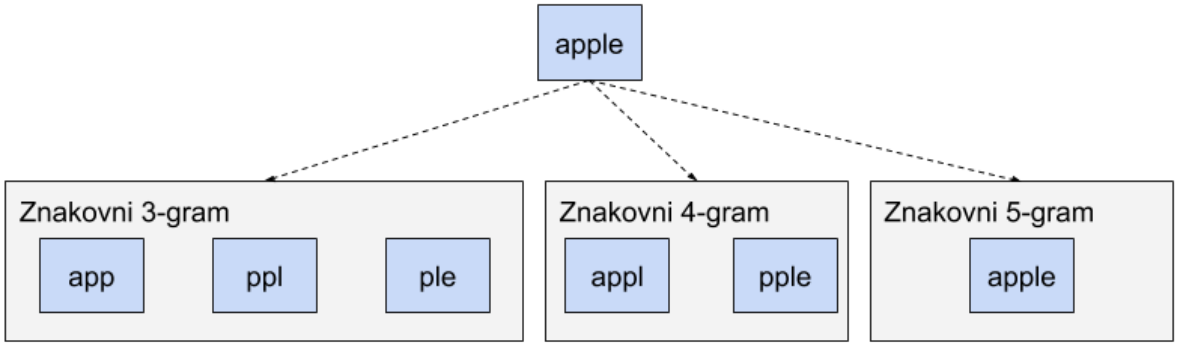
Osnovna intuicija ovog modela je da odnosi vjerojatnosti pojavljivanja riječi unutar korpusa imaju potencijal kodiranja značenja. Cilj ovog pristupa je da se vektorska reprezentacija ovih višedimenzionalnih podataka smanji na guste vektore nižih dimenzija. Ovi vektori su iskazani uz pomoć parametara Ω.

2.3.3.4 FastText

FastText model pretvaranja riječi u vektore je jedna od novijih metoda koja podržava više jezika. Osnovna razlika između FastText-a i Word2Vec pristupa je što Word2Vec tretira svaku riječ kao atomsku jedinicu te generira vektor za svaku riječ. FastText tretira svaku riječ kao sumu vektora svih ngram-a u toj riječi, pri čemu se vektori za n-grame dobivaju treniranjem nad pod nizovima riječi. U procesu treniranja se uzimaju n-grami minimalne i maksimalne dužine koje podešavamo kao hiper parametre modela. Prednosti ovog pristupa su generiranje boljih vektorskih reprezentacija za jako rijetke riječi. Također FastText modeli daju bolje rezultate za riječi koje nisu unutar rječnika.

U FastText pristupu proces treniranja može biti spor u ovisnosti o nekoliko faktora. Za razliku od Word2Vec koji obrađuje rečenice na razini riječi, FastText koristi i pod nizove znakova što

ovaj proces čini resursno zahtjevnim. Na slici su prikazani svi n-grami riječi **`apple`** minimalne dužine 3 i maksimalne dužine 5. S ovim pristupom eksponencijalno raste i broj podataka koje je potrebno obraditi, a pogotovo za manje n-grame i sa rječnikom koji uključuje više riječi.



Slika 2.14. Shematski prikaz znakovnih n-grama generiranih u FastText pristupu

FastText pristup je pokazao dobre rezultate na različitim jezicima, pogotovo morfološko bogatim jezicima. Ovaj pristup uključuje pred trenirane vektorske reprezentacije na 157 različitih jezika. Prednosti su što se ovi vektori mogu iskoristiti u treniranju neuronskih modela. Slični pristupi postoje koji koriste reprezentacije na razini znakova (*engl. char embeddings*) u kombinaciji sa dubokom neuronskom mrežom kako bi postigli bolje rezultate u raznim zadacima obrade prirodnog teksta. Ovakvi pristupi daju dobre rezultate nad zadacima označavanja riječi u rečenici i zadacima modeliranja jezika [41]. U sljedećem poglavlju dati ćemo pregled metoda temeljenih na dubokim neuronskim mrežama.

2.3.4. Metode temeljene na dubokim neuronskim mrežama

Plitke neuronske mreže daju dobre rezultate u zadacima obrade teksta kao što je dodjeljivanje govornih oznaka riječima i izgradnja vektorskih reprezentacija. Problemi nastaju kod zadataka naprednije obrade rečenica gdje pristupi na ručno definiranim značajkama daju bolje rezultate. Razlog tomu je što ove metode omogućavaju stručnjaku ugrađivanje lingvističkog znanja unutar podataka. Duboke neuronske mreže omogućavaju učenje kompleksnih struktura unutar podataka na osnovu velikog skupa podataka i niza skrivenih slojeva. Duboke neuronske mreže se mogu koristi za zadatke razumijevanja teksta i plitke semantičke obrade. Posebna vrsta duboke neuronske mreže su rekurentne neuronske mreže koje se koriste u obradi teksta. Ideja ovog koncepta je da postoji ovisnost između podataka unutar jedne rečenice koja je jako bitna za razumijevanje njenog značenja. U nastavku ćemo objasniti koje vrste dubokih neuronskih mreža se koriste u zadacima obrade prirodnog jezika.

2.3.4.1 Konvolucijska neuronska mreža

Konvolucijska neuronska mreža (*engl. convolutional neural network CNN*) [42] se sastoji od jednog ili više konvolucijskih slojeva i slojeva udruživanja (*engl. pooling layers*). Konvolucijski slojevi koriste konvolucijske operatore nad ulaznim podacima, a rezultate šalju sljedećem sloju. Iako konvolucijske neuronske mreže se najčešće koriste u obradi slike gdje su pokazale zadivljujuće rezultate [43], [44], konvolucijska neuronska mreža se može koristiti i u zadacima obrade prirodnog jezika. Struktura konvolucijskog modela pretpostavlja stroga ograničenja i poveznice u podacima. Konvolucijske neuronske mreže se koriste nad podacima gdje se tipovi uzoraka mogu pronaći na više mjesta, a pronalazak istih uzoraka u različitim podacima općenito je jako koristan koncept.

Budući da su podaci unutar zadataka obrade prirodnog jezika jednodimenzionalni i različite dužine, CNN nije najpogodnije rješenje za takve zadatke. CNN se mogu primijeniti nad jednodimenzionalnim podacima i mogu pronaći slične uzorke nad različitim podacima. Nadalje, CNN mogu postići jako dobre rezultate na zadacima obrade prirodnog jezika bez poznavanja informacija o frazama, rečenicama ili bilo kojim drugim sintaktičkim i semantičkim informacijama.

Budući da CNN pretpostavljaju jako stroge povezanosti između podataka često se koriste u procesu predprocesiranja za pronalaženje reprezentacije riječi ili rečenica. Upotrebom vektorskih reprezentacija riječi pojavila se potreba i za dobrim funkcijama koje pronalaze značajke višeg reda iz riječi. U zadacima kao što je analiza sentimenta, strojno prevođenje CNN daje dobre rezultate. CNN ima mogućnost ekstrakcije latentnih reprezentacija rečenica koje se mogu koristiti u brojnim problemima obradi prirodnog jezika.

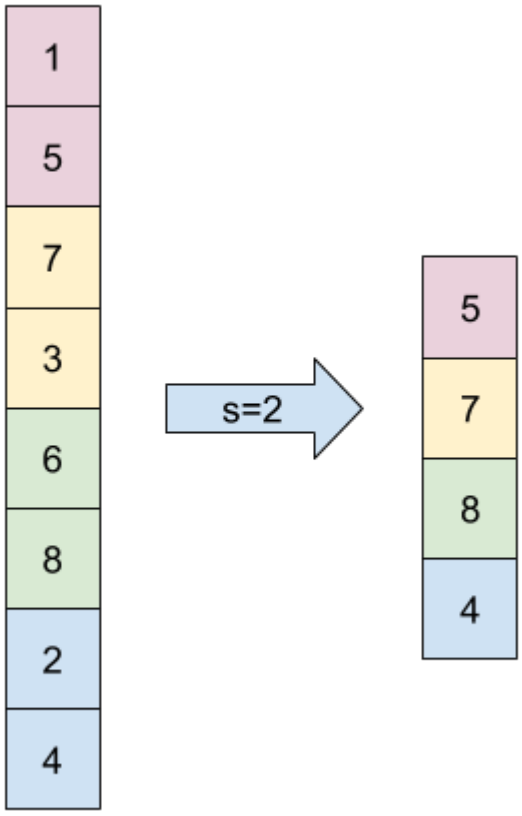
Ulazna matrica Xsu podaci vektorizirani putem BOW pristupa, ali također se često koriste nisko dimenzionalne vektorske reprezentacije. Ukoliko vektorske reprezentacije riječi složimo u rečenicu dobivamo matricu $X \in \mathbb{R}^{n \times d}$gdje je $n$broj riječi u rečenici, a $d$je dimenzija vektorske reprezentacije, tipično je to broj od 100 do 1000. U CNN konvolucijske matrice c^jse primjenjuju na ulaznu matricu X.

$$c^j = \begin{bmatrix} c^j_{1,1} & \cdots & c^j_{1,d} \\ \vdots & \ddots & \vdots \\ c^j_{j,1} & \cdots & c^j_{j,d} \end{bmatrix}, j \in \mathbb{N} \tag{2.14}$$

Koristi se pomični prozor koji uzima d kao širinu, a visina j ovisi o broju riječi koje želimo uzeti u obzir, te na taj način prolazi kroz cijelu rečenicu i generira mape značajki (*engl. feature maps*).

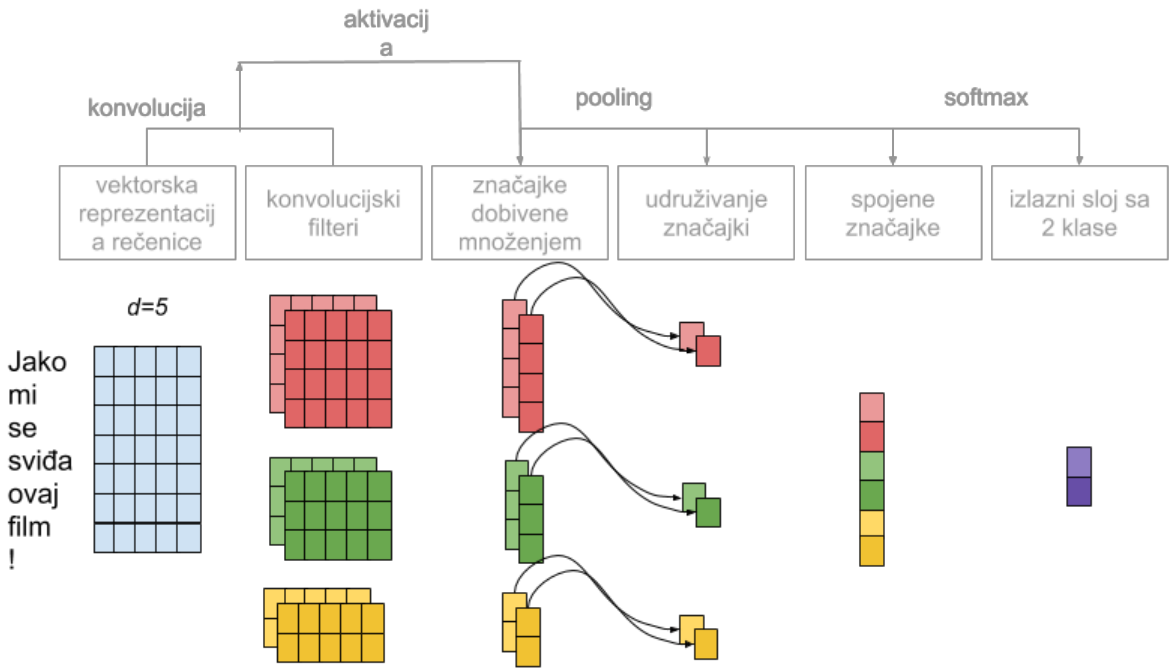
$$m_{p,a}^{j} = \begin{bmatrix} x_{p,1} & \cdots & x_{p,d} \\ \vdots & \ddots & \vdots \\ x_{p+j,1} & \cdots & x_{p+j,d} \end{bmatrix} * c_a^j, p \in [1, \ldots, n-j], j \in \mathbb{N}, a \in \mathbb{N} \qquad (2.15)$$

Mape značajki se dobivanju umnoškom i sumiranjem dijela rečenice označene podskupom matrice X i konvolucijske matrice c^j. Broj konvolucijskih matrica definira se proizvoljno i označen je sa prirodnim brojem a. Svaki od a filtera izvlači određeni uzorak $m_{p,a}^{j}$ koji se dalje procesira uz pomoć operacije udruživanja (*engl. pooling*). Tipično se koristi maksimalno udruživanje (*engl. max-pooling*). Za svaki dobiveni vektor $m_{n-p,a}^{j}$ pomičnim prozorom dimenzija s generira se vektor m'_{n-p-s} koji se dobije uzimanjem maksimalnih vrijednosti unutar prozora s kao što je prikazano na slici 2.15.



Slika 2.15. Prikaz operacije udruživanja nad vektorom dobivenim primjenom operacije konvolucije

Ovakav pristup modeliranja rečenica omogućava razvoj vektorskih semantičkih reprezentacija rečenica koje se mogu primijeniti u brojnim zadacima obrade prirodnog jezika.

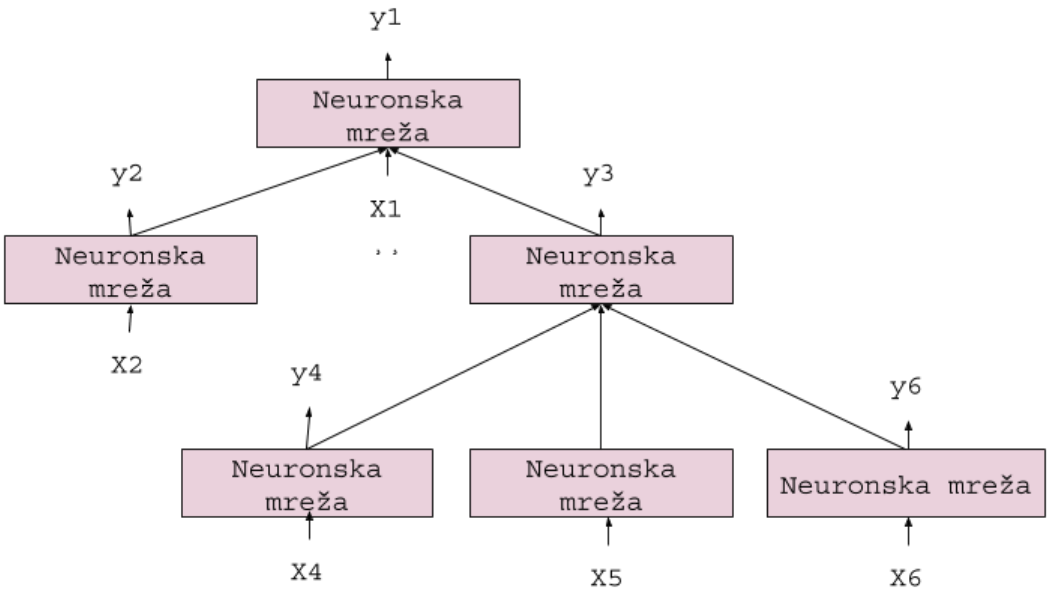


Slika 2.16. Shematski prikaz primjene konvolucijskih neuronskih mreža u obradi prirodnog jezika

Na slici 2.16. je prikazana arhitektura konvolucijske neuronske mreže, gdje je opisan cijeli proces klasifikacije rečenica u sentimentalne kategorije. Nad vektorskom reprezentacijom rečenice primjenjuje se šest različitih konvolucijskih filtera, nakon čega se radi udruživanje i spajanje značajki. Arhitektura konvolucijske neuronske mreže spojene značajke povezuje sa potpuno povezanim slojem nad kojim se primjenjuje logistička prijenosna funkcija ili softmax prijenosna funkcija.

2.3.4.2 Rekurzivne neuronske mreže

Rekurzivne neuronske mreže [45] su vrste dubokih neuronskih mreža koje se dobivaju primjenom istog skupa težinskih vrijednosti rekurzivno nad cijelom strukturom kako bi ostvarili predikcije nad podacima različite dužine. U najjednostavnijoj arhitekturi koristi se nelinearna hiperbolična tangens funkcija i težinska matrica koja je dijeljena nad cijelom neuronskom mrežom. Arhitektura rekurzivne neuronske mreže je prikazana na slici 2.17.

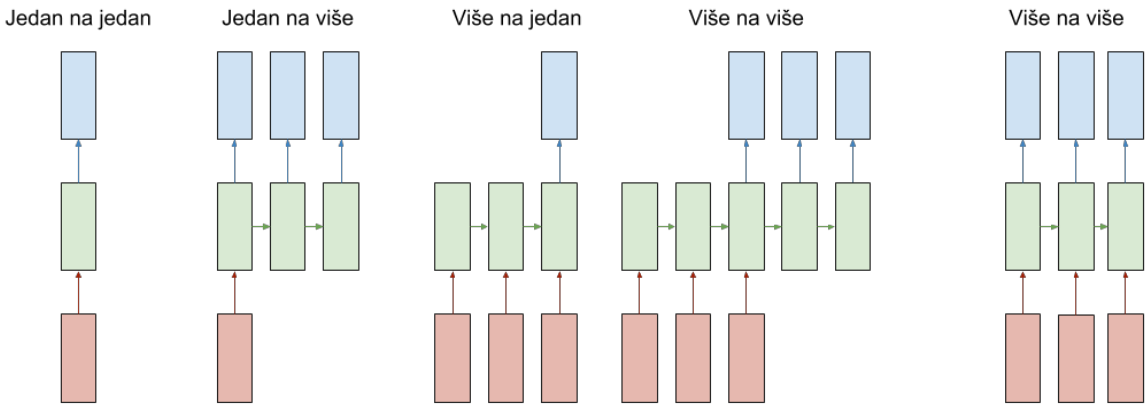


Slika 2.17. Shematski prikaz rekurzivne neuronske mreže

Prednosti rekurzivnih neuronskih mreža je što jako dobro "uče" hijerarhijske strukture i često se primjenjuju u obradi prirodnog jezika. Osnovni nedostatak je što te iste strukture moraju biti poznate tijekom procesa treniranja, te što je proces treniranja složeniji jer se struktura stabla mijenja za svaki primjerak treniranja. Rekurzivne neuronske mreže se često koriste za zadatke kao što je sintaktičko parsiranje (*engl. parse tree*).

2.3.4.3 Rekurentne neuronske mreže

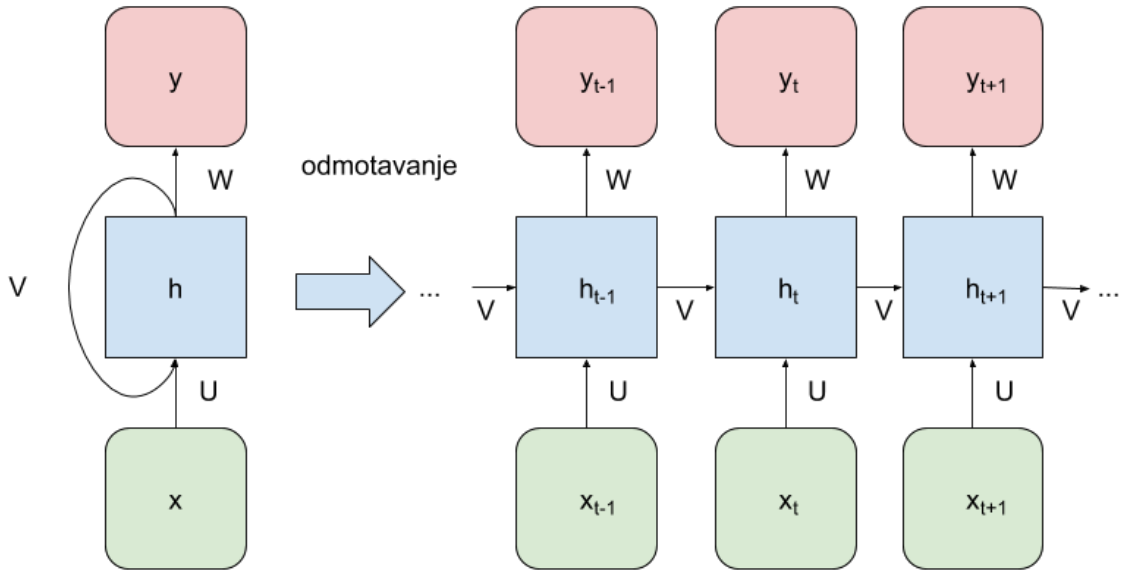
Jedan od nedostataka "običnih" i konvolucijskih neuronskih mreža je što primaju podatke fiksne veličine i proizvode izlazne podatke fiksne veličine. Rekurentna neuronska mreža (*engl. recurrent neural network RNN*) je vrsta umjetne neuronske mreže koja omogućava obradu nizova ili sekvenci podataka na ulazu i izlazu. Ovo znači da izlaz ne ovisi samo o trenutnom ulazu već i o prethodnim ulazima i izlazima što je jako čest slučaj u obradi prirodnog jezika. Rekurentne neuronske mreže se mogu "složiti" u više arhitektura kao što je prikazano na slici 2.18.



Slika 2.18. Prikaz arhitektura neuronskih mreža prema vrsti ulaznih i izlaznih podataka

Na slici 2.18. su prikazane arhitekture neuronskih mreža prema ulazu i izlazu. Jedan na jedan arhitektura se koristi u običnim i konvolucijskim neuronskim mrežama gdje se uzima jedan podatak na ulazu i dobiva se jedan podataka na izlazu. Arhitektura jedan na više se koristi gdje imamo jedan podatak na ulazu a više izlaznih podataka. Primjerice kod anotacije slike gdje imamo jednu sliku na ulazu, a više izlaznih podataka. Arhitektura više na jedan je primjerice kod zadataka analize sentimenta gdje na ulazu imamo više riječi, a na izlazu jednu klasu. Više na više arhitekture se koriste kada imamo više podataka na ulazu i njih koristimo da bismo dobili više podataka na izlazu. Primjer ove arhitekture je strojno prevođenje gdje imamo niz riječi na engleskom jeziku te želimo ih pretvoriti u niz podataka na hrvatskom jeziku.

Termin RNN se koristi za dvije klase neuronskih mreža sa sličnom strukturom, jedna sa konačnim skupom stanja, a druga sa beskonačnim skupom stanja. Obje klase neuronskih mreža predstavljaju vremenski ovisan dinamički sustav. RNN sa konačnim skupom stanja je zapravo usmjereni aciklički graf (engl. Directed acyclic graph DAG) koji se može “odmotati” te zamijeniti sa običnom neuronskom mrežom sa propagacijom unaprijed, dok se RNN sa beskonačnim skupom stanja ne može odmotati. RNN imaju dodatne spremnike za stanja s njome neuronska mreža direktno upravlja. Takva kontrolirana stanja se zovu memorija sa vratima (*engl. gated memory*). RNN sa konačnim skupom stanja prikazana je na slici 2.19.



Slika 2.19. Shematski prikaz rekurentne neuronske mreže sa konačnim skupom stanja

Kao što je vidljivo na slici težinske vrijednosti iz prethodnog koraka se koriste u sljedećem koraku jednostavnim kombiniranjem sa ulazom, što znači da stanja iz prethodnog koraka utječu na sljedeće stanje čime se u neuronsku mrežu ugrađuju prethodne ovisnosti. Na primjer u jeziku smisao rečenice određuju riječi koje čak ne moraju biti blizu.

Zbog načina na koji težinske vrijednosti primjenjuju na sljedeći korak (primjenom prijenosnih funkcija koje su u intervalu od -1 do 1 i matričnim množenjem) vrlo brzo dolazi do toga da se težinske vrijednosti smanjuju pa čak i postanu jednake 0. Ovaj problem onemogućava hvatanje dugoročnih ovisnosti jer višestrukim množenjem parametri postanu jednaki nuli. Ovaj problem se zove problem nestajućeg gradijenta (*engl. vanishing gradient problem*), te su upravo zbog njega razvijene metode kao što je duga kratkoročna memorija (engl. Long short-term memory LSTM) i vrata s ponavljajućom jedinicom (engl. Gated recurrent unit GRU).

2.3.4.4 Duga kratkoročna memorija

Duga kratkoročna memorija [46] je vrsta RNN arhitekture koja je oblikovana kako bi modelirala vremenske nizove i njihove dugoročne ovisnosti. LSTM ne koristi klasične prijenosne funkcije već implementira tzv. LSTM blokove koji se sastoje od nekoliko jedinica. Ovakvi blokovi se sastoje od troja ili četvora vrata koja kontroliraju tok informacija.

LSTM arhitektura je nastala 1997. godine te postigla jako dobre rezultate u prepoznavanju rukopisa, sintezi i prepoznavanju govora te strojnom prevođenju. Postoje brojne LSTM arhitekture, a česta arhitektura se sastoji od memorijske ćelije, ulaznih vrata, izlaznih vrata i zaboravna vrata. Sukladno tome ulazna vrata kontroliraju ulazni tok informacija, zaboravna

vrata određuju količinu informacija koja ostaje u memoriji, a izlazna vrata kontroliraju izlazni tok informacija. LSTM mreže se treniraju algoritmom vremenske propagacije unazad (*engl. Backpropagation through time BPTT*) [47].

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{2.16}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2.17}$$

$$o_i = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{2.18}$$

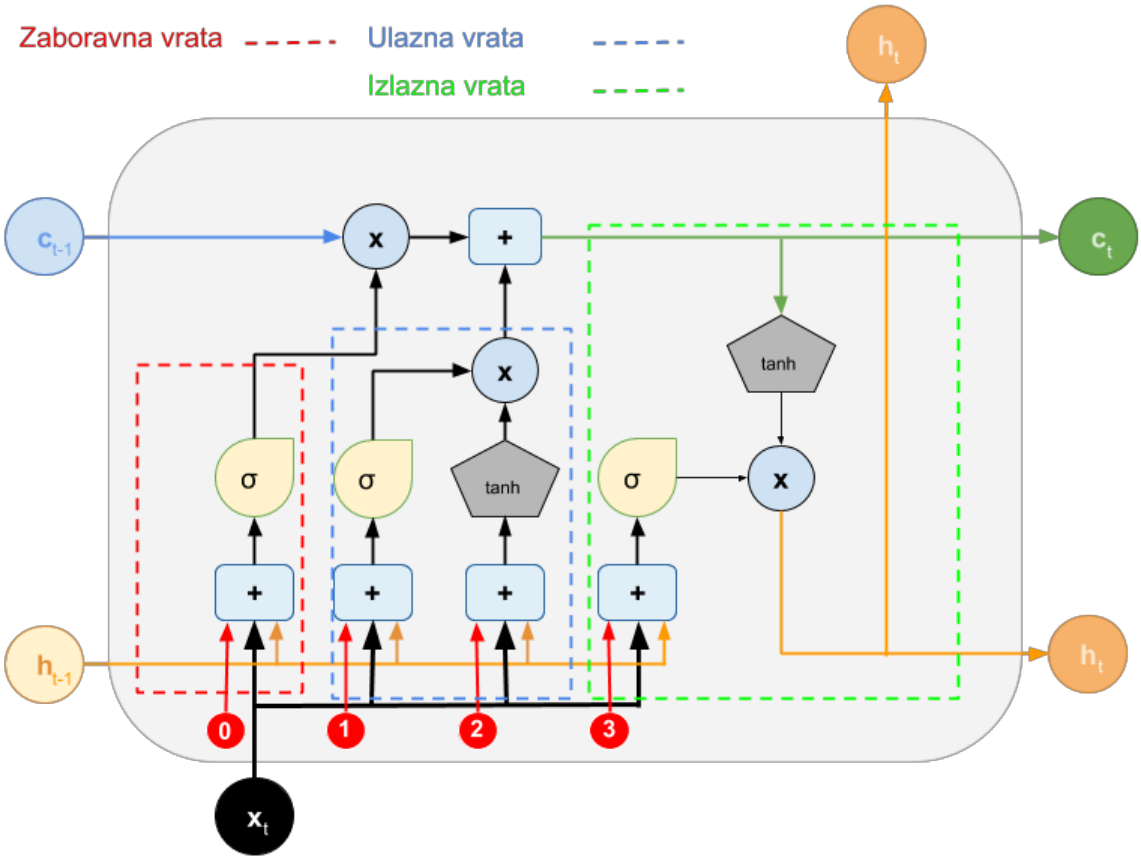
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \tag{2.19}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{2.20}$$

U jednadžbama (2.16), (2.17), (2.18), (2.19) i (2.20) su prikazani svi parametri LSTM bloka. Ove jednadžbe koriste sljedeću notaciju:
inicijalne vrijednosti c_0 i h_o jednake 0,
operator $\circ$ označava Hadamardov umnožak,
$t$označava indeks vremenskog koraka.

- $x_t \in \mathbb{R}^d$je ulazni vektor LSTM bloka,
- $f_t \in \mathbb{R}^h$prijenosni vektor vrata zaboravljanja,
- $i_t \in \mathbb{R}^h$prijenosni vektor vrata ulaza,
- $o_t \in \mathbb{R}^h$prijenosni vektor vrata izlaza,
- $h_t \in \mathbb{R}^h$je izlazni vektor LSTM bloka i
- $c_t \in \mathbb{R}^h$je vektor stanja (memorije) bloka, a
- $W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$, $b \in \mathbb{R}^{d \times h}$ su težinske matrice i vektori pristranosti koji se trebaju naučiti tijekom treniranja.

Parametri $\sigma_g - sigmoid$, $\sigma_c - \tanh$ i $\sigma_h - \tanh$ su prijenosne funkcije. Svi ovi parametri i način funkcioniranja LSTM ćelije prikazan je na slici 2.20.



Slika 2.20. Shematski prikaz LSTM bloka

Ako zanemarimo unutrašnjost bloka i gledamo samo ulaze i izlaze primijetimo da pored ulaznog vektora x_t koji predstavlja ulazni podatak u vremenu *t,* imamo h_{t-1} što je izlaz prethodne LSTM ćelije. Vrijednost c_{t-1} je memorija/stanje prethodnog bloka, sukladno tome h_t i $c_t$su izlaz i memorija/stanje trenutnog LSTM bloka. Možemo onda zaključiti da LSTM ćelija odluke donosi na osnovu prethodnog izlaza. Na grafu, plavom strelicom je prikazan memorijski ulaz koji kontrolira količinu "stare" memorije koja se prenosi na novi ulaz. Informacije na c_{t-1} se matrično množe, te ukoliko želimo uključiti "više" memorije iz prethodnog bloka tada množimo s matricom koja sadrži brojeve bližim 0, a ukoliko želimo "propustiti" više memorije tada množimo sa matricom koja sadrži brojeve bližim jedinici.

Jednadžbom (2.16). prikazujemo takozvana zaboravna vrata (*engl. forget gate*) koja primjenjuju sigmoid prijenosnu funkciju nad ulaznim podacima x_t, stanja prethodnog LSTM bloka h_{t-1}i mjere pristranosti $b_t$koje su na slici označene crvenom bojom. Jednadžbom (2.20) prikazujemo tzv. memorijska vrata koja proizvode stanje trenutne LSTM ćelije $c_t$rezultat zaboravnih vrata $f_t$se množi memorijom prethodnog LSTM bloka c_{t-1}što se dalje množi sa zbrojem ulaznih stanja i_t, x_ti stanja prethodnog LSTM bloka h_{t-1}te mjere pristranosti $b_t$na koje je primijenjena hiperbolična tangens funkcija. Naposljetku izlazna vrata izvedena su uz

pomoć izlaznih stanja o_t prikazanih jednadžbom (2.19) i memorijskog stanja $c_t$na kojem je primijenjena hiperbolična tangens funkcija.

2.3.4.5 Vrata s ponavljajućom jedinicom

Još jedan tip rekurentne neuronske mreže je vrata s ponavljajućom jedinicom (*engl. Gated recurrent unit GRU*) [48]. GRU daje jako dobre rezultate u modeliranju govornog signala i jako je slična LSTM arhitekturi, pokazuje rezultate jako slične LSTM no učinkovitiji je nad manjim skupovima podataka. GRU ima manje parametara od LSTM mreže jer ne sadrže izlazna vrata. Postoje brojne varijacije GRU jedinica. Jedna od varijacija je FGU (engl. Fully gated unit). FGU možemo prikazati sljedećim jednadžbama:

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{2.21}$$

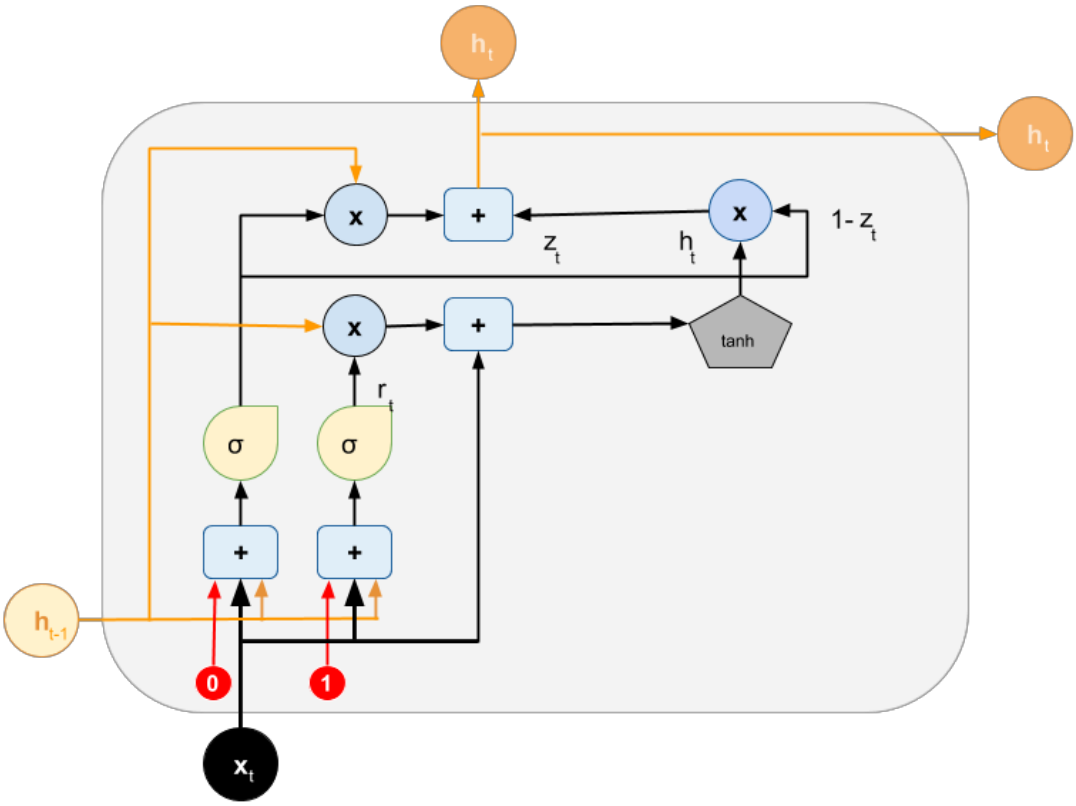
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2.22}$$

$$h_t = z_t \circ h_{t+1} + (1 - z_t) \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{2.23}$$

Parametri neuronske mreže su jednaki kao i u primjeru za LSTM.

- $x_t \in \mathbb{R}^d$je ulazni vektor GRU bloka,
- $h_t \in \mathbb{R}^h$je izlazni vektor GRU bloka,
- $z_t \in \mathbb{R}^h$je vektor vrata ažuriranja GRU bloka i
- $r_t \in \mathbb{R}^h$je vektor vrata ažuriranja GRU bloka.

Shematski prikaz GRU bloka prikazan je na slici 2.21.



Slika 2.21. Shematski prikaz GRU bloka

Preko parametara na slici u nastavku ćemo objasniti GRU blok:

- $x_t$predstavlja ulazni podatak u vremenu *t,*
- h_{t-1}je izlaz prethodne GRU ćelije,
- $h_t$je izlaz GRU bloka.

Jednadžbom 2.21. prikazujemo takozvana vrata ažuriranja (*engl. update gate*) koja primjenjuju sigmoid aktivaciju nad

- ulaznim podacima x_t,
- težinskim vrijednostima W_z,
- stanjima prethodnog GRU bloka h_{t-1}i
- mjerama pristranosti b_z.

Rezultat $z_t$množimo sa prethodnim stanjem GRU bloka h_{t-1}i zbrajamo sa umnoškom $1 - z_t$i sa primjenom hiperbolične tangens funkcije nad

- ulaznim podacima x_t,
- težinskim vrijednostima W_z,

- stanjima prethodnog GRU bloka i umnoškom h_{t-1}sa početnim stanjem r_ti mjerom pristranosti b_h.

3. AUTOMATSKO OZNAČAVANJE SEMANTIČKIH ULOGA

Brojni radovi pokazuju da se semantičke uloge mogu iskoristiti kao pomoć u raznim naprednim metodama obrade teksta. Neki od navedenih zadataka su statističko strojno prevođenje [49][50], detekcija plagijata [51], [52], te sažimanje više dokumenata [53]. Zbog toga jako je bitno da se razviju precizne metode za strojno prepoznavanje semantičkih uloga (*engl. Semantic Role Labeling*). Semantičke uloge pružaju sloj apstrakcije nad sintaktičkim ovisnostima riječi u rečenici. U ovim oznakama se kriju informacije koje su neosjetljive na sintaktičke promjene te pružaju određenu razinu semantike zbog čega se ovaj zadatak često naziva plitko semantičko označavanje (*engl. shallow semantic parsing*).

Strojno prepoznavanje semantičkih oznaka unutar rečenica se može promatrati kroz dva pristupa strojnog učenja:

Prvi pristup semantičko označavanje uloga promatra kao zadatak klasifikacije, gdje se svakoj riječi u rečenici pokušava u ovisnosti od predikata odrediti oznaka semantičke uloge. Tipično takvi pristupi koriste sintaktičke informacije gdje se kroz ručno definirane značajke (*engl. features*) pokušavaju iz označenog teksta "naučiti" semantičke uloge. Ovakvi pristupi zahtijevaju jako veliku količinu teksta označenih semantičkim ulogama te su pogodni za resursno bogate jezike (*engl. resource rich languages*). Osnovni nedostatak ovakvih modela je što su ograničeni na domenu na kojoj su trenirani.

Drugi pristupi semantičko označavanje uloga predstavljaju kao zadatak grupiranja riječi i rečenica. Semantičko označavanje uloga putem metoda nenadziranog strojnog učenja imaju par nedostataka te ne daju jednako dobre rezultate kao pristupi koji se temelje na bogatim leksičkim resursima i različitim pristupima u klasifikaciji. Neki od nedostataka nenadziranih metoda strojnog učenja je da postavljaju stroge pretpostavke nad podacima. Kao što je recimo pretpostavka da su semantički argumenti neovisni o predikatu. Nadalje za razliku od nadziranih metoda oslanjaju se na jednostavne značajke unutar rečenice. Jednostavne značajke jako puno utječu na razvoj alata koji imaju slobodniji poredak riječi od engleskog jezika. Ovo su neki od problema koje se pojavljuju prilikom strojnog prepoznavanja semantičkih uloga putem ne nadziranih metoda strojnog učenja.

Oba pristupa koriste već razvijene semantičke resurse no nenadzirane metode se mogu proširiti na i druge jezike projicirajući sintaktičke strukture na semantičke uloge.

Pored navedenih metoda u posebnu grupu nadziranih metoda možemo uvrstiti neuronske modele. Na ovom području trenutno se aktivno radi. Neuronske mreže doživjele su svoj procvat

u području obrade prirodnog teksta razvojem rekurentnih neuronskih mreža (*engl. recurrent neural networks RNN*).

3.1. Pristupi temeljeni na nadziranim metodama strojnog učenja

Semantičko označavanje uloga sastoji se od nekoliko koraka gdje se koristi nekoliko binarnih i više klasnih klasifikatora. Na slici prikazan je globalni pogled na zadatak semantičkog označavanja uloga koji se sastoji od dva zadatka. Prvi zadatak uključuje dva koraka, a to je pronalazak i određivanje smisla predikata. Drugi korak je identifikacija argumenta i klasifikacija argumenata.

Ovaj proces temeljen na značajkama uglavnom zahtjeva izvlačenje mnogih značajki iz stabla strukture rečenice (*engl. parse tree*) ili iz stabla ovisnosti riječi u rečenici. Pioniri u ovom pristupu ekstrakcije značajki i klasifikacije semantičkih uloga su Daniel Gildea i Daniel Jurafsky koji su prvi napravili alat za semantičko označavanje uloga. U svom radu [54] opisuju značajke koje se mogu kategorizirati u sljedeće kategorije:

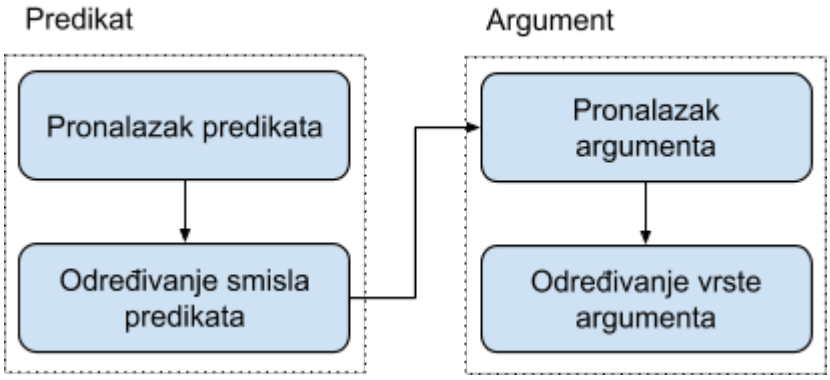
- govorne oznake riječi u rečenici (*engl. part of speech POS*)
- pozicija riječi koja se klasificira u odnosu na predikat
- sintaktička putanja do predikata iz stabla strukture rečenice
- je li rečenica napisana u pasivnom ili aktivnom obliku
- pod kategorizacijski okvir glagola [6]

Ovaj pristup je koristio FrameNet kao bazu za treniranje i testiranje i ostvario relativno dobre rezultate. FrameNet, zbog svoje namjene, se nije pokazao dobrim kao korpus za treniranje zbog čega je razvijen PropBank. Pradhan i ostali [55] usvajaju značajke od Glidea i Jurafskog na PropBank korpusu i ostvaruju jako dobre rezultate. Osnovni skup značajki je proširen oznakama imenovanih entiteta, POS oznakama glavne riječi u frazi (*engl. headword*) te brojne druge (ukupno 25 kombinacija 12 novih značajki). Klasifikacija ovih značajki izvršavala se metodom potpornih vektora (*engl Support Vector Machine SVM*). Ovaj sustav je ostvario jako dobru preciznost od 77,30% u F1 mjeri. Postignuti rezultat ovom metodom je prijavljen kao jedan od najboljih na CoNLL-2004 i CoNLL-2005 natjecanju za semantičko označavanje uloga. Zajedničke zadaće CoNLL-2004 i CoNLL-2005 odnosile su se na prepoznavanje semantičkih uloga za engleski jezik, temeljene na PropBank predikat-argument strukturi. U radu [56] heuristička pravila se koriste za eliminaciju fraza koje nisu pogodne za klasifikaciju.

[6]Pojam pod kategorizacije (*engl. subcategorization*) u lingvistici se koristi za riječi koje se često pojavljuju zajedno u određenim kontekstima i koje su međusobno ovisni

Pravila su preuzeta iz sustava za semantičko označavanje teksta opisanih u radu Xue i ostali [57]. U radu se navode sedam tipova značajki:

- je li rečenica napisana u aktivnom ili pasivnom obliku,
- govorne oznake predikata,
- pozicija argumenta u odnosu na predikat,
- putanja od argumenta do predikata na stablu govornih oznaka (engl. parse tree) i
- pod kategorizacijski okvir glagola.



Slika 3.1. Shematski prikaz semantičkog označavanja uloga

CoNLL zadatci se organiziraju godišnje za različite probleme u obradi prirodnog jezika. Postoji nekoliko zadataka koji su orijentirani na pronalazak semantičkih uloga. Prvi takav zadatak održao je se 2004 godine, a održavali su se nakon toga nekoliko puta. Svake godine zadatak je postajao teži prema zahtjevima i ograničenjima koji su bili postavljeni. Godine 2009. organiziran je prvi višejezični pristup koji je pored pored PropBank resursa, je uključivao NomBank za prepoznavanje imenskog predikata. U nastavku teksta ćemo dati pregled svih metoda, a usporediti ćemo ih prema rezultatima koji su postignuti na ovim zadacima.

3.1.1. Semantičko označavanje uloga uz pomoć metoda temeljenih na ručno definiranim značajkama

Mate-tool [58], [59] je jedan od prvih alata koji pored glagolskih predikata koristi i imenske predikate, a podržava i više jezika. Paket mate-tool označavanje semantičkih uloga provodi kroz tri različita koraka. Prvi korak je identifikacija predikata i razdvojba smisla glagola (*engl. verb sense disambiguation*). Drugi korak je identifikacija argumenata za određeni predikat, i klasifikacija argumenata. Klasifikacija i identifikacija argumenata bi se mogli promatrati kao jedan zadatak, ali zbog postizanja bolje preciznosti i finog podešavanja značajki (*engl. fine tuning*) ovaj proces je razdvojen u dvije faze. Treći korak koji se uvodi u mate-tool alat je ponovno rangiranje semantičkih argumenata. Za predikat ponovno se rangiraju argumenti

prema tome odgovaraju li smislu predikata. Identifikacija predikata i identifikacija argumenata koriste binarni klasifikator dok određivanje smisla predikata i klasifikacija argumenata koriste višeklasni klasifikator. Mate-tool ne prepoznaje samo glagolski predikat već i predikatnu imenicu.

U klasifikaciji predikata i argumenata korištena je L2-regulirana logistička regresija. Različite faze koriste različite vrste značajki. Za engleski korpus ovaj pristup koristi dva skupa značajki s ukupno 32 značajke, posebno za predikate i za argumente. Uglavnom su korištene sintaktičke informacije iz stabla ovisnosti (*engl. dependency tree*). Koriste se značajke kao što su lijevi i desni najbliži ovisnici (*engl. dependant*) argumenta i/ili predikata te lijevi i desni susjedi argumenata (elementi koji imaju istog roditelja kao i argument). Još neke od jako bitnih značajki su položaj argumenta u odnosu na predikat, okvir podkategorizacije i putanja od predikata do argumenta iz stabla ovisnosti, te sve relacije ovisnosti između riječi na tom putu. Ovaj pristup, koji se zasniva na sintaktičkim značajkama, postigao je preciznost 80.30% na CoNLL-2009 zadatku, te je donedavno bio najbolji alat za prepoznavanje semantičkih uloga.

Mate-plus [60] predstavlja nadogradnju mate-tool alata. Ovaj alat dodaje guste vektorske reprezentacije riječi u procesu klasifikacije. Mate-plus kombinira tradicionalne značajke sa modelima vektorskog prostora, te je treniran na PropBank i FrameNet leksičkim resursima. Kao što je već navedeno, tradicionalni pristupi ispitivaju povezanost riječi i njenih sintaksnih osnova u rečenici kako bi se odredila njena povezanost sa predikatom u rečenici. Nedostatak ove metode je u vektorizaciji budući da ovakve reprezentacije daju rijetke podatke (*engl. sparse*). Jedan od rješenja je korištenje distribucijske reprezentacije podataka kao što je matrica susjedstva. S takvim reprezentacijama klasifikatori daju bolje rezultate. Ukoliko samo promijenimo reprezentaciju riječi u procesu klasifikacije ne mora značiti da se toj riječi mora dodijeliti određena oznaka. Sukladno tome autori ovog alata pored vektorske reprezentacije predikata i argumenta, također uključuju i vektorske reprezentacije putanje stabla ovisnosti do predikata, okolnih riječi oko argumenta te zbroj predikata i argumenta koja se koristi kao posebna značajka. U procesu definiranja vektorske reprezentacije riječi, korišteni su vektorski modeli temeljeni na GloVe arhitekturi. Ovaj alat sa podacima iz domene postiže F1 mjeru od 86.34% a s podacima van domene na kojim je treniran F1 mjera je 81.38%.

3.1.2. Semantičko označavanje uloga uz pomoć dubokih neuronskih mreža

Za semantičko označavanje uloga mogu se koristiti neuronske mreže. Većina radova koji koriste neuronske pristupe opisuju različite arhitekture koje koriste i koji parametri daju najbolje rezultate. Većina radova opisuju sintaksno neovisne sustave koji uče semantičke uloge

iz teksta. U nastavku opisati ćemo koje vrste neuronskih mreža se koriste u zadacima automatskog označavanja semantičkih uloga i dati usporedbu rezultata za sve te pristupe.

3.1.2.1 Semantičko označavanje uloga primjenom konvolucijskih neuronskih mreža

Jedan od najutjecajnijih radova u ovom području [61] opisuje alat temeljen na neuronskoj arhitekturi pod nazivom Senna. Senna alat koristi arhitekturu neuronske mreže za zadatak semantičkog označavanja uloga. Neuronska mreža trenira se minimiziranjem logaritamske vjerojatnosti nad podacima za trening, stohastičkim podizanjem gradijenta. Svi parametri neuronske mreže prikazani su u jednadžbi 3.1, gdje je $T = S \times O$skup svih parova riječi S i skupa svih oznaka riječi O.

$$\Omega \rightarrow \sum_{(s,o)\in T} \log p(o|s,\Omega) \tag{3.1}$$

S matricom $A_{i,j}$ označavamo vjerojatnosti prijelaza iz oznake riječi t_i u oznaku riječi $t_j$za sve riječi u određenoj rečenici. Funkcija gubitka obuhvaća poticanje valjanih putanja tijekom treninga, a obeshrabrivanje svih ostalih putanja. To postiže tako da se parametri neuronske mreže $[f_\Omega]_{i_t,t}$ spoje sa matricom svih prijelaznih rezultata $[A]_{i_{t-1},i_t}$ kao što je označeno jednadžbom 3.2.

$$f\left([s]_1^T, [i]_1^T, \hat{\Omega}\right) = \sum_{t=1}^{T} [A]_{i_{t-1},i_t} + [f_\Omega]_{i_t,t} \tag{3.2}$$

Na osnovu jednadžbe se primjenjuje normalizacija preko svih mogućih putanja oznaka putem softmax metode. Primjenom ove funkcije nad parametrima neuronske mreže i istinitih vrijednosti dobiva se funkcija gubitka oznaka. Funkcija gubitka je jednaka razlici svih mogućih predviđenih putanja $[i]_1^T$i istinitih putanja $[o]_1^T$koje su prikazane jednadžbom 3.3.

$$J_{Senna} = \log p([o]_1^T \vee [s]_1^T, \Omega') = f([s]_1^T, [o]_1^T, \Omega') - logadd_{\forall [j]_i^T} f([s]_1^T, [i]_1^T, \Omega') \tag{3.3}$$

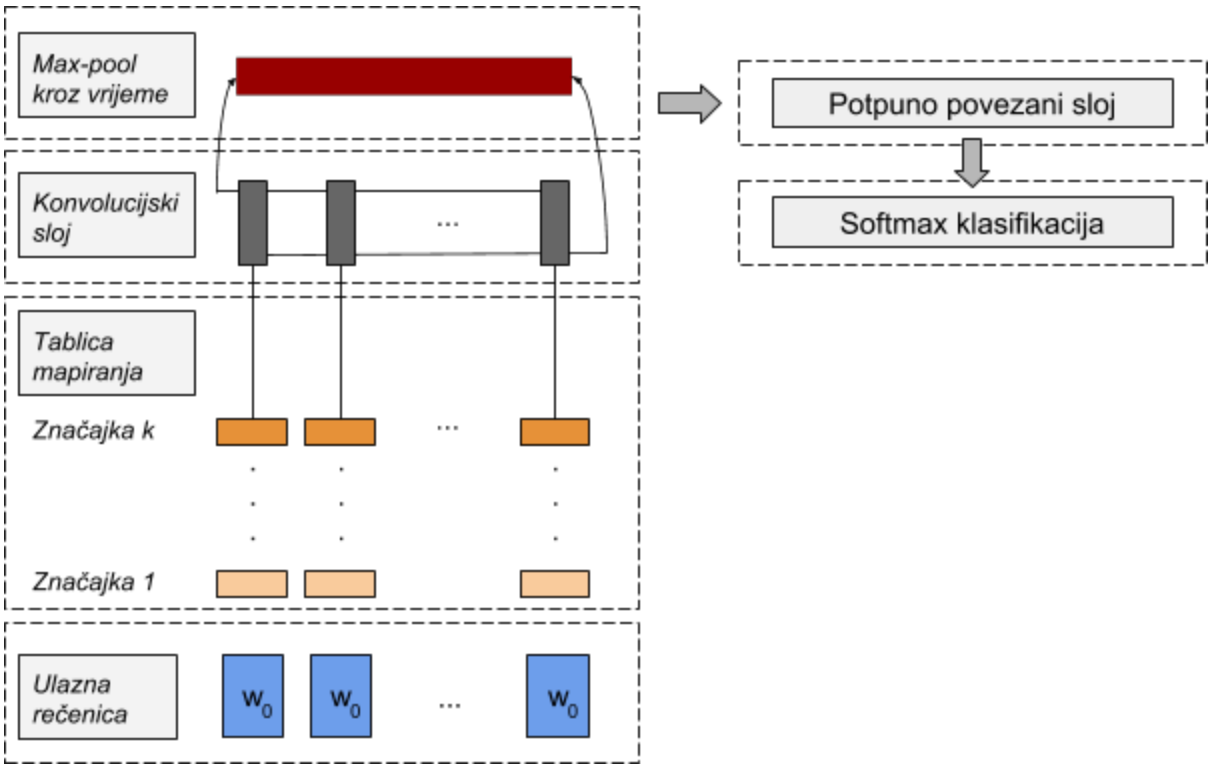
U jednadžbi sa Ω' su označeni parametri neuronske mreže koji se mogu trenirati, a dobiveni su unijom svih parametara neuronske mreže Ω i matrice $A_{i,j}$. Metodom logadd se računaju logaritamske sume svih dobivenih putanja za ulazni niz.

Sukladno tome možemo minimizirati log uvjetnu vjerojatnost iz jednadžbe 8 preko svih puteva $([s]_1^T, [o]_1^T)$. Minimizacija se izvršava putem SGD algoritma uzimajući u obzir slučajni par (s, o)te izvršavajući korak spusta prikazan u jednadžbi 3.4.

$$\Omega \leftarrow \Omega + \mu \frac{\partial J_{Senna}}{\partial \Omega} \tag{3.3}$$

μ je stopa učenja, a tijekom donošenja odluke o najboljoj mogućoj putanji koristi se Viterbi algoritam.

Alati poput Senne koriste neuronsku mrežu gotovo bez ikakvih ručno izrađenih značajki i sa što manjim predprocesiranjem teksta za ulaz u neuronsku mrežu.



Slika 3.2. Shematski prikaz CNN arhitekture za klasifikaciju rečenica

Na slici 3.2. je prikazana arhitektura Senna alata koji za klasifikaciju rečenica koristi arhitekturu konvolucijske neuronske mreže. Sirovi tekst i značajke se zajedno spajaju u matricu koja se provlači kroz konvolucijske slojeve. Nekoliko slojeva značajki automatski se izvlače iz ulaznih rečenica uz pomoć konvolucijskih slojeva nad kojim se provodi operacija udruživanja koja smanjuje dimenzionalnost podataka. Sloj udruživanja je spojen sa potpuno povezanim slojem na osnovu kojeg se izvršava softmax funkcija kako bi se izvršila klasifikacija putanja. Za izbor najbolje moguće putanje koristi se Viterbi algoritam iako autori u radovima [62], [63] navode metodu uvjetnih nasumičnih polja kao moguće bolje rješenje. Senna alat je na zadatku semantičkog označavanja uloga CoNLL 2005 postigao preciznost od 76.06%.

3.1.2.2 Semantičko označavanje uloga primjenom rekurentnih neuronskih mreža

Temeljni rad koji se bavi primjenom neuronskih mreža u modeliranju ovog problema je FitzGerald i ostali [64]. Ovaj rad opisuje rješavanje problema semantičkog označavanja uloga koji argumente kodira u dijeljeni vektorski prostor na čemu se primjenjuje neuronska mreža sa propagacijom unaprijed.

Osnovna razlika ovog pristupa je što oznake semantičkih argumenata ne koristi kao izlazne varijable već značajke riječi kodira zajedno sa njegovom oznakom. Također se trenira i binarni klasifikator koji određuje je li semantička uloga odgovara riječi koja se klasificira u kontekstu predikata.

Za generiranje svih potencijalnih argumenata koristi se niz heuristika iz prethodnog rada [65]. Autori navode da se u procesu generiranja potencijalnih argumenata koriste sintaktičke informacije iz stabla ovisnosti. Iz stabla ovisnosti nije jednostavno dobiti sve kandidate za argumente jer se nerijetko može desiti da su semantički argumenti spojeni. Rješenje ovog problema je u nizu pravila definiranih prema [65] koji sve riječi koje su povezane direktno s predikatom smatra kandidatom, s time da se uzimaju u obzir zatim i sve riječi koje su povezane tim kandidatom. Budući da semantički argumenti ne moraju biti podstabla predikata, uzimaju se i susjedni argumenti kao potencijalni kandidati. Ovaj proces se ponavlja i za lijeve i desne nasljednike roditelja predikata te također riječi povezane s njima. U radu [64] autori pokazuju da čak i plitke neuronske mreže mogu se koristiti za rješavanje problema označavanja semantičkih uloga sa jako dobrim rezultatima. Ovaj pristup na CoNLL 2012 skupu podataka postiže preciznost od 62.6% nad podacima za testiranje, a na CoNLL 2009 postiže jako dobre rezultate od čak 84.3% no u ovom zadatku nije rađena usporedba za višejezične pristupe i identifikaciju predikata.

U većini radova se koriste LSTM neuronske mreže. Zhou i Xu [66] u svom radu predlažu arhitekturu neuronske mreže koja koristi LSTM ćelije kao sustav za prepoznavanje semantičkih uloga na engleskom jeziku. Kao ulaz u neuronsku mrežu se koriste originalne riječi označene semantičkim ulogama bez ikakvih informacija o sintaksi. Ovaj pristup je pokazao da neuronske mreže mogu postići jako dobre rezultate i bez sintaktičkih informacija već može napraviti jako precizan model samo na osnovu riječi. Ovaj pristup na CoNLL 2005 zadatku postigao je preciznost od 81.07%. Pored visoke preciznosti ovaj model je jako brz u odnosu na ostale, a brzina sustava je 6,700 riječi u sekundi i daje dobre rezultate na dugim rečenicama. Prednosti ovog pristupa su u tome što se zaobilazi sintaksno predprocesiranje i parsiranje koje je u većini slučajeva glavni razlog za greške u izgradnji semantičkog stabla.
Sustavi temeljeni na klasifikaciji putem SVM algoritma [67] i ručno definiranim značajkama oslanjaju se na stručnost eksperta te ne dozvoljavaju da model sam uči na osnovu podataka. Pristup u semantičkom označavanju uloga u ovom radu koristi dvosmjerne LSTM mreže. Dvosmjerne LSTM mreže (*engl. Bidirectional LSTM DB-LSTM*) [68] sastoji se od para LSTM ćelija gdje se izlaz iz jedne LSTM ćelije uzima kao ulaz u drugu LSTM ćeliju, ali i obnuto. Istraživanja u radu pokazuju da upravo ova arhitektura je jako bitna za postizanje dobrih rezultata. Implementacija semantičkog označavanja uloga uključuje prvo procesiranje označene rečenice riječ po riječ. Dvije osnovne značajke koje se koriste su predikat i argument, a izlaz je uloga semantičkog argumenta za taj predikat. Ako rečenica ima n predikata tada će

se rečenica obraditi n puta. Pored ovih značajki uvode se i dodatne značajke kao što je kontekst predikata, koji sadrži kontekstne riječi koje se nalaze uz predikat, te oznaka regije. Oznaka regije se definira na osnovu konteksta predikata. Ukoliko je riječ koja se klasificira u kontekstu predikata, tada je njena vrijednost 1. Inače je jednaka 0. Primjer značajki za jednu rečenicu je prikazan u tablici 1.

Tablica 3-1 Primjer rečenice i značajki pri čemu se koriste "IOB" shema označavanja *za argumente*

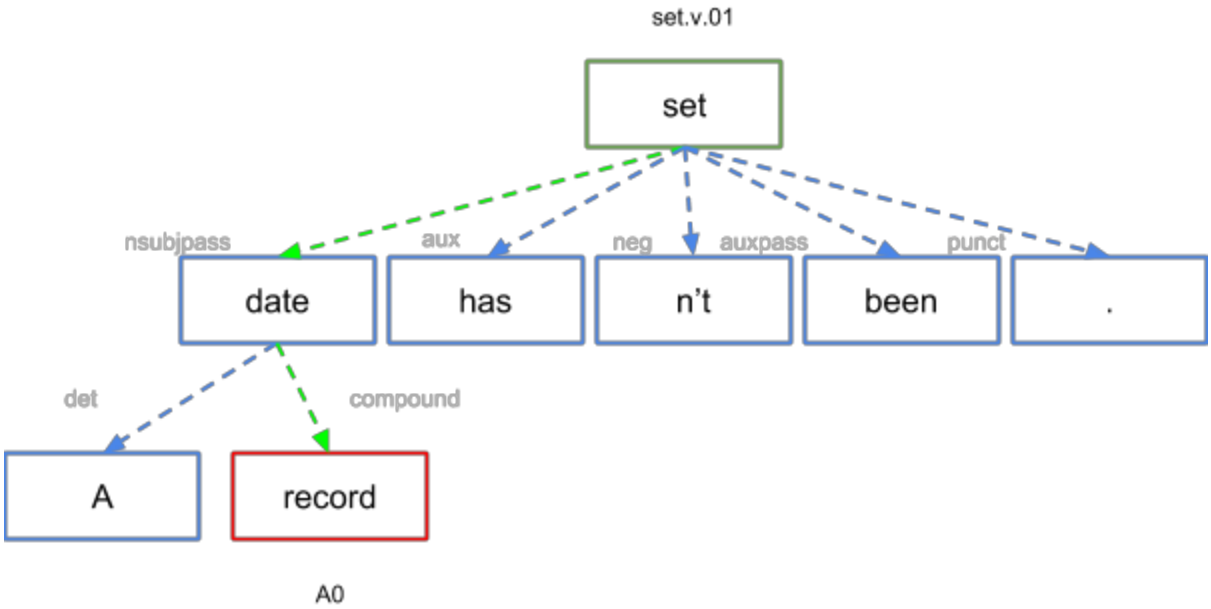
Redni broj riječi	Argument	Predikat	Kontekst	Oznaka regije	Oznaka argumenta
1	A	set	been set .	0	B-A1
2	record	set	been set .	0	I-A1
3	date	set	been set .	0	I-A1
4	has	set	been set .	0	O
5	n't	set	been set .	0	B-AM-NEG
6	been	set	been set .	1	O
7	set	set	been set .	1	B-V
8	.	set	been set .	1	O

Za rješavanje problema rijetkih podataka korišteni su neuronski modeli za reprezentaciju riječi i, kao što autor navodi u radu, sistematizacija dobrih reprezentacija za zadatak semantičkog označavanja uloga je posebna tema koju je potrebno istražiti. Četiri značajke prikazane u tablici 1 su spojene u jedan ulaz koji se dalje koristi kroz LSTM slojeve. Primjer za klasifikaciju jedne riječi je prikazan na slici 3.3.

Slika 3.3. Shematski prikaz DB-LSTM arhitekture za prepoznavanje semantičkih uloga

Zadatak semantičkog označavanja uloga duboko se oslanja na sintaktičke informacije. Upravo zbog toga zanimljivo je vidjeti kako jedan ovakav model može nadmašiti metode koje se zasnivaju na sintaktičkim značajkama.

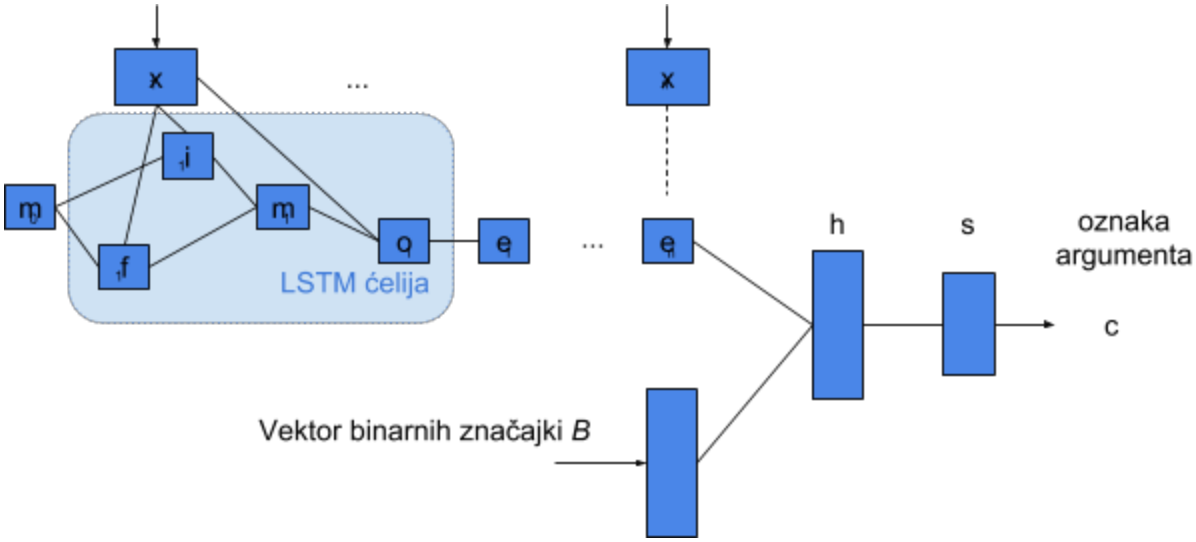
Ukoliko duboke neuronske mreže mogu postići dobre rezultate, ubacivanje sintaktičkih informacija u ovakve pristupe sigurno bi trebao postići bolje rezultate. Roth i ostali [69] predstavljaju model koji koristi jednu od najinformativnijih sintaktičkih značajki za ovaj zadatak, a to je stablo ovisnosti. Ovaj pristup koristi označene sekvence na putu od argumenta do predikata iz stabla ovisnosti kako bi izgradio vektorske reprezentacije ovih putanja. Primjer takve putanje prikazan je na slici .



Slika 3.4. Shematski prikaz stabla ovisnosti

Putanja od argumenta **`record`** do predikata **`set`** je ***`record`*** ← ***`compound`*** ← ***`date`*** ← ***`nusbjpass`*** ← ***`set`***. Jednostavan način za treniranje neuronske mreže sa putanjama stabla ovisnosti je da se svaka putanja enkodira sa one-hot vektorom, ali to nije optimalno iz više razloga. Osnovna hipoteza ovog rada je da putanje iz stabla ovisnosti koje dijele iste riječi, govorne oznake ili ovisnosti, na sličan način utječu na semantičke uloge.

Pristup koristi LSTM mrežu kako bi se naučile vektorske reprezentacije za ove putanje. Ulazne putanje se provlače kroz LSTM ćelije, te se posljednje težinske vrijednosti iz skrivenih slojeva uzimaju u procesu daljnje klasifikacije. Izlaz iz LSTM ćelija se spaja sa binarnim značajkama. Pri tome se koristi rektificirana linearna aktivacija (*engl ReLU*), te se na tim vrijednostima skrivenog sloja primjenjuje softmax klasifikacija. Izgled arhitekture neuronske mreže je prikazan na slici 3.5.



Slika 3.5. Shematski prikaz arhitekture za klasifikaciju semantičkih uloga zasnovanih na putanjama stabla ovisnosti

Nakon klasifikacije koristi se reranker struktura koja na osnovu logističkog klasifikatora predviđa cjelokupno najbolju strukturu argumenata. Najbolji rezultat dobivamo proračunom geometrijske sredine regresije i svih bodova specifičnih za argument.

PathLSTM arhitektura naznačena u ovom radu je vrednovana na CoNLL 2009 zadatku, te je postigla dobre rezultate. U zadatku semantičkog označavanja uloga postignuti na podacima iz domene za engleski jezik daje preciznost od 86.7%, a na podacima izvan domene daje preciznost od 76.1%. Ovaj model je jedan od najuspješnijih modela s obzirom na preciznost i donedavno je bio najbolji klasifikator za prepoznavanje semantičkih uloga na engleskom jeziku. Programski kod je dostupan online za slobodno korištenje i nadogradnju. Ssav programski kod je pisan u programskom jeziku Java.

Noviji pristupi koriste razne mehanizme regulizacije i inicijalizacije nad BiLSTM (engl. Bidirectional Long-Short Term Memory) neuronskom mrežom kako bi povećali preciznost. He i ostali [70] predlažu osam-slojnu arhitekturu sa ograničenim dekodiranjem i ovaj model nad CoNLL 2005 i CoNLL 2012 zadacima postiže dobru preciznost. Ovaj pristup, kako bi postigao dobru preciznost, koristi duboke BiLSTM-ove sa poveznicama (*engl. highway connections*) [70][71]. U ovom pristupu predlaže se semantičko označavanje uloga s detekcijom predikata. Sustav prvo detektira sve predikate, te nakon toga semantičke uloge za svaki predikat. Identifikacija predikata je prvi korak koji je odvojeno treniran. Proces treniranja uzima sve riječi iz rečenice. Nad tim riječima primjenjuje se jednostavna BiLSTM arhitektura s binarnom softmax funkcijom za klasifikaciju koja određuje je li riječ predikat ili ne.

Osnovni faktori koji su doveli do poboljšanja tradicionalnih pristupa za označavanje semantičkih argumenata putem BiLSTM je dodavanje poveznice između memorijskih ćelija.

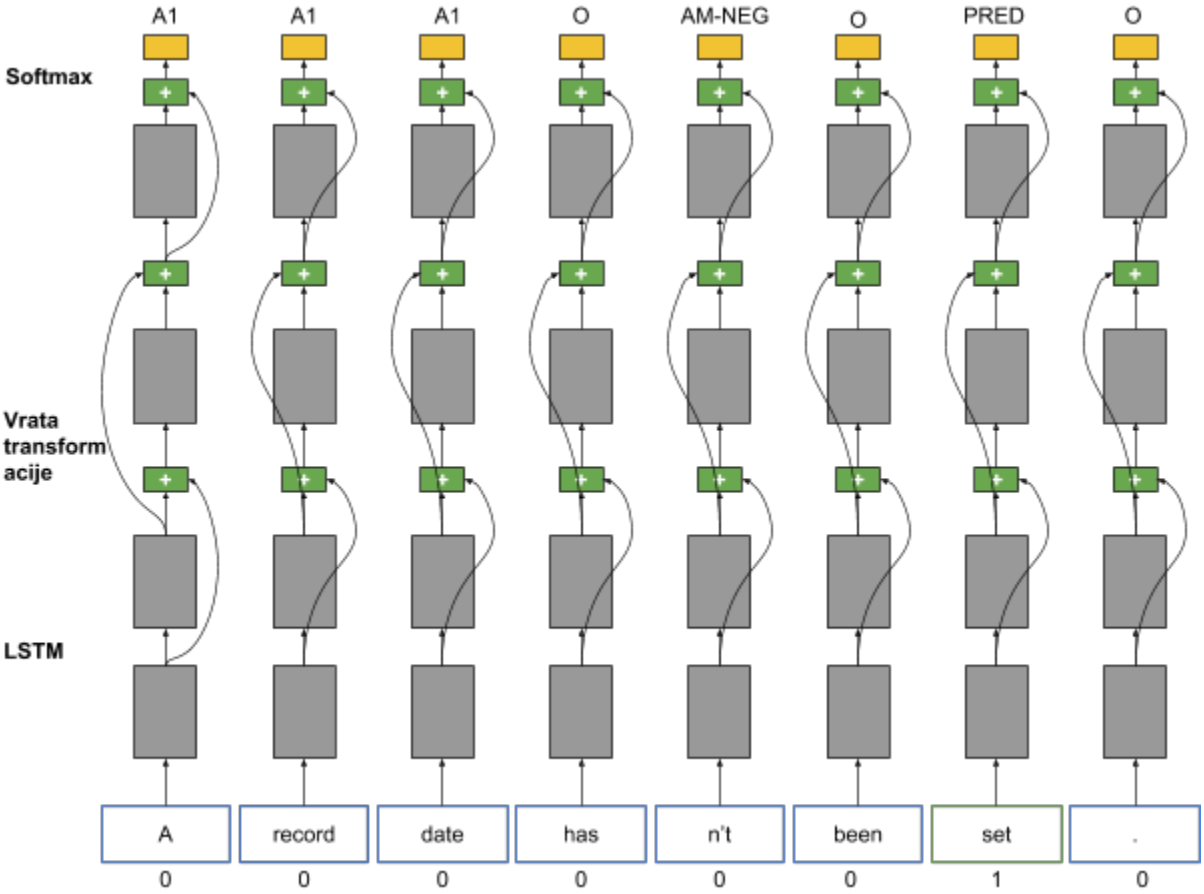
Poveznice omogućavaju lakši protok informacija kroz različite slojeve neuronske mreže bez uzrokovanja problema rastućeg ili padajućeg gradijenta (engl. vanishing or exploding gradient). Ulaz u neuronsku mrežu je vektorska reprezentacija riječi koja je spojena s binarnom značajkom koja predstavlja je li riječ predikat ili ne. Kako bi smanjili problem padajućeg gradijenta ovaj model uvodi vrata transformacije $r_t$koja se koristi između slojeva. Ova vrata spajaju izlazne vrijednosti prošlog i pretprošlog sloja LSTM ćelija. Način implementacije ove ćelije prikazan je jednadžbama :

$$r_t = \sigma_g(W_r[h_{t-1}, x_t] + b_l) \tag{3.4}$$

$$h'_t = o_t \circ \tanh(c_t) \tag{3.5}$$

$$h_t = r_t \circ h'_{l,t} + (1 - r_t) \circ W_r x_t \tag{3.6}$$

Ova vrata se smatraju kao spojnici između dvaju slojeva LSTM ćelija. Cijeli model ove arhitekture je prikazan na slici 3.6.



Slika 3.6. Shematski prikaz arhitekture BiLSTM sa poveznicama

Kako bi smanjili pretjeranu prilagođenost podacima za treniranje odbacuju se težinske vrijednosti iz skrivenih slojeva neuronske mreže. Ovaj koeficijent z se primjenjuje nad težinskim vrijednostima iz skrivenih slojeva kao što je prikazano u jednadžbama:

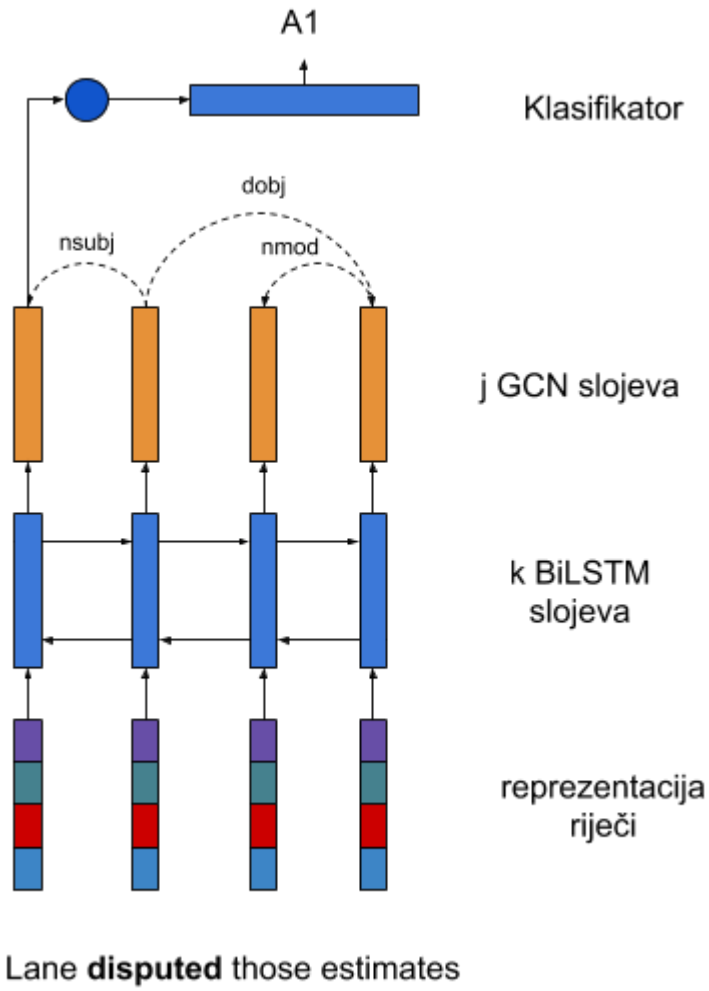
$$\mathrm{h_t} = r_t \circ h'_{l,t} + (1 - r_t) \circ W_r x_t \tag{3.7}$$

$$h_t = z \circ \mathrm{h_t} \tag{3.8}$$

Ovim pristupom ne pretpostavlja se nikakva povezanost između argumenata u procesu predviđanja oznake. Da bi se riješio ovaj problem koristi se A^* algoritam koji pretražuje i stvara označeni niz na osnovu sume svih mogućih oznaka neuronske mreže.

U radu [72] Marcheggiani i ostali predstavljaju model zasnovan na jednostavnoj BiLSTM arhitekturi. Ovaj pristup ne koristi nikakve naprednije sintaktičke značajke osim oznaka riječi koje su ugrađene unutar vektorskog prostora određenih dimenzija. Koristi se pristup gdje se iz zadnjeg sloja BiLSTM modela uzimaju skrivena stanja predikata i riječi, te na osnovu log-linearnog modela daje predikcija koji argument je najbolji. Umjesto korištenja klasične matrice koriste se nasumični vektori za leme i argumente koji su specifični za taj predikat. S ovim se očekuje da će neuronska mreža pokupiti dobre reprezentacije povezanosti koji predikati mogu očekivati koje uloge. Ovaj pristup koristi vektore dobivene primjenom vektorskih modela, gdje se koristi tehnika pod nazivom strukturirani skok n-grami (*engl. structured skip n-grams*) koja je opisana u radu [73]. Za brojne druge jezike ovaj pristup pokazao se jako dobrim te je pokazao najbolje rezultate u području. Za zadatak semantičkog označavanja CoNLL 2009 postigao je rezultat od 87.7% a na podacima van domene 77.7%

Osnovni nedostatak ovog pristupa je što ne koristi nikakve sintaktičke informacije u procesu označavanja. Ovaj nedostatak se riješio u radu [74] gdje autori za zadatak semantičkog označavanja uloga koriste se konvolucijske mreže za grafove (engl. Graph convolutional network GCN). GCN se koristi kako bi enkodirali strukture sintaktičkih ovisnosti rečenica u graf na osnovu kojih neuronska mreža uči prepoznati semantičke uloge. Ovo je svojevrsna nadogradnja koja je dodatno poboljšala ovaj proces. Nedostatak GCN-a je što nemaju mogućnost praćenja dugoročnih veza između riječi. Budući da je oko 20% argumenata u engleskom jeziku udaljeno za više od 5 riječi ovo znači jako slabu preciznost GCN-a u ovom zadatku. Taj problem je riješen na način da se riječi prvo enkodiraju u vektorski prostor uz pomoć BiLSTM. Na osnovu tih parametara primjenjuje se sloj GCN-a koji enkodira sintaktičke informacije i na kraju se koristi logaritamsko linearna klasifikacija. Cijeli ovaj proces je prikazan na slici 3.7.



Slika 3.7. Shematski prikaz arhitekture BiLSTM sa GCN slojevima

GCN arhitektura pokazala je još bolje rezultate u odnosu na prethodne gdje je na testnom skupu postignut rezultat od 89.01% što je trenutno najbolji rezultat u rezultatima na engleskom jeziku. Također ovaj model postiže najbolje rezultate na kineskom jeziku od 82.5%. U radu [75] za problem semantičkog označavanja uloga uvode se posebni mehanizmi pažnje (*engl. attention*). Pažnja sagledava rečenicu i generira oznaku na osnovu trenutne rečenice koja se klasificira. Ovaj mehanizam omogućava da se ne uzimaju sve oznake u rječniku već samo one koje odgovaraju kontekstu. U radu za određivanje odgovarajuće oznake koristi se smisao glagola prema kojem se određuje koja oznaka je prikladna za riječ i predikat koji se klasificiraju.

3.1.3. Usporedba metoda za semantičko označavanje uloga zasnovanih na nadziranim metodama strojnog učenja

U tablici prikazani su najbolji rezultati za CoNLL 2005, CoNLL 2009 i CoNLL 2012 natjecanja rezultate. Rezultati u tablicama pokazuju da u zadatku semantičkog označavanja uloga neuronski modeli danas postižu najbolje rezultate. Modeli koji uče iz teksta i ne uključuju

sintaktičke informacije daju bolje rezultate od tradicionalnih metoda zasnovanih na ručno definiranim značajkama. Također je vidljivo da je dosta poboljšanja moguće postići raznim tehnikama regulizacija kako bi se spriječila pretjerana prilagođenost ulaznom skupu za treniranje. Još jedan veliki problem je nestajući i rastući gradijent gdje je vidljivo da mehanizmi vrata puno poboljšavaju dugoročne ovisnosti između predikata i argumenata. Najbolji modeli najčešće koriste BiLSTM arhitekturu. Također poboljšanje donose i modeli koji na ove modele dodaju sintaktičke informacije. Primjer je korištenje GCN gdje se kodiraju sintaktičke informacije nad arhitekturom BiLSTM neuronske mreže.

Tablica 3-2 Rezultati različitih pristupa na CoNLL 2005 zadatku

Autor	**Metoda**	**Preciznost**	**Odziv**	**F1 mjera**	**Kombinirano[7]**
Tan[75]	Mehanizmi pažnje + BiLSTM	84.30	84.90	84.60	84.60
He[70]	BiLSTM poveznice	83.10	82.40	82.70	83.20
Zhou[66]	RNN	83.10	82.40	81.07	81.10
Punyakanok[56]	SVM + značajke	80.10	74.80	77.40	77.90
FitzGerald[64]	Neuronska mreža	81.20	76.70	78.90	-
Täckström	Dinamički algoritam + značajke	81.20	76.20	78.60	-
Toutanova[76]	CRF	-	-	78.60	-
Pradhan[55]	SVM + značajke	-	-	78.34	75.67
Koomen[77]	Logistička regresija + značajke	-	-	77.35	-
Collobert[61]	CNN	-	-	76.06	-

[7]Kombinirano kolona obuhvaća različite korpuse na kojima je testiran pristup, a označava globalnu preciznost na različitim tipovima korpusa

Tablica 3-3 Rezultati različitih pristupa na CoNLL 2009 zadatku

Autor	**Metoda**	**Preciznost**	**Odziv**	**F1 mjera**
Marcheggiani [74] (ensamble)	GCN + BiLSTM	90.50	87.70	89.10
Roth i Lapata [69] (ensamble)	LSTM	90.30	85.70	87.90
FitzGerald [64] (ensamble)	RNN	-	-	87.7
Marcheggiani [72] (globalni)	BiLSTM	-	-	-
Roth i Lapata [69] (globalni)	LSTM	90.00	85.50	87.70
FitzGerald [64] (globalni)	RNN	-	-	87.3
Marcheggiani [74] (lokalni)	GCN + BiLSTM	89.10	86.80	88.00
Marcheggiani [72] (lokalni)	BiLSTM	88.60	86.70	87.60
Roth i Lapata [78] (lokalni)	LSTM	88.10	85.30	86.70
FitzGerald [64] (lokalni)	RNN	-	-	86.70

Tablica 3-4 Rezultati različitih pristupa na CoNLL 2012 zadatku

Autor	Metoda	Preciznost	Odziv	F1 mjera
He	BiLSTM poveznice	83.50	83.20	83.40
Zhou	RNN	-	-	81.50
FitzGerald	Neuronska mreža	81.00	78.50	79.70
Täckström	Dinamički algoritam	80.5	77.80	79.70

Tablice pokazuju da u novim pristupima najbolje rezultate pokazuju neuronski modeli, ovi modeli rijetko koriste sintaktičke informacije već koriste čisti tekst enkodiran u višedimenzionalni vektorski prostor. Također mehanizmi sa vratima i raznim tehnikama regulacije mogu dodatno utjecati na preciznost ovakvih pristupa. Iako metoda koje zanimaju sintaksu daju relativno dobre rezultate, takvi pristupi obogaćeni sintaktičkim informacijama u složenijim zadacima daju bolje rezultate.

3.2. Pristupi temeljeni na polu-nadziranim i ne nadziranim metodama strojnog učenja

Resursi za metode nadziranog učenja su dostupni na isključivo engleskom jeziku. PropBank korpus pored engleskog dostupan je i na njemačkom, kineskom, katalonskom, španjolskom, korejskom i japanskom, no ovi korpusi su znatno manji od engleskog korpusa. PropBank na engleskom sadrži oko 113,000 struktura na drugim jezicima ima najviše oko 30,000 struktura. Pristupi iznad bi znatno mogli napredovati uz polu-nadziranu metodu koja će pomoći ljudima ekspertima prilikom ručnog označavanja ovakvih struktura. Ovakva metoda bi znatno ubrzala ovaj jako težak proces ručnog označavanja semantičkih argumenata. Metode koje se zasnivaju na nenadziranim metodama pokušavaju problem svesti na problem grupiranja argumenata određivanjem njihovih distribucija prema predikatu unutar određenih rečenica.

Pado i ostali [79] predlažu polu nadziranu metodu baziranu na poravnavanju rečenica (engl. sentence alignment) i ovaj problem predstavljaju kao problem optimizacije grafa. U procesu poravnavanja argumenata uključuju rečenice sa poravnatim tekstom između dva korpusa na različitim jezicima. Jedan korpus je preveden na engleski. U radu je korišten SALSA [80] korpus i engleski FrameNet korpus. U procesu povezivanja anotacija, ispravne anotacije se smatraju podgrafom potpuno povezanog asocijativnog grafa između dviju rečenica. Pronalazak semantičkih anotacija između dvaju jezika promatra kao pronalazak podskupa anotacija svih mogućih kombinacija anotacija između dvaju jezika. Ovom problemu se pristupa na način da se riječi iz rečenica spajaju prema njihovoj sličnosti. Funkcija koja ukoliko u rječniku postoji mapiranje između riječi vraća 1 kao rezultat, a ako ne postoji zapis u rječniku vraća 0. Budući da se koriste algoritmi za minimalni potpuno povezani graf, problem sličnosti se svodi na problem pronalaženja minimalnih putanja kao što je definirano jednadžbom 3.9 gdje je funkcija težine definirana prema jednadžbi 3.10.

$$A = argmin_{A \in A} \sum_{(w_s, w_t) \in A} weight(w_s, w_t) \tag{3.9}$$

$$weight(w_s, w_t) = -\log \sim (w_s, w_t) \tag{3.10}$$

Budući da se riječi u rečenicama mogu usporediti i prema frazama u kojima se nalaze, za definiranje grafa i težinskih vrijednosti mogu se koristiti i fraze. Npr. na engleskom jeziku *to be on time* se može prevesti *biti točan*. Očito je da ne postoji veza jedan naprema jedan između dvaju rečenica, stoga se posebna funkcija koristi za izračunavanje sličnosti dvaju fraza. U radu je predložen pristup koji koristi Jaccard-ov koeficijent[81]. Pronalazak maksimalnog težinskog grafa unutar asocijativnog grafa će pronaći potpuno odgovarajući graf između dvaju rečenica na stranom jeziku, nakon toga projekcija anotacija je trivijalan posao uprivanja.

Pristupi temeljeni na particiji grafa [82][83] koriste razne algoritme nad grafovima. Graf se konstruira preko predikata koji je s argumentima povezan preko težinske vrijednosti. Težinska vrijednost u grafu kvantificira prikladnost semantičkoj ulozi. Ovaj pristup je vrlo jednostavan, a ideja je da se oznake koriste kao čvorovi grafa a težinske vrijednosti između čvorova izražavaju njihovu međusobnu povezanost. Informacija o oznaci je propagirana kroz graf na način da čvorovi koji imaju slične vrijednosti imaju i slične oznake. Standardni pristup ovih metoda proces identifikacije argumenta i njihovu klasifikaciju svodi na samo problem klasifikacije argumenta. Za proces identifikacije argumenta koriste se heuristike koje se sastoje od 8 pravila postavljenih prema [84]. Za svaki glagol se nakon identifikacije argumenata napravi neusmjereni graf gdje su čvorovi instance argumenta, a težinske vrijednosti kvantificiraju sličnost između predikata i argumenta.

Za svaki graf $G = (V, E, \varphi)$ gdje je Vskup rubova, a E je skup veza, a svakoj vezi $e = (v_i, v_j)$se dodjeljuje težinska vrijednosti preko funkcije φ.Ova funkcija je simetrična i dozvoljava negativne vrijednosti. Određivanja funkcije sličnosti, koja u ovom slučaju predstavlja osnovu za particioniranje grafa, je vrlo zahtjevan zadatak. Upravo zbog toga ova funkcija zahtjeva da se u nju enkodira lingvističko znanje. Proces ugradnje lingvističkog znanja se zasniva na tri intuitivna kriterija:

- prvi kriterij je je su li čvorovi leksički slični,
- drugi argument je jesu li argumenti na sličnim pozicijama i
- zadnji kriterij je da li se čvorovi nalaze u istom okviru.

Funkcija za određivanje leksičke sličnosti zasniva se na pronalaženju kosinusne udaljenosti između vektorskih reprezentacija glavnih riječi u frazi. Sintaktička sličnost se određuje na osnovu:

- veze glavne riječi s roditeljom u stablu ovisnosti,
- je li rečenica u pasivu i aktivu te
- linearne pozicije u odnosu na glagol.

Težina se određuje podjelom ovih kriterija sa brojem 4. Usporedba i mjerenje preciznosti pristupa je presjekom grupa nastalih procesom grupiranja i grupa napravljenim iz označenog skupa podataka. Ukoliko identificirani argumenti pripadaju istom okviru, tada se pokušava izbjeći dodjeljivanje različitih argumenata istoj grupi. Proces grupiranja semantičkih uloga se izvršava na način da se radi particioniranje grafova uz pomoć algoritma kineskog šapata (*engl.*

Chinese whisper) [85]. Pojedini pristupi proces particioniranja grafa svode na problem algomerativnog grupiranja (*engl. agglomerative clustering*) na višeslojnom grafu

Drugi pristupi u ne nadziranom modeliranju semantičkih oznaka koriste Bayesove mreže za predviđanje oznake sa najvećom vjerojatnošću određenog argumenta. Autor u radu [82] navodi da je jedna od najvećih zapreka u adaptaciji semantičkih oznaka upravo nemogućnost modela da se generalizira na razini više domena. Nedostatak korpusa na različitim jezicima, pretjerana prilagođenost podataka domeni na kojima je treniran, samo su neki od problema strojnog označavanja semantičkih uloga. U ovom radu jednostavan model zasnovan na Bayesovim mrežama. Od dva koraka u procesu označavanja argumenata, proces identifikacije predikata se može riješiti uz pomoć jednostavnih heurističkih pravila [86]. Članak [87] proces označavanja uloga promatra kao proces grupiranja, gdje se koristi proces kineskog restorana (*engl. Chinese restaurant process CRP*) [88], zapravo daljinski ovisan proces kineskog restorana (*engl. distance dependant Chinese restaurant process dd-CRP*). Na osnovu prethodnih vjerojatnosti generiraju se distribucije vjerojatnosti argumenta za svaki predikat te se primjenjuje maksimalna a posteriori vjerojatnost (MAP) za svaki rezultat sličnosti minimizacijom funkcije cilja prikazane u jednadžbi 3.11.

$$argmax_{d_{i,j},i\neq i} \sum_p \sum_{k\in K_p} \log \frac{d_{k,c_{pk}}}{\sum_{k'\in K_p} d_{k,k'}} \tag{3.11}$$

Uvodi se pojam argument ključa koji se sastoji od sintaktičkih značajki. Za svaki ključ argumenta CRP određuje s kojim drugim argument ključem najbolje odgovara za određeni predikat. U jednadžbi skup $K_p$je skup svih argument ključeva, $p$je skup svih predikata a c_{pk}je argument ključ koji pripada određenoj grupi. Evaluacija modela se izvršava na PropBank korpusu, te dobiveni rezultati pokazuju jako dobru preciznost. Ovaj pristup se ne može usporediti sa ostalim nadziranim metodama jer ovaj proces daje samo grupe argumenata ali ne i oznaku semantičke uloge.

3.2.1. Usporedba metoda za semantičko označavanje uloga zasnovanih na nenadziranim metodama strojnog učenja

Klasična usporedba sa metodama nadziranog strojnog učenja nije moguća. Razlog tome je što nenadzirane metode generiraju grupiranja semantičkih argumenata koji su slični, ali ne određuje koje su zapravo njihove uloge. Proces vrednovanja se sastoji od određivanja mjere

pročišćenosti grupiranih oznaka (*engl. cluster purity PU*) i kolokacije (*engl. collocation CO*). Pročišćenost mjeri do koje granice grupirani podaci sadrže istu semantičku ulogu, dok kolokacija mjeri do koje granice argumenti s istom oznakom su pridruženi istoj grupi. Posljednja mjera je F1 oznaka koja je zapravo harmonijska sredina ovih dvaju oznaka.

Tablica 3-5 Usporedba rezultata različitih pristupa na CoNLL 2008 zadatku

Metoda	**PU**	**CO**	**F1**
Titov[87]	88.70	78.10	83.00
Woodsend	93.90	85.30	88.80
Lang[82]	80.60	81.30	80.30
Lang[83]	89.30	77.90	82.40

3.3. Implementacija komponente za semantičko označavanje uloga na hrvatskom standardnom jeziku

Proces označavanja semantičkih uloga na hrvatskom jeziku izvršili smo na korpusu prikazanom u poglavlju 2.2.6. Odabrali smo nadzirani pristup sa ručno definiranim značajkama, a u procesu klasifikacije koristili smo nasumična uvjetna polja (CRF). Značajke koje smo koristili definirane su prema [58] pokušali smo adaptirati značajke prema radu no korištenje svih definiranih značajki daju najbolje rezultate. Usporedili smo dobivene rezultate sa rezultatima dobivenim mate-tool alatom sa definiranim njemačkim značajkama (jer je od sviju koji su podržani najsličniji hrvatskom jeziku), u procesu klasificiranja korištene su značajke zasnovane na sintaktičkim informacijama. Puni skup značajki je korišten u radu. Sustav je treniran na skupu za treniranje a testiran na skupu za testiranje, te je korišten korpus na hrvatskom standardnom jeziku. U tablici su prikazani rezultati dobiveni mate-tool alatom i uvjetnim nasumičnim poljem (*engl. conditional random field CRF*) [89] pristupom. Prilikom testiranja CRF korišten je također i linearni klasifikator opisan u radu, ali CRF je za ovaj zadatak davao malo bolje rezultate. Poboljšanje u odnosu na prethodni rezultat je oko 6% iako značajke nisu bile prilagođene hrvatskom jeziku. Definiranje još kvalitetnijih značajki zahtjeva kvalitetno poznavanje hrvatskog jezika na sintaksnoj razini, ali i na semantičkoj razini, te odnos među njima. CRF je vrsta statističke metode za modeliranje koja se koristi za označavanje sekvenci podataka. Tradicionalne metode predviđaju oznaku na osnovu jednog uzorka iako ne uzimaju okolinu u obzir. CRF u obzir uzima širi kontekst. Možemo reći da je CRF pristup preteča današnjim rekurentnim neuronskim mrežama. CRF je često korišten alat u obradi

prirodnog jezika, a primjenu nalazi u označavanju dijelova riječi [90], [91], označavanju imenovanih entiteta [92] te brojnih drugih zadataka označavanja nizova. CRF je zapravo neusmjereni grafički model čiji se vrhovi mogu podijeliti u dva disjunktna skupa X i Y. U modeliranju sekvenci, graf koji nas obično zanima je usmjeren. Ulazni niz varijabli X predstavlja niz zapažanja, dok Y predstavlja skrivena stanja, tj. stanja koja se trebaju dobiti na osnovu ulaznih parametara. Uvjetna zavisnost ulaznih i izlaznih varijabli je definirana kroz niz funkcija značajki $f(i, Y_{i-1}, Y_i, X)$ ova funkcija pronalazi koje varijable na ulazu definiraju vjerojatnost svake moguće opcije pojavljivanja izlazne varijable Y_i. Svakoj značajki model dodjeljuje numeričku težinsku vrijednost te ih kombinira kako bi odredio vjerojatnosti "oznaka" za ulazne vrijednosti. Učenje ovih parametara θ se izvršava uz pomoć maksimalne vjerojatnosti za $p(Y_i \mid X_i, \theta)$. Za eksponencijalne distribucije, ovaj problem se može riješiti uz pomoć stohastičkog padajućeg gradijenta.

Tablica 3-6 Rezultati različitih pristupa klasifikacije semantičkih argumenata na hrvatskom jeziku

Identifikacija predikata									
Mate					**CRF**				
	P	R	F1	Ukup.		P	R	F1	Ukup.
Y	0.9863	0.8936	0.9377	2021	Y	0.9627	0.9698	0.9662	2021
_	0.9855	0.9983	0.9918	14614	_	0.9918	0.9918	0.9918	14614
Ukup.	0.9856	0.9856	0.9853	16635	Ukup.	0.9918	0.9918	0.9918	16635
Klasifikacija argumenta									
ACMP	0.82	0.78	0.80	23	ACMP	0.75	0.52	0.62	23
ACT	0.88	0.94	0.91	962	ACT	0.90	0.94	0.92	962
AIM	0.56	0.40	0.47	45	AIM	0.74	0.51	0.61	45
CAUS	0.25	0.09	0.14	53	CAUS	0.62	0.25	0.35	53
COND	0.57	0.62	0.59	13	COND	0.54	0.54	0.54	13
CONT	0.40	0.14	0.21	14	CONT	0.56	0.36	0.43	14
DUR	0.59	0.38	0.46	88	DUR	0.73	0.49	0.59	88

EVEN	0.83	0.58	0.68	26	EVEN	0.62	0.58	0.60	26
FREQ	1.00	0.33	0.50	15	FREQ	0.91	0.67	0.77	15
GOAL	0.52	0.30	0.38	43	GOAL	0.54	0.30	0.39	43
LOC	0.48	0.67	0.56	97	LOC	0.56	0.60	0.58	97
MANN	0.42	0.48	0.45	100	MANN	0.56	0.48	0.52	100
MEAN	0.38	0.52	0.44	21	MEAN	0.38	0.57	0.45	21
MODA	0.92	0.97	0.94	115	MODA	0.81	0.99	0.89	115
MWP	0.85	0.63	0.72	35	MWP	0.93	0.37	0.53	35
ORIG	0.81	0.55	0.65	64	ORIG	0.58	0.59	0.58	64
PAT	0.82	0.81	0.81	1052	PAT	0.82	0.81	0.82	1052
PHRA	0.17	0.08	0.11	12	PHRA	0.17	0.08	0.11	12
QUAN	0.85	0.40	0.54	43	QUAN	0.77	0.53	0.63	43
REC	0.80	0.75	0.78	117	REC	0.80	0.70	0.75	117
REG	0.52	0.36	0.43	47	REG	0.47	0.36	0.41	47
RESL	0.84	0.82	0.83	562	RESL	0.81	0.83	0.82	562
REST	0.00	0.00	0.00	7	REST	1.00	0.43	0.60	7
SOUR	0.67	0.18	0.29	11	SOUR	0.50	0.18	0.27	11
TIME	0.58	0.73	0.65	238	TIME	0.74	0.81	0.77	238
Ukup.	0.72	0.72	0.72	4062	Ukup.	0.79	0.78	0.78	3803

U tablici je prikazana usporedba mate-tool alata i vlastitog pristupa koji koristi iste značajke kao i mate-tool. Jedina je razlika u tome što je naš pristup testiran i treniran koristeći CRF. Korištene su sve značajke koje su navedene u mate-tool alatu. U mate-tool alatu koristili smo značajke za njemački jezik jer daje najbolje rezultate. CRF pristup je dao bolje rezultate u prepoznavanju predikata sa preciznošću od 99% i klasifikaciji argumenta sa preciznošću od 78%.

4. PRIMJENA SEMANTIČKOG OZNAČAVANJA ULOGA U SUSTAVIMA E-UČENJA

Obrada prirodnog jezika uključuje niz zadataka koje bi se mogli primijeniti unutar raznih sustava koji olakšavaju svakodnevni život. Klasa takvih sustava su i sustavi e-učenja. Komponenta koja određuje semantičke uloge iz teksta na prirodnom jeziku omogućila bi ovakvim sustavima niz novih funkcionalnosti koje uključuju elemente upravljanja jezikom. Takvi elementi uključuju automatsko generiranje pitanja i rečeničnih iskaza, provjere sličnosti teksta, automatsko generiranje područnog znanja iz teksta napisanog na prirodnom jeziku te brojne druge primjene. Unutar sustava e-učenja postoji podklasa sustava koji se zasnivaju na pretpostavci da učenje temeljeno na paradigmi jedan-na-jedan daje najbolje rezultate. Takvi sustavi nazivaju se inteligentni tutorski sustavi (*engl. Intelligent Tutoring Systems*), a pomoću njih se pokušavaju postići efekti poučavanja jedan na jedan. Jedan od najpoznatijih primjera primjene tehnika obrade prirodnog jezika u inteligentnim tutorskim sustavima je AutoTutor [2], [93]–[95]. AutoTutor koristi razne algoritme za obradu teksta kako bi omogućio poučavanje na prirodnom jeziku. Razni testovi primjene AutoTutora u nastavi pokazuje jako dobre rezultate. Jedini inteligentni tutorski sustav koji objedinjuje tehnike obrade prirodnog jezika na hrvatskom standardnom jeziku je CoLaB Tutor [3]. Ovaj sustav nad definiranim područnim znanjem primjenjuje razne tehnike obrade prirodnog jezika kako bi učeniku omogućio komunikaciju na prirodnom jeziku. Za ostvarivanje ovakve vrste komunikacije u fazama poučavanja i ispitivanja znanja CoLaB Tutor koristi razne leksičke resurse i niz ručno definiranih pravila.

Međutim vrijeme i cijena razvoja sadržaja za ovakav sustav je puno veća od razvoja tradicionalnog nastavnog materijala koji se danas koristi u nastavi. Ovo je najveći nedostatak primjene inteligentnih tutorskih sustava. Metodologije za ubrzavanje ovog procesa su trenutno aktivna tema istraživanja. Komponenta za semantičko označavanje uloga bi omogućila razvoj automatskog alata za razvoj nastavnog sadržaja. Sustav bi na ulazu primio nestrukturirani tekst koji bi se nizom transformacija pretvorio u sadržaj kojim sustav može lako manipulirati. Sustav bi mogao s takvim sadržajem generirati pitanja unutar podsustava za ispitivanje znanja, te generirati nastavni materijal unutar podsustava za poučavanje. Sustav bi na osnovu semantičkih relacija u rečenicama mogao odrediti semantičku sličnost u podsustavu za ispitivanje znanja. Zbog navedenog, postoji velika motivacija za izgradnjom sustava za označavanje semantičkih uloga na hrvatskom standardnom jeziku, kako bi se mogla primijeniti u inteligentnim tutorskim sustavima. Ovakva komponenta bi omogućila stvaranje slike učenikovog znanja na osnovu

unesenih odgovora na tutorska pitanja. Jedna od glavnih komponenti za automatsko generiranje pitanja je alat za označavanje semantičkih uloga, stoga bi sustav mogao uz ovo komponentu automatski generirati pitanja iz teksta napisanog na prirodnom jeziku. Brojne su druge primjene ovog alata i bitno je napraviti precizan alat za semantičko označavanje uloga kako bi se mogao primijeniti u stvarnom sustavu koji samostalno obavlja složeni proces poučavanja.

CM Tutor [96] (engl. Content Modeling Tutor) je ITS sustav koji provodi proces poučavanja bilo kojeg područnog znanja uz pomoć konceptualne mape. Na osnovu konceptualne mape upotrebom raznih algoritama se provodi proces poučavanja. Konceptualna mapa se proizvodi ručno i učitava se u sustav. Ručno definiranje konceptualne mape je zamoran posao, koji bi se mogao automatizirati upotrebom alata za prepoznavanje semantičkih uloga. Tehnike obrade prirodnog jezika se mogu primijeniti u drugim komponentama ovog tutorskog sustava kako bi se olakšala komunikacija i omogućili bolji rezultati u nastavnom procesu.

5. ZAKLJUČAK

Semantičko označavanje uloga je jedan od zahtjevnijih zadataka obrade prirodnog jezika. Složenost ovog zadatka proizilazi upravo iz toga što identifikacija semantičkih informacija zahtjeva razumijevanje teksta napisanog na prirodnom jeziku. Postavlja se logično pitanje kako predstaviti tekst u obliku razumljivom i računalu. Većina pristupa danas temelje enkodiranje riječi upravo na distribucijskoj hipotezi, te su na tom području postignuti jako dobri rezultati. Predstavljanje riječi gustim vektorskim informacijama koje u sebe enkodiraju i semantičke informacije danas je vrlo aktualna tema istraživanja. Pristupi temeljeni na Word2Vec arhitekturi su vrlo popularni, prvenstveno što ne zahtijevaju velike ručno označene leksikone. Ovaj pristup podržava i razvoj vektorskih reprezentacije na različitim jezicima, iako zbog načina na koji su modeli projektirani, najbolje rezultate daje na engleskom jeziku. Zbog potrebe razvoja vektorskih reprezentacija na drugim jezicima, pojavili su se pristupi koji u obzir uzimaju i podnizove riječi, takozvane n-grame. Ovaj pristup se zove FastText i daje obećavajuće rezultate kao jedan od multijezičnih pristupa vektorske reprezentacije riječi. Ovakve guste vektorske reprezentacije, za razliku od rijetkih podataka koji su inače korišteni u problemima obrade prirodnog jezika, daju jako dobre rezultate, jer kodiraju smisao, što je jasno vidljivo i na rezultatima najnovijih metoda za semantičko označavanje.

Najnovije metode postižu rezultate bolje od metoda pogonjenih ručno definiranim značajkama samo na osnovu riječi. Nad vektorskim reprezentacijama riječi, koje su označene semantičkim argumentima u kontekstu određenog predikata, primjenjuju se različite neuronske arhitekture. Ovakvi pristupi trenutno daju najbolje rezultate koristeći neuronske mreže koje u obzir uzimaju dugoročne ovisnosti unutar teksta. Danas se RNN ćelije zamjenjuju raznim mehanizmima vrata kako bi riješili osnovni problem kod RNN, a to su problem nestajućeg i eksplodirajućeg gradijenta. Ovi mehanizmi daju najbolje rezultate kombiniranjem različitih metoda regulacije i podešavanja hiperparametara kako bi ostvarili veću preciznost modela. Neki pristupi dodatno enkodiraju i sintaktičke informacije unutar neuronskih modela kako bi dobili još bolje rezultate. Kombiniranjem grafičkih modela (GCN) neuronskih mreža i LSTM neuronskih mreža postižu se jako dobri rezultati do čak 90% na cjelokupnim procesom označavanja imenskih i glagolskih predikata semantičkim ulogama. Iako ove metode daju jako dobre rezultate na engleskom jeziku, postavlja se pitanje koja su ograničenja ovih metoda na više jezika. GCN pristup trenutno daje najbolje rezultate na engleskom i kineskom jeziku, razlog tome je postojanje leksičkih resursa na kojima se ova metoda može trenirati.

Razvoj multijezičnog pristupa u semantičkom označavanju uloga je još uvijek područje koje je je u svojim začetcima. Postoji nekoliko metoda koji se zasnivaju uglavnom na polu-nadziranim metodama strojnog učenja koje, uz pomoć paralelnih jezičnih resursa, pokušavaju razviti ove već postojeće resurse na drugim jezicima. Jedno veliko ograničenje je nepostojanost jako velikih resursa koje se mogu iskoristiti za treniranje. Metode koje se temelje na nenadziranim metodama su jedini višejezični pristupi za semantičko označavanje uloga danas, ali i one su u pojedinim koracima ograničene na heuristike koje su specifične samo za pojedine jezike. Kao što je prikazano, postojeće metode semantičkog označavanja teksta ne daju jednako dobre rezultate na svim jezicima. Potrebno je dodatno izučiti koji set lingvističkih pravila bi se mogao primijeniti za razvoj robusne metode koja postiže dobre rezultate na hrvatskom jeziku, ali i eventualno drugim jezicima. Kao što je prikazano u poglavlju 3.2.1. jednostavnim podešavanjem argumenata mogu se dobiti rezultati koji su bolji od osnovnih rezultata na hrvatskom jeziku. Cilj daljnjeg istraživanja je definiranje neuronskog modela koji će dati bolje rezultate na morfološko bogatim jezicima u području semantičkog označavanja uloga. Kao posljedica, ovaj alat će se moći primijeniti za semantičko označavanje teksta u komponentama inteligentnih tutorskih sustava.

6. LITERATURA

[1] J. W. Rickel, "Intelligent computer-aided instruction: a survey organized around system components," *IEEE Trans. Syst. Man. Cybern.*, vol. 19, no. 1, pp. 40–57, 1989.

[2] A. C. Graesser, K. Wiemer-Hastings, P. Wiemer-Hastings, and R. Kreuz, "AutoTutor: A simulation of a human tutor," *Cogn. Syst. Res.*, vol. 1, no. 1, pp. 35–51, Dec. 1999.

[3] B. Žitko, "Model inteligentnog tutorskog sustava zasnovan na obradi kontroliranog jezika nad ontologijom." 25-Mar-2010.

[4] C. J. Fillmore, "The Case for Case," *Texas Symp. Lang. Universals*, p. 109, 1967.

[5] N. Chomsky, *Aspects of the theory of syntax*. M.I.T. Press, 1965.

[6] C. J. Fillmore, "Some Problems for Case Grammar," Ohio, 1971.

[7] Z. Glavaš, "Semantičke uloge u suvremenim lingvističkim teorijama," *Hrvatistika : studentski jezikoslovni časopis*, vol. 6., no. 6, pp. 135–148, Dec. 2012.

[8] D. Dowty, "Thematic Proto-Roles and Argument Selection," 1991.

[9] C. F. Baker, C. J. Fillmore, J. B. Lowe, C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet Project," in *Proceedings of the 36th annual meeting on Association for Computational Linguistics* -, 1998, vol. 1, p. 86.

[10] "Welcome to FrameNet! | fndrupal." [Online]. Available: https://framenet.icsi.berkeley.edu/fndrupal/. [Accessed: 27-Sep-2018].

[11] P. Kingsbury and M. Palmer, "From TreeBank to PropBank," *Proc. Int. Conf. Lang. Resour. Eval.*, pp. 1989–1993, 2002.

[12] "The Proposition Bank (PropBank)." [Online]. Available: https://propbank.github.io/. [Accessed: 27-Sep-2018].

[13] "Treebank-3 - Linguistic Data Consortium." [Online]. Available: https://catalog.ldc.upenn.edu/ldc99t42. [Accessed: 27-Sep-2018].

[14] Karin Kipper Schuler, "Verbnet: a broad-coverage, comprehensive verb lexicon," University of Pennsylvania, 2005.

[15] "Martha Palmer | Projects | Verb Net." [Online]. Available: https://verbs.colorado.edu/~mpalmer/projects/verbnet.html. [Accessed: 27-Sep-2018].

[16] B. Levin, "English Verb Classes and Alternations A Preliminary Investigation," 1993.

[17] "NomBank v 1.0 - Linguistic Data Consortium." [Online]. Available: https://catalog.ldc.upenn.edu/LDC2008T23. [Accessed: 27-Sep-2018].

[18] M. Gerber, J. Y. Chai, and A. Meyers, "The role of implicit argumentation in nominal

SRL.” 2009.

[19] “Proteus Project: NOMLEX.” [Online]. Available: https://nlp.cs.nyu.edu/nomlex/. [Accessed: 27-Sep-2018].

[20] “WordNet | A Lexical Database for English.” [Online]. Available: https://wordnet.princeton.edu/. [Accessed: 27-Sep-2018].

[21] A. Akbik and Y. Li, “POLYGLOT: Multilingual Semantic Role Labeling with Unified Labels,” *Proc. ACL-2016 Syst. Demonstr.*, pp. 1–6, 2016.

[22] S. Pado and M. Lapata, “Cross-lingual Annotation Projection for Semantic Roles,” *J. Artif. Intell. Res.*, vol. 36, pp. 307–340, Nov. 2009.

[23] “hrWaC – Croatian web corpus | Natural Language Processing group.” [Online]. Available: http://nlp.ffzg.hr/resources/corpora/hrwac/. [Accessed: 27-Sep-2018].

[24] “CROVALLEX 2.0008.” [Online]. Available: http://theta.ffzg.hr/crovallex/. [Accessed: 27-Sep-2018].

[25] N. Ljubešić, Ž. Agić, F. Klubička, V. Batanović, and T. Erjavec, “Training corpus hr500k 1.0,” *https://github.com/nljubesi/hr500k*, Apr. 2018.

[26] “MULTEXT-East Home Page.” [Online]. Available: http://nl.ijs.si/ME/. [Accessed: 27-Sep-2018].

[27] L. Dimitrova, N. Ide, V. Petkevic, T. Erjavec, H. J. Kaalep, and D. Tufis, “Multext-East,” in *Proceedings of the 17th international conference on Computational linguistics* -, 1998, vol. 1, p. 315.

[28] J. Nivre *et al.*, “Universal Dependencies 2.0 – CoNLL 2017 Shared Task Development and Test Data,” *http://universaldependencies.org/conll17/*, May 2017.

[29] “Universal Dependencies.” [Online]. Available: http://universaldependencies.org/. [Accessed: 27-Sep-2018].

[30] Franck Rosenblatt, *The Perceptron, a Perceiving and Recognizing Automaton Project Para - F. Rosenblatt - Google Knjige*. 460-461: Cornell Aeronautical Laboratory, 1957.

[31] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in …,” *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[32] C. Van Der Malsburg, “Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms,” in *Brain Theory: Proceedings of the First Trieste Meeting on Brain Theory, October 1--4, 1984*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 245–248.

[33] S. S. Haykin and Simon, *Neural networks : a comprehensive foundation*. Prentice Hall, 1999.

[34] J. Duchi JDUCHI and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” 2011.

[35] M. Baroni, G. Dinu, and G. Kruszewski, “Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 238–247.

[36] T. Landauer, P. Foltz, and D. Laham, “An Introduction to Latent Semantic Analysis,” *Discourse Process.*, no. 25, 1998.

[37] R. Collobert and J. Weston, “A unified architecture for natural language processing,” in *Proceedings of the 25th international conference on Machine learning - ICML ’08*, 2008, pp. 160–167.

[38] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” *CoRR*, pp. 1–9, 2013.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Jan. 2013.

[40] J. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[41] W. Ling *et al.*, “Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation,” Aug. 2015.

[42] Y. LeCun *et al.*, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[43] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” Nov. 2013.

[44] K. Simonyan and A. Zisserman, “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION,” 2015.

[45] C. Goller and A. Kuchler, “Learning task-dependent distributed representations by backpropagation through structure,” in *Proceedings of International Conference on Neural Networks (ICNN’96)*, vol. 1, pp. 347–352.

[46] S. Hochreiter and J. J. Urgen Schmidhuber, “Long Short-Term Memory,” 1997.

[47] M. C. Mozer, “A Focused Backpropagation Algorithm for Temporal Pattern Recognition,” *Complex Syst.*, 1989.

[48] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” Jun. 2014.

[49] W. Aziz, M. Rios, and L. Specia, “Shallow Semantic Trees for SMT,” in *Proceedings of*

WMT, 2011, pp. 316–322.

[50] D. Xiong, M. Zhang, and H. Li, "Modeling the Translation of Predicate-Argument Structure for SMT," Association for Computational Linguistics, 2012.

[51] M. Paul and S. Jamal, "An Improved SRL Based Plagiarism Detection Technique Using Sentence Ranking," *Procedia Comput. Sci.*, vol. 46, pp. 223–230, Jan. 2015.

[52] A. H. Osman, N. Salim, M. S. Binwahlan, R. Alteeb, and A. Abuobieda, "An improved plagiarism detection scheme based on semantic role labeling," *Appl. Soft Comput.*, vol. 12, no. 5, pp. 1493–1502, May 2012.

[53] A. Khan, N. Salim, and Y. Jaya Kumar, "A framework for multi-document abstractive summarization based on semantic role labelling," *Appl. Soft Comput.*, vol. 30, pp. 737–747, May 2015.

[54] D. Gildea and D. Jurafsky, "Automatic Labeling of Semantic Roles," 2002.

[55] S. Pradhan, K. Hacioglu, W. Ward, J. H. Martin, and D. Jurafsky, "Semantic role chunking combining complementary syntactic views," *Proc. Ninth Conf. Comput. Nat. Lang. Learn. - CONLL '05*, no. June, pp. 217–220, 2005.

[56] V. Punyakanok, D. Roth, and W.-T. Yih, "The Importance of Syntactic Parsing and Inference in Semantic Role Labeling," 2008.

[57] N. Xue and M. Palmer, "Calibrating Features for Semantic Role Labeling."

[58] A. Björkelund, L. Hafdell, and P. Nugues, "Multilingual Semantic Role Labeling," pp. 43–48.

[59] R. Johansson and P. Nugues, "Dependency-based Semantic Role Labeling of PropBank," 2008.

[60] M. Roth and K. Woodsend, "Composition of Word Representations Improves Semantic Role Labelling," *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process.*, no. 2002, pp. 407–413, 2014.

[61] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2461–2505, 2011.

[62] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *Proc. ACL*, pp. 655–665, 2014.

[63] Y. Kim, "Convolutional Neural Networks for Sentence Classification," pp. 1746–1751, 2014.

[64] N. Fitzgerald, O. Täckström, K. Ganchev, and D. Das, "Semantic Role Labeling with Neural Network Factors," *Emnlp*, no. September, pp. 960–970, 2015.

[65] K. Ganchev and D. Das, “Efficient Inference and Structured Learning for Semantic Role Labeling,” *Trans. ACL*, vol. 3, no. 2008, pp. 29–41, 2015.

[66] J. Zhou and W. Xu, “End-to-end learning of semantic role labeling using recurrent neural networks,” *Acl*, pp. 1127–1137, 2015.

[67] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[68] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[69] M. Roth and M. Lapata, “Neural Semantic Role Labeling with Dependency Path Embeddings,” pp. 1192–1202, 2016.

[70] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, “Deep Semantic Role Labeling: What Works and What’s Next,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 473–483.

[71] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training Very Deep Networks,” Jul. 2015.

[72] D. Marcheggiani, A. Frolov, and I. Titov, “A Simple and Accurate Syntax-Agnostic Neural Model for Dependency-based Semantic Role Labeling,” 2017.

[73] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, “Transition-Based Dependency Parsing with Stack Long Short-Term Memory,” pp. 334–343, 2015.

[74] D. Marcheggiani and I. Titov, “Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling,” Mar. 2017.

[75] Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi, “Deep Semantic Role Labeling with Self-Attention,” Dec. 2017.

[76] K. Toutanova, A. Haghighi, and C. D. Manning, “A Global Joint Model for Semantic Role Labeling,” 2008.

[77] P. Koomen, V. Punyakanok, D. Roth, and W.-T. Yih, “Generalized Inference with Multiple Semantic Role Labeling Systems,” 2005.

[78] M. Roth and M. Lapata, “Neural Semantic Role Labeling with Dependency Path Embeddings,” May 2016.

[79] S. Padó and M. Lapata, “Cross-lingual Annotation Projection of Semantic Roles,” 2009.

[80] A. Burchardt, K. Erk, A. Frank, A. Kowalski, S. Padó, and M. Pinkal, “The SALSA corpus: a German corpus resource for lexical semantics,” *Proc. 5th Int. Conf. Lang. Resour. Eval. Lr.*, 2006.

[81] P. Jaccard, “The distribution of the flora in the alpine zone,” *New Phytol.*, vol. 11, no. 2,

pp. 37–50, Feb. 1912.

[82] J. Lang and M. Lapata, “Unsupervised semantic role induction with graph partitioning,” *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2011.

[83] J. Lang and M. Lapata, “Similarity-driven semantic role induction via graph partitioning,” *Comput. Linguist.*, 2014.

[84] J. Lang and M. Lapata, “Unsupervised semantic role induction via split-merge clustering,” *Proc. 49th Annu. Meet. ...*, pp. 1117–1126, 2011.

[85] C. Biemann, “Chinese Whispers - An Efficient Graph Clustering Algorithm And Its Application To Natural Language Processing Problems.” 2006.

[86] T. Grenager and C. D. Manning, “Unsupervised discovery of a statistical verb lexicon,” *Proc. 2006 Conf. Empir. Methods Nat. Lang. Process.*, no. 1997, pp. 1–8, 2006.

[87] I. Titov and a Klementiev, “A Bayesian approach to unsupervised semantic role induction,” *Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguist. Assoc. Comput. Linguist.*, 2012.

[88] D. J. Aldous, “Exchangeability and related topics,” Springer, Berlin, Heidelberg, 1985, pp. 1–198.

[89] S. Zheng *et al.*, “Conditional Random Fields as Recurrent Neural Networks,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1529–1537.

[90] V. Krishnapriya, P. Sreesha, T. R. Harithalakshmi, T. C. Archana, and J. N. Vettath, “Design of a POS tagger using conditional random fields for Malayalam,” in *2014 First International Conference on Computational Systems and Communications (ICCSC)*, 2014, pp. 370–373.

[91] M. Silfverberg, T. Ruokolainen, K. Lindén, and M. Kurimo, “Part-of-Speech Tagging using Conditional Random Fields: Exploiting Sub-Label Dependencies for Improved Accuracy,” 2014.

[92] K. U. Senevirathne, N. S. Attanayake, A. W. M. H. Dhananjanie, W. A. S. U. Weragoda, A. Nugaliyadde, and S. Thelijjagoda, “Conditional Random Fields based Named Entity Recognition for Sinhala,” in *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, 2015, pp. 302–307.

[93] S. D’mello and A. Graesser, “AutoTutor and affective autotutor,” *ACM Trans. Interact. Intell. Syst.*, 2012.

[94] S. D’Mello *et al.*, “AutoTutor detects and responds to learners affective and cognitive states,” *IEEE Trans. Educ.*, 2008.

[95] A. C. Graesser *et al.*, “AutoTutor: A tutor with dialogue in natural language,” in

Behavior Research Methods, Instruments, and Computers, 2004.

[96] T. Volarić, D. Vasić, and E. Brajković, “Adaptive Tool for Teaching Programming Using Conceptual Maps,” 2017, pp. 335–347.

POPIS OZNAKA I KRATICA

KRATICE

SRL	semantičko označavanje uloga (engl. Semantic Role Labeling)
ITS	inteligentni tutorski sustav (engl. Intelligent Tutoring System)
ANN	umjetna neuronska mreža (engl. Artificial Neural Network)
LSTM	duga kratkoročna memorija (engl. Long Short-Term Memory)
GRU	vrata s ponavljajućom jedinicom (engl. Gated Reccurent Unit)
RNN	rekurentna neuronska mreža (engl. Reccurent Neural Network)
CNN	konvolucijska neuronska mreža (engl. Convolutional Neural Network)
CRF	uvjetna nasumična polja (engl. Conditional Random Field)
CRP	proces kineskog restorana (engl. Chinese Restaurant Proces)
dd-CRP	proces kineskog restorana ovisan o udaljenosti (engl. distance dependent Chinese Restaurant Proces)
DP	Dirichletov proces (engl. Dirichlet Proces)
POS	govorna oznaka riječi (engl. Part of Speech)
MAP	maksimalna aposteriori vjerojatnost (engl. Maximal Aposteriori Probability)
BiLSTM	dvosmjerna duga kratkoročna memorija (engl. Bidirectional Long Short-Term Memory)

MATEMATIČKE OZNAKE

x	vrijednost
{}	skup
[]	vektor
$[x]^T$	stupac matrice
$\hat{X}$	dijagonalna matrica
$\mathbb{R}$	skup realnih brojeva
$\mathbb{N}$	skup prirodnih brojeva
$\circ$	Hadamardov umnožak
$argmax$	vrijednosti maksimalne vjerojatnosti
f	funkcija
σ	prijenosna funkcija
Ω	težinske vrijednosti neuronske mreže
B	vrijednosti pristranosti unutar neuronske mreže

SAŽETAK

U ovom radu kvalifikacijskom radu prikazana je analiza i primjena metoda automatskog označavanja semantičkih uloga unutar teksta. Analizirani su pristupi na raznim jezicima, najnovije metode i rezultati koji su prikazani uglavnom su orijentirani oko engleskog jezika. Semantičko označavanje uloga je trenutno aktualno područje istraživanja i posebno je zanimljivo jer se bavi značenjem. Brojni su sustavi koji koriste semantičko označavanje uloga za strukturiranje teksta. Alati za automatsko odgovaranje na pitanja, strojno prevođenje, sumarizacije teksta i sustavi za ekstrakciju informacija su samo neki od sustava za obradu teksta koji koriste ovu komponentu.

Sustavi za obradu semantičkih uloga doživjeli su znatna poboljšanja pojavom neuronskih mreža stoga smo u analizi metoda poseban naglasak stavili na arhitekture neuronskih mreža i njihove primjene u obradi teksta. Prvo poglavlje ovog kvalifikacijskog ispita uključuje teorijsku osnovicu semantičkih uloga i pregled leksičkih resursa koji su potrebni za semantičku obradu teksta. Leksički resursi su osnovica za sve pristupe automatske obrade teksta, stoga samo u posebnom poglavlju analizirali metode koje na osnovu ovih resursa izvršavaju automatsku obradu. U fokusu su metode koje uče na osnovu označenih resursa. U istom poglavlju smo pružili pregled trenutno aktualnih tema u obradi prirodnog jezika, a to su modeliranje jezika i primjena dubokih neuronskih mreža.

U primjenskom dijelu pružili smo pregled metoda obrade semantičkih uloga na engleskom jeziku. Bitno je naglasiti da pojedini pristupi imaju podršku na više jezika te se jednostavno mogu primijeniti i na druge jezike ukoliko postoje resursi na kojemu mogu biti trenirani. U ovom poglavlju kronološki su nabrojane metode označavanja semantičkih uloga i arhitekture koje postižu najbolje rezultate. Posebno su obrađene metode ne nadziranog strojnog učenja koji ovaj problem promatraju kao problem klasifikacije. Ovakve metode su popularne u jezicima gdje ne postoje bogati skupovi ručno označenih podataka.

U implementacijskom dijelu ovog klasifikacijskog ispita smo prikazali rezultate dobivene izgradnjom sustava za semantičko označavanje uloga na hrvatskom standardnom jeziku. Jedan od sustava koji bi mogao jako puno profitirati ugradnjom komponente za semantičko označavanje su inteligentni tutorski sustavi. Stoga u zadnjem poglavlju dajemo moguću primjenu ovog alata unutar podsustava tutorskih sustava temeljenih na prirodnom jeziku.