TakeLab

Laboratorij za analizu teksta i inženjerstvo znanja
Text Analysis and Knowledge Engineering Lab
Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3793

Označavanje vrste riječi u hrvatskome jeziku modelom uvjetnih slučajnih polja

Vjeran Crnjak

Zagreb, lipanj 2014.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA ZAVRŠNI RAD MODULA

Zagreb, 13. ožujka 2014.

ZAVRŠNI ZADATAK br. 3793

Pristupnik: **Vjeran Crnjak**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Označavanje vrste riječi u hrvatskome jeziku modelom uvjetnih slučajnih polja**

Opis zadatka:

Označavanje vrste riječi jedan je od osnovnih zadataka u obradi prirodnog jezika i preduvjet za mnoge druge zadatke. Uobičajeno se za označavanje vrste riječi koriste probabilistički modeli strojnog učenja za označavanje slijedova. Posebice se uspješnim pokazao model uvjetnih slučajnih polja (engl. Conditional Random Field, CRF). Međutim, za visokoflektivne jezike poput hrvatskoga označavanje vrste riječi i dalje je izazovan problem.

U okviru završnoga rada potrebno je proučiti postupke za označavanje vrste riječi temeljene na strojnom učenju s naglaskom na postupke temeljene na probabilističkim modelima. Proučiti model uvjetnih slučajnih polja (CRF) i njegovo proširenje, model uvjetnih slučajnih polja s domenski-ovisnim ograničenjima (CCRF), predložen u (Waszczuk, 2012). Razraditi postupak označavanje vrste riječi u tekstovima na hrvatskome jeziku temeljen na modelu CCRF. Razviti programsku implementaciju postupka, po potrebi se oslanjajući na postojeća rješenja. Provesti iscrpno eksperimentalno vrednovanje modela na ispitnim skupovima podataka te analizu pogrešaka. Radu priložiti izvorni i izvršni kod razvijenog sustava, skupove podataka i programsku dokumentaciju te citirati korištenu literaturu.

Zadatak uručen pristupniku: 14. ožujka 2014.
Rok za predaju rada: 13. lipnja 2014.

Mentor:

Doc. dr.sc. Jan Šnajder

Djelovođa:

Doc. dr.sc. Tomislav Hrkać

Predsjednik odbora za završni rad modula:

Prof. dr.sc. Siniša Srbljić

Hvala Jakubu Waszczuku i mentoru Janu Šnajderu na trudu i pomoći.

SADRŽAJ

Popis slika **vi**

Popis tablica **vii**

1. Uvod **1**

2. Kratak pregled pristupa **3**

2.1. Označivači temeljeni na pravilima 3

2.1.1. Brillov označivač 3

2.1.2. Označivači za visokoflektivne jezike 4

2.2. Stohastičke metode 5

2.2.1. Generativni grafički modeli 6

2.2.2. Diskriminativni grafički modeli 11

3. Uvjetna slučajna polja **14**

3.1. Pristranost oznakama – „label bias” 14

3.2. Linearni lanac 15

3.2.1. Definicija 16

3.2.2. Zaključivanje 17

3.2.3. Ocjena parametara 19

3.3. Uvjetna slučajna polja s ograničenjima 22

3.3.1. Definicija 22

3.3.2. Morfosintaktičko pogađanje 23

3.3.3. Morfosintaktičko razrješavanje višeznačnosti 24

3.3.4. Rezultati 25

4. Označivač za hrvatski jezik **26**

4.1. Implementacija 26

4.1.1. Morfosintaktički analizator 26

4.1.2. Prilagodba uvjetnih slučajnih polja s ograničenjima 28
4.2. Vrednovanje uspješnosti 28
4.2.1. Korpus 28
4.2.2. Korištene mjere uspješnosti 29
4.2.3. Analiza 30
4.2.4. Ostvarena uspješnost 32

5. Zaključak **35**

Literatura **36**

A. Definiran skup oznaka **39**

B. Raspored po slojevima **41**

POPIS SLIKA

2.1. Prikaz predložaka grafičkih modela. 6
2.2. Faktorizacija modela naivnog Bayesovog klasifikatora na graf. 8
2.3. Rešetka usmjerenog acikličkog grafa za riječi x_i uz oznake y_j. 9
2.4. Faktorizacija modela logističke regresije na graf. 12

3.1. Prikaz problema pristranosti oznaka. 15
3.2. Prikaz L_1 i L_2 regularizacijskih funkcija 21

4.1. Prikaz raspodjele višeznačnosti različitih izvora. 30

POPIS TABLICA

4.1. Prosječne veličine skupa mogućih oznaka 31
4.2. Realistični rezultati uspješnosti 32
4.3. Optimistični rezultati uspješnosti 32

1. Uvod

Označavanje vrste riječi postupak je određivanja oznake za danu riječ u tekstu (korpusu) dajući joj odgovarajuću vrstu riječi, obraćajući pažnju na definiciju i kontekst – povezanost s riječima u frazi, rečenici ili paragrafu. Najjednostavniji oblik označavanja vrste riječi je upravo onaj kojeg učimo u osnovnim školama: identifikacija je li riječ imenica, glagol, pridjev, zamjenica itd.

Nekad ručno rađen postupak danas se, u kontekstu računarske lingvistike, vrši uz pomoć algoritama koji pokušavaju diskretne izraze (riječi, fraze i sl.) povezati s opisnim oznakama.

Smatra se da je zadatak označavanja vrste riječi rečenica u engleskom jeziku poprilično zatvoren problem. Razlog tome jest vrlo malo poboljšanje u rasponu od desetak godina, od Brants (2000) do Søgaard (2011) poboljšanje od samo 1.04% gdje je trenutni najjači rezultat došao u raspon ljudske pogreške od 97.50%.[1] Za jezike s bogatijom morfologijom i slobodnim redoslijedom riječi u rečenici (poput hrvatskog i srpskog) to nije slučaj.

Hrvatski je morfološki složen jezik. Morfološka analiza riječi sastoji se od određivanja vrijednosti velikog broja obilježja kao što su tip riječi, lice, broj, rod, padež itd. Najnovija verzija[2] standarda MULTEXT-East (Erjavec, 2004) dopušta za hrvatski jezik oko 1200 različitih morfosintaktičkih opisnika (engl. *morphosyntactic descriptors* – MSD), od kojih je oko 660 u označenom korpusu SETimes.HR. Usporedivši broj oznaka s oznakama u korpusu Brown (Francis i Kucera, 1964) – 87, problem određivanja vrste riječi više nije tako jednostavan. Složenost algoritama koji se koriste za ovaj problem može rasti, ovisno o modelu, eksponencijalno u broju oznaka te je za visokoflektivne jezike, poput hrvatskog, potrebno razmatrati drugačije pristupe od uobičajenih. U ovom radu uz pregled osnovnih pristupa s naglaskom na stohastičke metode detaljnije će se proučiti primjena uvjetnih slučajnih polja i njihova nadogradnja opisana za poljski jezik u (Waszczuk, 2012).

[1] `aclweb.org/aclwiki/index.php?title=POS_Tagging_(State_of_the_art)`

[2] U trenutku pisanja ovog rada najnovija je verzija 5.

U ovome radu opisana je primjena navedenog modela za označavanje vrste riječi za hrvatski jezik i analiza uspješnosti. Ostvareni rezultati razvijenog označivača su obećavajući, a iznose oko 95% za POS-oznake i oko 81% za MSD.

U drugom poglavlju dan je kratak pregled pristupa označavanju vrste riječi s naglaskom na stohastičke metode.

U trećem poglavlju opisan je model uvjetnih slučajnih polja, postupci učenja i ocjene parametara te nadogradnja na model uvjetnih slučajnih polja s ograničenjima.

U četvrtom poglavlju opisana je implementacija označivača za hrvatski jezik i opis provedenog vrednovanja uspješnosti označivača, a i priložene su izmjerene veličine i dana usporedba s dosadašnjim rješenjima za hrvatski jezik.

2. Kratak pregled pristupa

2.1. Označivači temeljeni na pravilima

Označavanje vrste riječi koristeći pravila najstariji je pristup označavanja u kojem se koriste ili ručno izgrađena pravila – proizvod lingvističke intuicije – ili automatski postupci strojnog učenja za njihovo izgrađivanje. Moguće je koristiti pravila i za razrješavanje višeznačnosti, ako odmah na početku možemo ponuditi skup oznaka za određenu riječ. Razrješavanje se postiže analizom lingvističkih svojstava riječi uzimajući kontekst u obzir.

2.1.1. Brillov označivač

Pristup s ručno izgrađenim pravilima opisan je u (Brill, 1992) za engleski jezik. Po autoru nazvan, Brillov označivač, jedan je od najpoznatijih označivača zasnovanih na pravilima. Za razliku od integracije ručno građenih pravila u označivač, gdje je potrebno angažirati stručnjaka za njihovu izgradnju, Brillov označivač uz predefinirane početne obrade ima ugrađen postupak otkrivanja „zakrpa" (engl. *patch*) koje odgovaraju definiranim predlošcima. Definirani predlošci sljedećeg su oblika:

- Ako je riječ označena s a i ako je u kontekstu C onda promijeni oznaku u b.
- Ako je riječ označena s a i ako ima leksičko svojstvo P onda promijeni oznaku u b.
- Ako je riječ označena s a i ako riječ u području R ima leksičko svojstvo P onda promijeni oznaku u b.

Kontekst C može biti oznaka riječi prije ili poslije trenutne, a leksičko svojstvo P može biti informacija o tome je li početno slovo veliko, konkretni prefiks ili sufiks i slično. Za svaku trojku (M_b, M_b, id), gdje je M_a trenutna oznaka, M_b prava oznaka, a id identifikator „zakrpe", broje se ispravne i neispravne primjene zakrpe, te se kao poboljšanje vrednuje razlika ta dva broja – faktor smanjenja pogrešaka.

Na primjer, u početnom nizu oznaka postoji 200 riječi koje su krivo označene kao glagol, a zapravo su trebale biti označene kao imenica, ako „zakrpa" promijeni oznaku glagola u imenicu i time ispravi 100 od 200 grešaka, ali uz taj ispravak promijeni riječi koje su stvarno glagoli u imenicu, i to napravi za njih 30, onda će se takva zakrpa vrednovati faktorom smanjenja od 70. Ona zakrpa koja postigne najveći faktor dodaje se u skup zakrpa i slijedi potraga za novom.

Označavanje se vrši u četiri koraka:

1. Nauči se jednostavan označivač unigrama koristeći veliki korpus za učenje.
2. Označivač unigrama koristi se za oznaku korpusa „zakrpa"
3. Pogreške koje postoje u korpusu „zakrpa" treba ispraviti generirajući „zakrpe" koje će ispraviti što je više pogrešaka moguće. Zakrpa s najvećim faktorom smanjenja pogreške se primjenjuje na korpus „zakrpa" i traži se sljedeća.
4. Testni se korpus prvo označi označivačem unigrama, a onda se redom kojim su zapamćene „zakrpe" primjenjuje svaka.

2.1.2. Označivači za visokoflektivne jezike

Prilagodba Brillovog označivača

Za visokoflektivne jezike postoji primjer prilagodbe Brillovog označivača u (Acedański, 2010) prilagođen za poljski jezik. Kako oznake nisu samo za vrstu riječi već i atribute (lice, rod itd.), dodan je višerazinski prolaz gdje prolaz više razine pokušava ispraviti atribute definirane za tu razinu. Predlošci su, u usporedbi s Brillovima, generalizirani tako da bi dozvolili definiranje međusobnih ovisnosti između gramatičkih atributa oznake, a i dodano je proširenje opisano u (Brill, 1994) gdje se koriste predlošci za leksičke transformacije nad riječima. Kao primjer za generalizirani predložak može se izdvojiti sljedeći: ako je riječ prije pridjev, a trenutna riječ imenica, onda se obje trebaju poklapati u padežu. Predložak je podijeljen na *akciju* i *predikat* , gdje je u prijašnjem navedenom primjeru predikat očito uvjet da je prethodna riječ pridjev, a trenutna imenica, a akcija postavljanje padeža imenice u onaj koji je već postavljen na prethodnom pridjevu. Predložak se može primijeniti na sufiks ili prefiks riječi ili na riječ maknuvši joj sufiks ili prefiks određene duljine. Kao i u 2.1.1 „zakrpe" su se otkrivale automatski opisanim algoritmom s proširenim predlošcima. Performanse

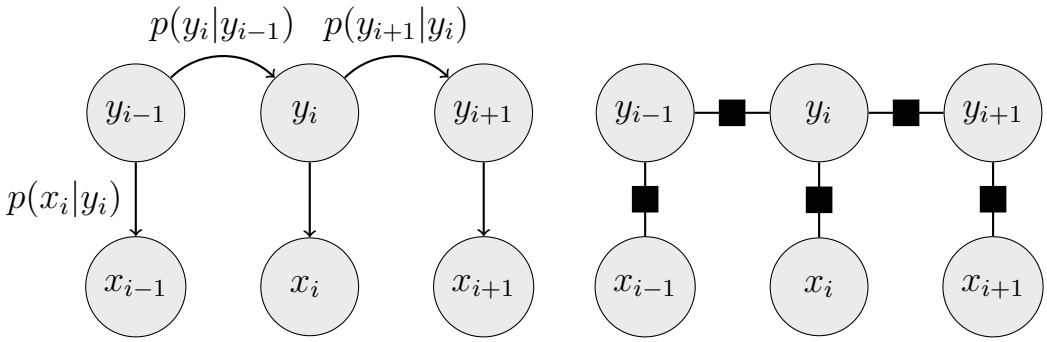
označivača (F-mjera) za poljski jezik su 89.13% za MSD, a 98.17% za POS.

Označivač za hrvatski jezik

Postoji označivač temeljen na pravilima i za hrvatski jezik opisan u (Peradin i Šnajder, 2012; Peradin, 2012). Označivač nije uopće sličan Brillovom. Prva razlika je korištenje gramatike ograničenja. Gramatika sadrži pravila za razrješavanje višeznačnosti što znači da je potrebno kao prvi korak za svaku riječ dati skup svih teoretski mogućih oznaka ako bismo željeli biti sigurni da će gramatika razrješavanjem višeznačnosti odabrati pravu oznaku za danu riječ. U ovom slučaju koristi se flektivni leksikon za morfološku analizu opisan u (Šnajder et al., 2008). Ostvarena gramatika sadrži 290 pravila za razrješavanje višeznačnosti, ali ipak je to mnogo manje u usporedbi s gramatikama zrelih sustava koji sadrže tisuće pravila. Performanse označivača su 95.30% za POS, a 86.36% za cijeli MSD.

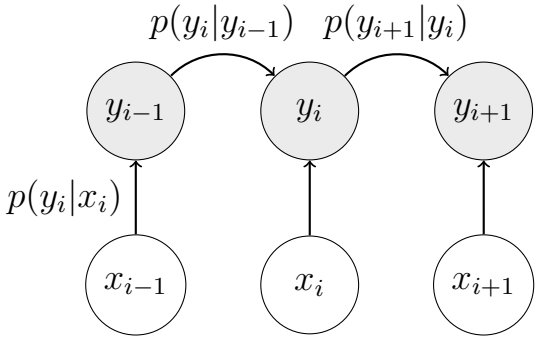
2.2. Stohastičke metode

Stohastički modeli koriste vjerojatnosti pojavljivanja oznake s riječi, a napredniji modeli uzimaju u obzir i statistička i leksička svostva riječi u okolini trenutne. U označavanju najprisutniji su grafički modeli gdje se razdioba slučajnih varijabli – kod zadatka označavanja vrste riječi to su riječi i oznake (uz druga svojstva) – pokušava faktorizirati u strukturu grafa. Strukturu možemo unaprijed odrediti (odrediti topologiju grafa) ili pretpostaviti jednostavan predložak koji se primjenjuje na svaku riječ zasebno. Grafički modeli mogu biti diskriminativni (uvjetni) i generativni. Kod diskriminativnih se modelira ovisnost oznake y o opaženoj varijabli x – u kontekstu teorije vjerojatnosti to se postiže modeliranjem razdiobe uvjetne vjerojatnosti $p(y|x)$ (engl. *conditional probability distribution*) pomoću koje možemo predvidjeti vjerojatnost oznake y ako je opažena riječ x. Kod generativnih se modelira zajednička razdioba vjerojatnosti $P(x, y)$ (engl. *joint probability distribution*) preko opaženih varijabli x i njihovih oznaka y. Grafički modeli također mogu u sebi sadržavati usmjerenost ili hibridnu usmjerenost.

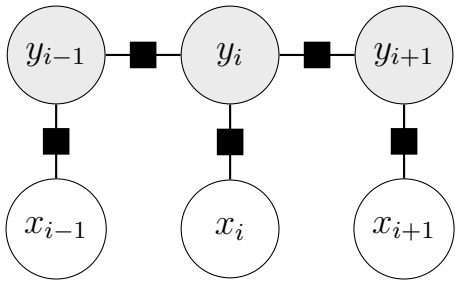


(a) Skriveni Markovljev model

(b) Markovljeva slučajna polja



(c) Markovljev model maksimalne entropije



(d) Uvjetna slučajna polja

Slika 2.1: Prikaz predložaka grafičkih modela.

Poznati grafički modeli su skriveni Markovljevi modeli (engl. *Hidden Markov Model* – generativni-usmjereni), Markovljeva slučajna polja (engl. *Markov Random Fields* – generativni-neusmjereni), Markovljevi modeli maksimalne entropije (engl. *Maximum-entropy Markov Model* – diskriminativni-usmjereni) i uvjetna slučajna polja (engl. *Conditional Random Fields* – diskriminativni-neusmjereni) – prikazani na slici 2.1, sivom bojom obojeni krugovi su ono što model može generirati. Čvorovi x_i i y_i su slučajne varijable gdje je x_i, u kontekstu označavanja riječi, opažena riječ, a y_i može poprimiti vrijednost iz skupa oznaka. Svi navedeni modeli našli su svoju primjenu i u označavanju vrste riječi.

Struktura pregleda koji slijedi preuzeta je iz (Srihari, 2014), a sadržaj iz (Wainwright i Jordan, 2008; Sutton i McCallum, 2011).

2.2.1. Generativni grafički modeli

Generativni modeli aproksimiraju združenu razdiobu vjerojatnosti slučajnih varijabli, a u slučaju potpune zajedničke razdiobe možemo generirati – zbog toga ime „generativni" – najvjerojatnije oznake y ili čak najvjerojatniji slijed opažanja x ako damo

slijed oznaka y. Kako bi se jasnije opisala primjena stohastičkih modela na problem označavanja teksta, potrebno je navesti dva klasifikatora čiji je pristup uvelike utjecao na već navedene grafičke modele.

Bayesov klasifikator

Ako nam je dana varijabla s određenim svojstvima $x = (x_1, \ldots, x_n)$ i slučajna varijabla razreda y, zajednička razdioba slučajnih varijabli $P(x, y)$ generativni je model. Ako je prisutna potpuna zajednička razdioba možemo marginalizirati $p(y) = \sum_x p(x, y)$, uvjetovati $p(y|x) = \frac{p(x,y)}{p(x)}$, a uvjetovanjem na zajedničku razdiobu možemo formirati klasifikator koji se zbog Bayesovog pravila $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$ naziva Bayesov klasifikator, opisan jednostavnim matematičkim formalizmom

$$C^{Bayes}(x) = \underset{i \in \{1,2,\ldots,m\}}{\text{argmax}} \; P(Y = y_i | X = x) \,, \tag{2.1}$$

gdje je m broj različitih razreda, a x trenutni ulaz koji promatramo. Ako x_i može poprimiti dvije vrijednosti, onda nam je za modeliranje razdiobe potrebno 2^n primjera – ovo bi uvijek bio slučaj kod modeliranja zajedničke razdiobe vjerojatnosti.

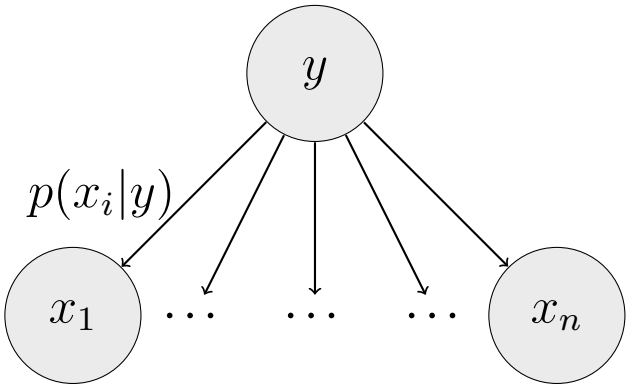
Naivni Bayesov klasifikator

Cilj je predvidjeti konkretan razred slučajne varijable y ako nam je dan vektor svojstava varijable $x = (x_1, \ldots, x_n)$. Pretpostavka jest da su svojstva x_i međusobno neovisna, što uvelike pojednostavljuje model te se združena razdioba vjerojatnosti može zapisati kao

$$p(y, x) = p(y) \prod_{i=1}^{n} p(x_i|y) \,,$$

što znači da trebamo samo n vjerojatnosti za modeliranje združene razdiobe, a razdioba se izravno faktorizira u graf na slici 2.4. Naravno, kao i u običnom klasifikatoru možemo, nakon primjene Bayesovog pravila, tražiti pravi razred za x koristeći sličan pristup kao i kod 2.1, tj. biramo razred koji je najvjerojatniji (engl. *maximum a posteriori decision rule*),

$$C^{NB}(x) = \underset{i \in \{1,2,\ldots,m\}}{\text{argmax}} \; p(y_i) \prod_{k=1}^{n} p(x_k|y_i) \,.$$



Slika 2.2: Faktorizacija modela naivnog Bayesovog klasifikatora na graf.

Skriveni Markovljev model

Kako prijašnji klasifikatori nisu pogodni za označavanje vrsta riječi, potrebno ih je proširiti na označavanje sljedova. Grafičkim modelima moguće je modelirati puno slučajnih varijabli i njihove međusobne ovisnosti, a zadatak bi bio za dani slijed opažanja $X = \{X_1, \ldots, X_n\}$ predvidjeti slijed stanja $Y = \{Y_1, \ldots, Y_n\}$ – riječi i oznake. Jedan od najpoznatijih stohastičkih grafičkih modela je skriveni Markovljev model. Skriveni Markovljev model je statistički Markovljev model u kojem se sustav modelira kao Markovljev proces – proces koji zadovoljava Markovljevo svojstvo, svojstvo da buduća razdioba vjerojatnosti procesa, za dano trenutno stanje i sva prošla stanja, ovisi samo o trenutnome stanju i niti o jednom drugome prethodnom – gdje su stanja modela (neopažena) skrivena. Skriveni Markovljevi modeli pokazali su se vrlo dobri za upravo gore opisan zadatak označavanja sljedova. Zajednička razdioba vjerojatnosti u kojoj se pretpostavlja da trenutna oznaka y_i ovisi o prošloj y_{i-1} i trenutnoj opaženoj riječi x_i glasi

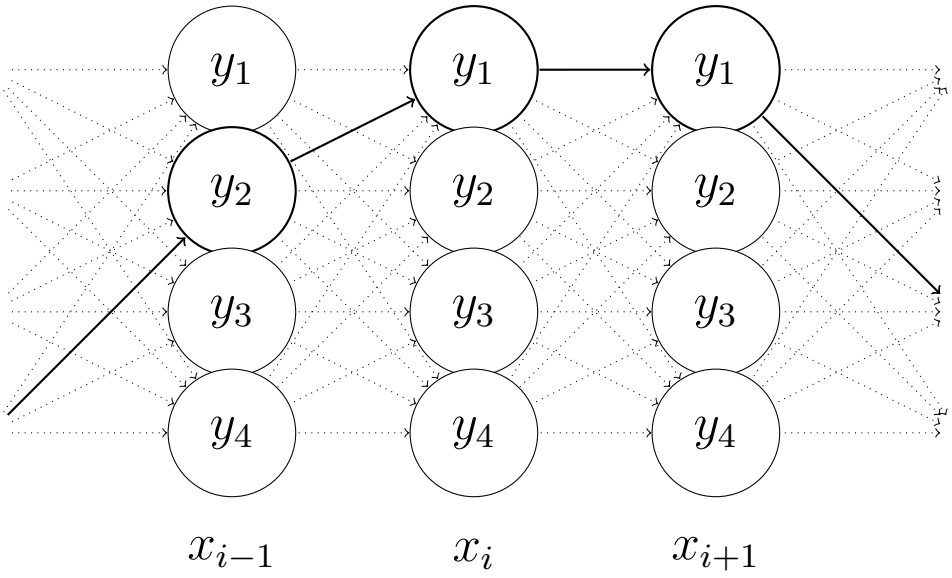
$$p(X, Y) = \prod_{i=1}^{n} p(y_i|y_{i-1})p(x_i|y_i) \,, \tag{2.2}$$

ali moguće je i pretpostaviti ovisnost o više prijašnjih oznaka bez smanjenja jednostavnosti modela. Za označavanje teksta potrebno je ili izračunati maksimalnu vjerodostojnost (engl. *maximum likelihood*) koja će aproksimirati $p(y_n|y_{n-1})$ i $p(x_n|y_n)$ ili pomoću algoritma Baum-Welch uz pomoć već označenog korpusa, a za označavanje danog niza može se iskoristiti Viterbijev algoritam. Zadnja dva navedena algoritma koriste svojstvo usmjerenosti i acikličnosti „raspletenog" lanca – prikazano na slici 2.3. U slučaju prvog, zadatak je aproksimirati prijelazne vjerojatnosti iz jednog stanja u drugo, a u slučaju drugug, naći put u grafu koji maksimizira združenu vjerojatnost $p(X, Y)$. Niti jedan navedeni pristup ne garantira globalni optimum aproksimacije za

generalne grafove, ali unatoč tome algoritmi su, za linearne lance, polinomske složenosti $O(|S|^k N)$ – upotrebom dinamičkog programiranja (naivni pristup enumeriranja svih mogućih oznaka za slijed vodi do eksponencijalne složenosti) – gdje za zanemarive veličine skupa mogućih oznaka S imaju približno linearnu složenost i moguće je gledati više koraka u nazad – k-gramski model.[1] Ako je broj mogućih oznaka velik, kao što je slučaj za hrvatski jezik (≈ 660), onda će prelazak sa skrivenog Markovljevog linearnog lanca (bigram) – prikazan na slici 2.1a – na lanac višeg stupnja drastično usporiti rad navedenih algoritama. Recimo, ako bi željeli koristiti trigramski model za označavanje vrste riječi hrvatskog jezika,

$$p(X, Y) = \prod_{i=1}^{n} p(y_n|y_{n-2}, y_{n-1})p(x_n|y_n) \ ,$$

broj koraka koji bi trebalo izvesti bio bi $\approx 300,000,000n$, zbog čega je za jezike poput hrvatskog potrebno razmotriti drugačije pristupe.



Slika 2.3: Rešetka usmjerenog acikličkog grafa za riječi x_i uz oznake y_j.

Za hrvatski jezik primijenjen je HunPos označivač opisan u (Halácsy et al., 2007). Označivač je prilagođen za korištenje postupka označavanja za više jezika, a u citiranom radu priloženi su rezultati za mađarski i engleski, a za hrvatski prisutni su u (Agić et al., 2013). U označivaču za hrvatski jezik (2.1.2) koristio se morfološki analizator

[1] Za unigramski model tražimo najvjerojatniju oznaku s obzirom na danu riječ, bigramski najvjerojatniju oznaku s obzirom na danu riječ i oznaku prethodne riječi, trigramski s obzirom na danu riječ i oznake prethodne dvije riječi itd.

koji bi dao kohortu za riječ (skup svih mogućih lema i MSD-oznaka za zadani oblik riječi), a HunPos označivač pokušava direktno iz korpusa za učenje konstruirati analizator koristeći dvije podatkovne strukture *trie*, gdje će za nepoznate i poznate riječi, ovisno o veličini prvog slova, umjesto skupa svih oznaka, biti ponuđen reducirani skup koji je bio prisutan u strukturi *trie*. Samim time složenost Viterbijevog algoritma opada – u slučaju trigramskog modela – s $O(|S|^3N)$ na $O(|\bar{S}|^3N)$, gdje je $|\bar{S}|$ prosječna veličina skupa oznaka koje izbacuje konstruirani analizator. Ako taj analizator radi dobro, onda će prosječna veličina skupa biti približno jednaka prosjeku mogućih oznaka za riječi nekog jezika. Rezultati za hrvatski jezik – na korpusu označenog sa standardom MULTEXT-East v5 – iznose 86.77% za MSD, a 97.82% za POS.

Markovljeva slučajna polja

Neusmjereni graf $G = (V, E)$, gdje je skup slučajnih varijabli $X = (X_v)_{v \in V}$ indeksiran skupom čvorova V, je Markovljevo slučajno polje ako slučajne varijable X zadovoljavaju Markovljevo svojstvo. Da bi Markovljevo svojstvo bilo zadovoljeno, potrebno je zadovoljiti sljedeća tri pravila.

1. Svake dvije nesusjedne slučajne varijable X_i i X_j moraju biti uvjetno neovisne ako je dana bilo koja druga varijabla ($\{i, j\} \notin E$).

2. Varijabla je uvjetno neovisna o svim ostalim varijablama u slučaju da je dan njen susjed.

3. Bilo koja dva podskupa varijabli A i B uvjetno su neovisna ako je dan separirajući skup S (skup gdje svaki put iz vrha skupa A do vrhu skupa B prolazi kroz vrh skupa S).

Log-linearni model je matematički model oblika

$$LL(x) = \exp\left(c + \sum_i \lambda_i f_i(X)\right)$$

čiji je logaritam polinom prvog stupnja kao funkcija parametara modela. Bilo koje Markovljevo slučajno polje može se zapisati kao log-linearni model s funkcijama značajki (engl. *feature functions*) f_i gdje je distribucija jednaka

$$P(X = x) = \frac{1}{Z}\left(\sum_i \lambda_i f_i(x_{\{i\}})\right)$$

gdje su λ_i težine modela koje treba naučiti, a Z particijska funkcija. Može se primijetiti sličnost s formulom modela 2.4 osim što je ovdje na graf faktorizirana zajednička razdioba slučajnih varijabli. Unatoč tome što Markovljeva slučajna polja nemaju široku primjenu u označavanju vrste riječi, taj model je, uz Markovljev model maksimalne entropije, poslužio kao inspiracija za uvjetna slučajna polja.

2.2.2. Diskriminativni grafički modeli

Generativni modeli pridružuju zajedničku razdiobu vjerojatnosti $p(x, y)$, a parametri se u većini slučajeva treniraju tako da maksimiziraju zajedničku vjerodostojnost (engl. *joint likelihood*) tj. želi se da vjerojatnost sljedova riječi i oznaka u korpusu bude maksimalna. Za definiranje zajedničke razdiobe potrebno je enumerirati sve moguće sljedove opažanja, a opažanja u većini slučajeva predstavljamo kao atomske entitete, kao što su riječi. Zapravo, nije praktično dodati više međuovisnih svojstava ili promatrati široki kontekst ovisnosti za opažanja jer je problem zaključivanja za takve modele netraktabilan. Navedene poteškoće jedne su od glavnih motivacija za razmatranje diskriminativnih modela. Modeliranjem uvjetne vjerojatnosti $p(y|x)$ ne troši se trud na modeliranje opaženih varijabli koje su ionako, za vrijeme označavanja, fiksirane. Također, uvjetna vjerojatnost slijeda oznaka može ovisiti o proizvoljnim svojstvima čiju međuovisnost ne moramo posebno modelirati. Ta svojstva mogu biti, ako je svaka riječ u slijedu dobila svoj skup mogućih oznaka, upravo taj skup oznaka, ili možda informacija počinju li susjedne riječi velikim slovom.

Logistička regresija

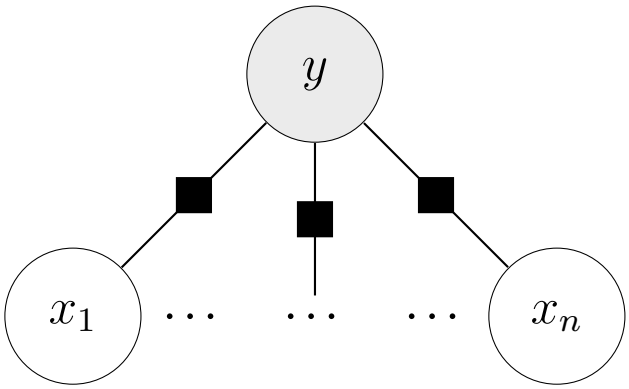
Logistička regresija (engl. *logistic regression* ili *maximum entropy classifier*) je još jedan klasifikator koji se može predstaviti kao grafički model. Klasifikator je motiviran pretpostavkom da je logaritamska vjerojatnost svakog razreda, $\log p(y|\mathbf{x})$, linearna funkcija u ovisnosti o $\mathbf{x}$ (uz normalizacijsku konstantu). Što nas vodi do zapisa za uvjetnu razdiobu

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\lambda_y + \sum_{j=1}^{K} \lambda_{y,j} x_j\right) ,$$

gdje je $Z(\mathbf{x}) = \sum_y \exp\left(\lambda_y + \sum_{j=1}^{K} \lambda_{y,j} x_j\right)$ normalizacijska konstanta, a λ_y takozvani pomak (engl. *bias*). Možemo primijetiti da svaki zaseban razred y ima svoje težine, ali u većini grafičkih modela za sljedove koristi se drugačija notacija gdje je prisutan samo jedan skup težina za sve razrede. Definiraju se funkcije značajki koje

su različite od nule samo za jedan razred. Definiramo ih kao $f_{y',j}(y, \mathbf{x}) = \mathbf{1}_{\{y'=y\}} x_j$ za težine pojedinih svojstva vektora $\mathbf{x}$, a za pomak kao $f_{y'}(y, \mathbf{x}) = \mathbf{1}_{\{y'=y\}}$. U tom slučaju, ako iskoristimo f_i za indeksiranje $f_{y',j}$ i λ_i za težine $\lambda_{y',j}$, dobijemo

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{i=1}^{K} \lambda_k f_i(y, \mathbf{x})\right) . \tag{2.3}$$



Slika 2.4: Faktorizacija modela logističke regresije na graf.

Markovljev model maksimalne entropije

Ovim su modelom postignute sve gore navedene prednosti diskriminativnih modela – detaljnije je opisan u (McCallum et al., 2000). Pretpostavimo da je prisutan slijed opažanja $X = (X_1, \ldots, X_n)$ i da tražimo slijed oznaka za X, $Y = (Y_1, \ldots, Y_n)$. X_i je bilo koja riječ jezika, a Y_i je bilo koja oznaka iz skupa oznaka S. Za Markovljev model maksimalne entropije uvjetna vjerojatnost $P(Y|X)$ se faktorizira u Markovljeve prijelazne vjerojatnosti (engl. *transition probabilities*) gdje vjerojatnost prelaska u određeno stanje oznake ovisi samo o opažanju na toj poziciji i o oznaci na prethodnoj poziciji.

$$P(Y|X) = \prod_{t=1}^{n} P(Y_t|Y_{t-1}, O_t)$$

Svaka od tih prijelaznih vjerojatnosti dio je opće razdiobe $P(y|y', x)$, gdje je za svaku moguću vrijednost prethodne oznake y' vjerojatnost oznake y modelirana isto kao i kod klasifikatora maksimalne entropije (engl. *multinomial logistic regression* ili MaxEnt),

$$P(y|y', x) = P_{y'}(y|x) = \frac{1}{Z(x, y')} \exp\left(\sum_{a} \lambda_i f_i(x, y)\right) , \tag{2.4}$$

gdje je $f_i(x, y)$ realna ili kategorička funkcija značajki, a $Z(x, y')$ je normalizacijski faktor koji osigurava da će razdioba sumirati u jedinicu. Parametri λ_i mogu biti aproksimirani generalnim iterativnim skaliranjem, a i također, pomoću varijante Baum-Welch algoritma mogu se aproksimirati parametri kada podaci nemaju sve moguće oznake. Dohvaćanje optimalnog slijeda oznaka Y moguće je koristeći varijantu Viterbijevog algoritma. Model nije savršen, a razlog tome je što, kao i kod skrivenih Markovljevih modela, potrebno je imati $O(|S|^2)$ parametara prijelaznih vjerojatnosti (bigram), što je kod jezika s velikim skupom oznaka problem. Ovaj model pati od takozvane „pristranosti oznaka" (engl. *"label bias"*) – više o tome u potpoglavlju 3.1.

3. Uvjetna slučajna polja

Mnoštvo primjena grafičkih modela orijentiralo se na korištenje generativnih modela, poput skrivenih Markovljevih modela, gdje se modelira zajednička razdioba vjerojatnosti $p(\mathbf{y}, \mathbf{x})$ preko ulaza $\mathbf{x}$ i izlaza $\mathbf{y}$. Takav pristup ima svojih prednosti, ali ima i važna ograničenja. Ako je dimenzionalnost $\mathbf{x}$ velika, ali i svojstva ulaza su međusobno ovisna, modeliranje takve distribucije nije jednostavno.

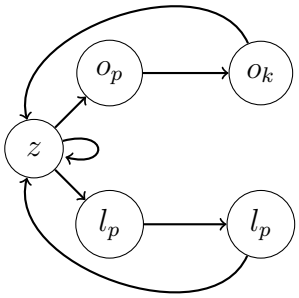
Diskriminativni pristup rješava taj problem i to je upravo pristup modela uvjetnih slučajnih polja. U uvjetnim slučajnim poljima prikupljene su prednosti diskriminativne klasifikacije i grafičkog modeliranja gdje se lako može modelirati višekategorijski izlaz $\mathbf{y}$ s mogućnošću korištenja velikog broja svojstava ulaza $\mathbf{x}$. Prednost uvjetnog modela je ta što međuovisnosti u svojstvima ulazne varijable $\mathbf{x}$ ne igraju nikakvu ulogu u modelu jer nije potrebno modelirati distribuciju preko njih.

U nastavku ovog poglavlja obradit će se algoritamski i reprezentativni dio modela uvjetnih slučajnih polja i osvrnuti se na problem pristranosti oznaka. Kako sam uvod u terminologiju grafičkih modela nije u opsegu ovog rada, za detaljniji pregled modeliranja stohastičkih procesa grafičkim modelima čitatelj se upućuje na (Wainwright i Jordan, 2008; Sutton i McCallum, 2011).

3.1. Pristranost oznakama – „label bias”

Primjer koji slijedi u nastavku je, s malim preinakama, preuzet iz (Grishman, 2014).

Radi jednostavnosti, uzmimo problem označavanja osoba i lokacija gdje svaka osoba i svaka lokacija imaju dvorječno ime. Skup mogućih oznaka S sadrži $\{o_p, o_k, l_p, l_k, z\}$ gdje su oznake – redom – za označavanje početka i kraja imena osobe, početka i kraja imena lokacije i oznaka za sve ostale riječi.



Imena	$\|o\|$	$\|l\|$
Ivan Ivić	9	1
Ivan Park	1	9
Sara Ivić	9	1
Sara Park	1	9

$P(y_i\|y_{i-1}, x)$	Iznos
$p(o_p\|z, w = \text{Ivan})$	0.5
$p(l_p\|z, w = \text{Ivan})$	0.5
$p(o_p\|z, w = \text{Sara})$	0.5
$p(l_p\|z, w = \text{Sara})$	0.5
$p(o_k\|o_p, w = \text{Ivić})$	1
$p(o_k\|o_p, w = \text{Park})$	1
$p(l_k\|l_p, w = \text{Ivić})$	1
$p(l_k\|l_p, w = \text{Park})$	1

Slika 3.1: Prikaz problema pristranosti oznaka.

Na slici 3.1 priložena je tablica s brojem pojavljivanja imena označenog s oznakom osobe ili lokacije u korpusu, a kraj te tablice priložene su vrijednosti uvjetnih vjerojatnosti. Graf prikazuje moguće prijelaze stanja koji ovise o ulaznoj riječi. Iz samih vrijednosti uvjetnih vjerojatnosti može se primijetiti da je uloga za oznaku kraja imena, bilo kod osobe ili lokacije, u potpunosti izgubljena. Da je neka od dvije vjerojatnosti oznake početka kod konkretnog imena malo veća od druge, u odlučivanju optimalne oznake uzeli bi u obzir put koji bi odbacio informaciju druge riječi. Problem je u tome što se vjerojatnosti za bridove računaju odvojeno za svako stanje dok su uvjetna slučajna polja jedan model za združenu vjerojatnost cijelog slijeda oznaka. Svi negenerativni modeli s konačnim brojem stanja pate od pristranosti oznakama, a eksperimentalno su rezultati pokazani u (Lafferty et al., 2001). Uvjetna slučajna polja, u ovom slučaju, imaju prednost jer pojava pristranosti nije moguća.

3.2. Linearni lanac

Postoji očita povezanost između naivnog Bayesovog modela i logističke regresije – oni dvoje formiraju diskriminativno-generativni par. Kod označavanje sljedova generativni model skrivenog Markovljevog modela svoj diskriminativni analog pronalazi u specijalnom slučaju modela uvjetnih slučajnih polja – linearni lanac. U nastavku će se pokazati da uvjetna razdioba $p(\mathbf{y}|\mathbf{x})$ koja proizlazi iz zajedničke razdiobe $p(\mathbf{y}, \mathbf{x})$ skrivenih Markovljevih modela je zapravo uvjetno slučajno polje s dodatkom funkcija značajki. Postupke zaključivanja i procjene parametara skrivenih Markovljevih modela možemo za linearni lanac uvjetnih slučajnih polja samo prilagoditi te se u nastavku takav postupak provodi.

3.2.1. Definicija

Možemo ponovo napisati zajedničku vjerojatnost skrivenih Markovljevih modela (2.2) u obliku kojeg je lakše generalizirati

$$p(\mathbf{y}, \mathbf{x}) = \frac{1}{Z} \prod_{t=1}^{T} \exp\left(\sum_{i,j \in S} \theta_{ij} \mathbf{1}_{\{y_t = i\}} \mathbf{1}_{\{y_{t-1} = j\}} + \sum_{i \in S} \sum_{o \in O} \mu_{oi} \mathbf{1}_{\{y_t = i\}} \mathbf{1}_{\{x_t = o\}} \right), \quad (3.1)$$

gdje su $\theta = \{\theta_{ij}, \mu_{oi}\}$ realne vrijednosti distribucije, a Z[1] normalizacijska konstanta tako da razdioba sumira u 1. Ako definiramo

$$\begin{aligned} \theta_{ij} &= \log p(y' = i | y = j) \\ \mu_{oi} &= \log p(x = o | y = i) \\ Z &= 1 \end{aligned}$$

dobivamo upravo zajedničku razdiobu skrivenog Markovljevog modela. Možemo zapisati (3.1) jednostavnije ako, po uzoru na (2.3), uvedemo funkcije značajki oblika $f_k(y_t, y_{t-1}, x_t)$. Za svaki par prijelaza treba postojati funkcija značajki $f_{ij}(y, y', x) = \mathbf{1}_{\{y=i\}} \mathbf{1}_{\{y'=j\}}$ i $f_{io}(y, y', x) = \mathbf{1}_{\{y=i\}} \mathbf{1}_{\{x=o\}}$. Funkcija značajki f_k obuhvaća sve definirane funkcije i onda izraz možemo zapisati kao

$$p(\mathbf{y}, \mathbf{x}) = \frac{1}{Z} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right) . \quad (3.2)$$

Zadnji korak prije mogućnosti definiranja uvjetnih slučajnih polja jest zapisati uvjetnu razdiobu $p(\mathbf{y}|\mathbf{x})$ koja proizlazi iz (3.2). Ona glasi

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y}, \mathbf{x})}{\sum_{y'} p(\mathbf{y'}, \mathbf{x})} = \frac{\prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right)}{\sum_{y'} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right)} . \quad (3.3)$$

Ovakva razdioba (3.3) predstavlja linearni lanac uvjetnih slučajnih polja koji kao svojstva uzima u obzir je li riječ prisutna ili ne, ali ono što ćemo kasnije vidjeti, moguće je definirati niz proizvoljnih funkcija značajki koje mogu uzimati u obzir svojstva okoline – identitet riječi, njihova leksička svojstva, skup mogućih oznaka itd.

Definicija 1. Neka su Y,X slučajni vektori, $\theta = \{\theta_k\} \in \Re^K$ vektor parametara i $\mathcal{F} = \{f_k(y, y', \mathbf{x}_t)\}_{k=1}^{K}$ skup funkcija značajki realnih vrijednosti. Linearni lanac uvjetnih slučajnih polja je razdioba $p(\mathbf{y}|\mathbf{x})$ oblika

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right) , \quad (3.4)$$

[1] $Z = \sum_{\mathbf{y}} \sum_{\mathbf{x}} \prod_{t=1}^{T} \exp\left(\sum_{i,j \in S} \theta_{ij} \mathbf{1}_{\{y_t = i\}} \mathbf{1}_{\{y_{t-1} = j\}} + \sum_{i \in S} \sum_{o \in O} \mu_{oi} \mathbf{1}_{\{y_t = i\}} \mathbf{1}_{\{x_t = o\}} \right)$

gdje je $Z(\mathbf{x})$ normalizacijska funkcija ovisna o ulazu

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right) . \tag{3.5}$$

3.2.2. Zaključivanje

U ovom dijelu opisat će se algoritmi zaključivanja za skrivene Markovljeve modele i pokazati prilagodba za uvjetna slučajna polja. Algoritmi zaključivanja koji se koriste su algoritam *forward-backward* za izračun marginalnih razdiobi i *Viterbijev* algoritam za pronalaženje optimalnog slijeda oznaka.

Forward-backward algoritam

Skriveni Markovljev model možemo faktorizirati na graf $p(\mathbf{y}, \mathbf{x}) = \prod_t \Psi_t(y_t, y_{t-1}, x_t)$ gdje je $Z = 1$, a faktori su definirani kao

$$\Psi_t(j, i, x) \stackrel{\text{def}}{=} p(y_t = j | y_{t-1} = i) P(x_t = x | y_t = j) . \tag{3.6}$$

$\Psi_t(j, i, x)$ možemo interpretirati kao težinu prijelaza iz stanja i u stanje j za trenutno opažanje x. *Forward* algoritam koristimo za izračun vjerojatnosti opažanja $p(\mathbf{x})$. Ideja je prvo napisati naivan zbroj $p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$, a onda koristeći zakon distributivnosti

$$p(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{t} \Psi_t(y_t, y_{t-1}, x_t) \tag{3.7}$$

$$= \sum_{y_T} \sum_{y_{T-1}} \Psi_T(y_T, y_{T-1}, x_T) \sum_{y_{T-2}} \Psi_{T-1}(y_{T-1}, y_{T-2}, x_{T-1}) \sum_{y_{T-3}} \cdots \tag{3.8}$$

gdje dobijemo oblik takav u kojem je lako prepoznati da se sume u međukoracima više puta izračunavaju i pomoću dinamičkog programiranja moguće je uštedjeti eksponencijalno na vremenu. Potrebno je definirati rekurzivnu relaciju koja će nam omogućiti implementaciju algoritma dinamičkog programiranja.

Možemo definirati skup *forward* varijabli α_t, svaka vektor dimenzije M (gdje je M broj stanja), gdje svaka predstavlja sume u međukoracima.

$$\alpha_t(j) \stackrel{\text{def}}{=} p(x_{\langle 1 \dots t \rangle}, y_t = j) \tag{3.9}$$

$$= \sum_{\mathbf{y}_{\langle 1 \dots t-1 \rangle}} \Psi_t(j, y_{t-1}, x_t) \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) , \tag{3.10}$$

gdje je zbroj preko $\mathbf{y}_{\langle 1 \dots t-1 \rangle}$ intervala za sve moguće vrijednosti slučajnih varijabli $y_1, y_2, \dots, y_{t-1}$ – ovo za jezike s velikim brojem oznaka stvara probleme i tu (3.3) daje

svoj doprinos. Sada možemo izračunati $\alpha(j)$ koristeći

$$\alpha_t(j) = \sum_{i \in S} \Psi_t(j, i, x_t)\alpha_{t-1}(i) \,, \tag{3.11}$$

s početnom vrijednošću $\alpha_1 = \Psi_1(j, y_0, x_1)$ gdje je y_0 ona pomoćna oznaka prije početka rečenice.

Backward rekurzivna relacija je potpuno ista samo što su sume (3.8) obrnute što vodi do definicije

$$\beta_t(i) \stackrel{\text{def}}{=} p(x_{\langle t+1 \ldots T \rangle}, y_t = i) \tag{3.12}$$

$$= \sum_{\mathbf{y}_{\langle t+1 \ldots T \rangle}} \prod_{t'=t+1}^{T} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) \,, \tag{3.13}$$

a rekurzija koja slijedi je

$$\beta_t(i) = \sum_{j \in S} \Psi_{t+1}(j, i, x_{t+1})\beta_{t+1}(j) \,, \tag{3.14}$$

gdje je početna vrijednost $\beta_T(i) = 1$.

Za izračun marginalnih razdioba $p(y_{t-1}, y_t|\mathbf{x})$ koje su nam potrebne za procjenu parametara, mogu se rezultati *forward* i *backward* rekurzija ponovo iskoristiti:

$$p(y_{t-1}, y_t|\mathbf{x}) = \frac{p(\mathbf{x}|y_{t-1}, y_t)p(y_{t-1}, y_t)}{p(\mathbf{x})} \tag{3.15}$$

$$= \frac{p(\mathbf{x}_{\langle 1 \ldots t-1 \rangle}, y_{t-1})p(y_t|y_{t-1})p(x_t|y_t)p(\mathbf{x}_{\langle t+1 \ldots T \rangle}|y_t)}{p(\mathbf{x})} \tag{3.16}$$

$$= \frac{1}{p(\mathbf{x})}\alpha_{t-1}(y_{t-1})\Psi_t(y_t, y_{t-1}, x_t)\beta_t(y_t) \,, \tag{3.17}$$

gdje se u drugoj jednakosti koristi činjenica da je $\mathbf{x}_{\langle 1 \ldots t-1 \rangle}$ neovisan o $\mathbf{x}_{\langle t+1 \ldots T \rangle}$ i o x_t ako je dan y_{t-1}, y_t. Faktor $1/p(\mathbf{x})$ služi kao normalizacijska konstanta, a možemo ga izračunati koristeći $p(\mathbf{x}) = \beta_0(y_0)$ ili $p(\mathbf{x}) = \sum_{i \in S} \alpha_T(i)$.

Ako sve navedene dijelove spojimo algoritam *forward-backward* glasi:

---

1. Izračunati α_t za sve t koristeći (3.11).
2. Izračunati β_t za sve t koristeći (3.14).
3. Vratiti izračunate marginalne razdiobe koristeći (3.17)

Viterbijev algoritam

Potrebno je izračunati najvjerojatniju dodjelu slijeda oznaka $y^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$, a iz (3.8) moguće je zamijeniti sume operatorom maksimizacije što nas vodi do Viterbi rekurzivne relacije

$$\delta_t(j) \stackrel{\text{def}}{=} \max_{\mathbf{y}_{\langle 1\ldots t-1\rangle}} \Psi_t(j, y_{t-1}, x_t) \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) \,, \tag{3.18}$$

$$\delta_t(j) = \max_{i \in S} \Psi_t(j, i, x_t)\delta_{t-1}(i) \,, \tag{3.19}$$

što je analogno *forward* rekurziji, a da bi pronašli slijed njega možemo izračunati pomoću analoga *backward* rekurzije

$$y_t^* = \arg\max_{i \in S} \Psi_t(y_{t+1}^*, i, x_{t+1})\delta_t(i) \quad \text{za } t < T \tag{3.20}$$

Rekurzivne relacije δ_t i y_t^* zajedno obuhvaćaju *Viterbijev algoritam*. Sada nakon ovih definicija može se povući paralela s uvjetnim slučajnim poljima. Rekurzivne relacije izgledaju gotovo identično samo je $\Psi_t(j, i, x_t)$ drugačije definiran. Model iz (3.4) možemo zapisati kao

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \mathbf{x}_t) \,, \tag{3.21}$$

gdje je

$$\Psi_t(y_t, y_{t-1}, \mathbf{x}_t) = \exp\left(\sum_k \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right) \,. \tag{3.22}$$

Zbog takve definicije algoritmi za *forward* rekurziju (3.11), *backward* rekurziju (3.14) i Viterbi rekurziju (3.19) mogu se iskoristi bez ikakvih promjena za linearni lanac uvjetnih slučajnih polja. U slučaju uvjetnih slučajnih polja $p(\mathbf{x})$ je sada $Z(\mathbf{x})$ gdje izrazi za izračun $p(\mathbf{x})$ sada drugačije izgledaju – $Z(\mathbf{x}) = \beta_0(y_0)$ i $Z(\mathbf{x}) = \sum_{i \in S} \alpha_T(i)$. Marginalne razdiobe isto poprimaju drugačiji zapis,

$$p(y_{t-1}, y_t|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\alpha_{t-1}(y_{t-1})\Psi_t(y_t, y_{t-1}, x_t)\beta_t(y_t) \,, \tag{3.23}$$

$$p(y_t|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\alpha_{t-1}(y_{t-1})\beta_t(y_t) \,. \tag{3.24}$$

3.2.3. Ocjena parametara

Da bi model mogao biti iskorišten za klasifikaciju potrebno je pravilno aproksimirati parametre $\theta = \{\theta_k\}$.

Jedan od mogućih načina je tražiti maksimalnu vjerodostojnost, tj. parametri su izabrani tako da podaci u korpusu imaju najveću vjerojatnost u modelu.

Za linearni lanac uvjetnih slučajnih polja, parametre za maksimalnu vjerodostojnost moguće je izračunati numeričkim optimizacijskim metodama. Ako nam je dan korpus $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$ gdje je $\mathbf{x}^{(i)} = \{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_T^{(i)}\}$ slijed ulaza, a $\mathbf{y}^{(i)} = \{\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \ldots, \mathbf{y}_T^{(i)}\}$ slijed željenih oznaka za $\mathbf{x}$. Da bi se pojednostavila notacija pretpostavlja se da svaki slijed ima istu duljinu T, ali u generalnom slučaju očito je da sljedovi mogu biti različitih duljina.

Za procjenu parametara koristi se uvjetna log-vjerodostojnost (engl. *conditional log-likelihood*):

$$\ell(\theta) = \sum_{i=1}^{N} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}; \theta) \, . \tag{3.25}$$

Da bi izračunali maksimalnu vjerodostojnost moramo ju maksimizirati s obzirom na θ. Nakon što uvrstimo model uvjetnih slučajnih polja (3.4) u (3.25) dobit ćemo sljedeći izraz:

$$\ell(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k({y_t}^{(i)}, {y_{t-1}}^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)}) \, . \tag{3.26}$$

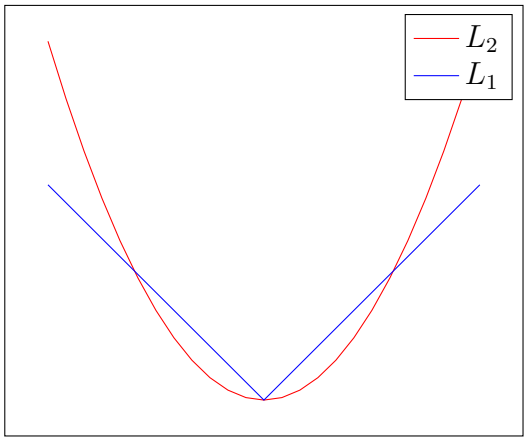
U slučaju velikog broja parametara θ korisno je koristiti *regularizaciju* koja daje kaznu na težinske vektore čija je norma prevelika. Čest izbor kazne je euklidska norma θ parametra koristeći *regularizacijski parametar* $1/2\sigma^2$ koji određuje snagu kazne. Ako dodamo regularizaciju izraz za maksimalnu vjerodostojnost glasi:

$$\ell(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k({y_t}^{(i)}, {y_{t-1}}^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)}) - \sum_{k=1}^{K} \frac{\theta_k^2}{2\sigma^2} \, . \tag{3.27}$$

Parametar σ^2 slobodan je parametar s kojim možemo odrediti koliko ćemo kazniti velike težine. Intuitivno, ideja je smanjiti potencijal da mali broj svojstava dominira predviđanje. Regularizaciju također možemo promatrati kao računanje MAP procjene (engl. *maximum a posteriori estimation*) težina θ, u slučaju da je dodijeljena Gaussova apriorna razdioba s očekivanjem 0 i kovarijancom $\sigma^2 I$. Pronalazak najboljeg parametra regularizacije zahtijeva intenzivno pretraživanje. U velikom broju slučajeva model nije mnogo osjetljiv na male promjene u σ^2.

Postoji i drugi izbor regularizacije, možemo koristiti L_1-normu umjesto euklidske:

$$\ell(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k({y_t}^{(i)}, {y_{t-1}}^{(i)}, \mathbf{x}^{(i)}) - \sum_{i=1}^{N} \log Z(\mathbf{x}^{(i)}) - \sum_{k=1}^{K} \frac{|\theta_k|}{\beta} \, . \tag{3.28}$$



Slika 3.2: Prikaz L_1 i L_2 regularizacijskih funkcija

Ovakav regularizator potiče rijetkost u naučenim parametrima. L_2-norma optimizira parametre na konkavnoj funkciji koja je kvadratna – kao što se može vidjeti sa slike 3.2 – i blizu minimuma pomaci po gradijentu su sve sporiji s čime puno parametara postiže vrijednost blizu nule. L_1-norma izgleda slično kao $|x|$ gdje će onda mnogo parametara dostići vrijednost nule. Tako da se istovremeno, koristeći L_1-normu, može vršiti optimizacija parametara i selekcija bitnih svojstava (engl. *feature selection* ili *structure learning*).

U generalnom slučaju, $\ell(\theta)$ ne možemo maksimizirati u zatvorenom obliku te je potrebno koristiti numeričke metode. Parcijalna derivacija (3.27) je

$$\frac{\partial \ell}{\partial \theta_k} = \sum_{i=1}^{N} \sum_{t=1}^{T} f_k(y_t{}^{(i)}, y_{t-1}{}^{(i)}, \mathbf{x}^{(i)}) \tag{3.29}$$

$$- \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{y,y'} f_k(y, y', \mathbf{x}_t{}^{(i)}) p(y, y'|\mathbf{x}^{(i)}) - \frac{\theta_k}{\sigma^2} \tag{3.30}$$

Za izračun vjerodostojnosti $\ell(\theta)$ i njene derivacije potrebno je koristiti algoritme zaključivanja. Potrebno je izračunati particijsku funkciju $Z(\mathbf{x}^{(i)})$, koja iznosi zbroju preko svih mogućih oznaka i potrebno je izračunati marginalnu razdiobu $p(y, y'|\mathbf{x}^{(i)})$. Kako obje veličine ovise o $\mathbf{x}^{(i)}$ trebat ćemo pokrenuti postupke zaključivanja za svaki primjerak slijeda iz korpusa. U usporedbi s Markovljevim slučajnim poljem (2.2.1) kod kojeg particijska funkcija ovisi samo o parametrima ovaj izračun se čini suvišan, ali nema načina da se ovaj postupak zaobiđe. Bez obzira na dodatno računsko opterećenje mora se naglasiti da kod Markovljevog slučajnog polja modeliramo cijelu zajedničku razdiobu tako da usporedba s uvjetnim slučajnim poljima nema veliku važnost. Upotrebom metode poput stohastičkog gradijentnog spusta (engl. *stochastic gradient descent*) moguće je u manje od 100 iteracija doći blizu optimuma u većini primjena.

Kao što je već spomenuto, funkcija $\ell(\theta)$ uz L_2 regularizaciju je konkavna funkcija i mogu se koristiti gradijentne metode s garancijom postignuća globalnog optimuma. Uobičajeno je koristiti i naprednije metode kao što su Newtonova metoda, kvazi-Newtonova metoda i slično – da bi se ubrzala optimizacija.

Složenost cijelog postupka lako se može izračunati. Algoritmi zaključivanja rade na linearnom lancu stoga je složenost algoritma *forward-backward* $O(TM^2)$ gdje je T duljina niza, a M veličina skupa svih mogućih oznaka. Kako se postupci zaključivanja trebaju pokrenuti N puta (za svaki niz u korpusu) ukupna složenost, ako se koristi gradijentni postupak, iznosi $O(TM^2NG)$ gdje je G broj iteracija gradijentnog postupka.

3.3. Uvjetna slučajna polja s ograničenjima

Moglo se primijetiti u prošlom poglavlju da algoritmi koji se koriste za zaključivanje i procjenu parametara ovise o veličini skupa svih mogućih oznaka S. U kontekstu označavanja vrste riječi za hrvatski jezik gdje je skup oznaka poprilično velik očito je da i kod modela uvjetnih slučajnih polja moramo nekako zaobići potrebu modeliranja razdiobe preko cijelog skupa mogućih oznaka za svaku riječ u slijedu. Model uvjetnih slučajnih polja s ograničenjima, prvi puta uveden u (Waszczuk, 2012) je upravo riješio i taj problem. Cijeli proces označavanja vrste riječi rastavljen je u dva dijela. Prvi dio ovisi o kvalitetnom morfosintaktičkom analizatoru koji bi trebao za svaku riječ u rečenici ponuditi skup mogućih oznaka (skup ograničenja). Nakon toga prvi model bi trebao za riječi koje nisu dobile neprazni skup konstruirati skup mogućih oznaka ovisno o svojstvima same riječi i njene okoline. Kada je to gotovo počinje druga faza označavanja. Drugi model trebao bi iz svakog skupa mogućih oznaka odabrati onu pravu. Tako završava procedura označavanja, a u nastavku će se svaku podrobnije promotriti.

Bez smanjenja općenitosti, pri definiciji, pretpostavljamo da je model linearni lanac i da je razdioba definirana preko mogućih oznaka i mogućih riječi gdje su uključena razna svojstva koja obuhvaćaju svojstva same riječi ili njenog konteksta.

3.3.1. Definicija

Neka je O skup opažanja, a Y skup oznaka, a $X = 2^O$. Neka je $\mathbf{x} = (x_1 \in X, \ldots, x_n \in X)$ ulazni sljed riječi, gdje je svaka riječ predstavljena opisnim skupom opažanja i $\mathbf{y} = (y_1 \in Y, \ldots, y_n \in Y)$ izlaznom slijedu oznaka. Neka je $\mathbf{r} = (r_1 \subseteq Y, \ldots, r_n \subseteq Y)$

slijed nepraznih ograničenja preko skupa oznaka za dani slijed **x**. Također pretpostavljamo da je $x_i = \emptyset$ i $y_i = \delta$ za $i < 1 \vee i > n$, gdje je δ pomoćna oznaka za pozicije izvan intervala slijeda. S obzirom na dana ograničenja nije teško prilagoditi definiciju (3.4):

$$p(\mathbf{y}|\mathbf{x}, \mathbf{r}) = \begin{cases} \frac{1}{Z(\mathbf{x},\mathbf{r})} \prod_{i=1}^{n} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_i, y_{i-1}, x_t)\right) , & \text{ako } \mathbf{y} \in \prod_{i=1}^{n} r_i \, . \\ 0 & \text{inače} \, . \end{cases} \tag{3.31}$$

Normalizacijski faktor u kontekstu ograničenja definira se:

$$Z(\mathbf{x}) = \sum_{\mathbf{y} \in \prod_{i=1}^{n} r_i} \prod_{i=1}^{n} \exp\left(\sum_{k=1}^{K} \theta_k f_k(y_i, y_{i-1}, x_i)\right) . \tag{3.32}$$

Postupci zaključivanja i optimizacije nisu puno drugačiji, jedino se prolazak kroz cijeli skup mogućih oznaka Y zamijenio prolaskom kroz oznake **y** koje zadovoljavaju ograničenja **r**.

3.3.2. Morfosintaktičko pogađanje

Model dobije skup ograničenja **r** za riječi koje je morfosintaktički analizator prepoznao. Za sve ostale riječi potrebno je konstruirati skup ograničenja u ovisnosti o raznim svojstvima.

Konstrukcija

Model je treniran koristeći L_1-normu (3.28) za koju je posebno razvijen brži algoritam stohastičnog gradijentnog spusta opisan u (Sokolovska et al., 2010). Ukratko, kako se istovremeno korištenjem L_1-norme vrši selekcija korisnih svojstava pokazalo se da je, zbog rijetkosti modela, moguće ubrzati izračun gradijenta i samim time ubrzati iteracije gradijentnog postupka. Navedena norma je korištena zbog nepotrebne visoke razine točnosti, skup mogućih ograničenja ne mora biti savršen, ali je zato stavljeno veće težište na drugi dio razrješavanja višeznačnosti.

Opažanja

Često rješenje za postupak morfosintaktičkog pogađanja jest konstrukcija skupa mogućih oznaka promatranjem ortografskih svojstava same riječi (npr. prefiksi i sufiksi). Kako je moguće definirati razne funkcije značajki tako su se, za ovaj model, uzeli u

obzir skupovi ograničenja susjednih pozicija. Prednost ovakvog pristupa jest ta da je u postupku pogađanja moguće odbaciti morfosintaktičke opisnike s obzirom na ortografska svojstva i konteksta istovremeno.

U implementaciji modela za poljski jezik definirali su se prefiksi i sufiksi duljine 1 i 2, binarna vrijednost koja nam govori je li riječ poznata ili ne (je li morfosintaktički analizator dao neprazan skup ograničenja) i spremio se skraćen oblik riječi (npr. „Kokoš-2014" — „ullllxdddd"), gdje se kapitalizirano slovo zamijenilo znakom 'u', znamenke znakom 'd', malo slovo znakom 'l', a svi ostali znakovi znakom 'x'. Sva uzastopna ponavljanja znaka unutar skraćenog niza su izbrisana. U skraćen oblik riječi dodaje se i informacija je li riječ na prvom mjestu u rečenici.

3.3.3. Morfosintaktičko razrješavanje višeznačnosti

Zadatak modela za razrješavanje višeznačnosti je, za svaku riječ, iz zadanog skupa mogućih oznaka odabrati onu pravu (ako se prava ne nalazi odabrat će se najvjerojatnija).

Konstrukcija

Postupak razrješavanja višeznačnosti zadnji je korak u modelu uvjetnih slučajnih polja s ograničenjima te je na njemu stavljeno najveće težište. Bitno je da iz skupa oznaka model uspije pronaći onu pravu stoga je za njegovu konstrukciju odabrana L_2-norma i linearni lanac uvjetnih slučajnih polja drugog stupnja (uz dodijeljenu oznaku prošle riječi gleda se i dodijeljena oznaka pretprošle). Ovdje se za postupak učenja ponovo koristi stohastički gradijentni spust, ali zbog L_2-norme i većeg broja opažanja (koje su navedene u nastavku) postupak učenja je puno sporiji, ali zato je povećana vjerojatnost ispravnog odabira.

Opažanja

U postupku razrješavanja višeznačnosti koristi se bogatiji skup opažanja u usporedbi s postupkom pogađanja. Neka je w_i riječ na i-toj poziciji u ulaznom slijedu. Koristi se sljedeći skup opažanja:

- ortografski oblici (mala slova) riječi w_{i-1}, w_i i w_{i+1},
- ako riječ w_i nije poznata:
    - prefiksi i sufiksi duljine 1, 2 i 3 riječi w_i (mala slova)
    - skraćen oblik riječi w_i i informacija je li riječ pozicionirana na prvom mjestu u rečenici

Za poznate riječi u (Waszczuk, 2012) spomenuto je da nisu viđena poboljšanja ako je promatran skraćeni oblik riječi za koje je dan neprazan skup ograničenja stoga skraćeni oblik nije korišten.

Funkcije značajki

Morfosintaktički opisnici sastoje se od više atributa uz kategoriju vrste riječi stoga je za model razrješavanja višeznačnosti definiran niz novih funkcija značajki f_k koje bi trebale pomoći u odabiru. Funkcije, zbog lanca drugog reda, poprimaju drugačiji oblik nego što je prikazan u (3.31). Definiran je i skup slojeva u koje je moguće rastaviti atribute svih morfosintaktičkih opisnika (preporučuje se da u sloju budu atributi sa što većom neovisnošću). Neka je L broj slojeva i $y(l)$ dio morfosintaktičkog opisnika y koji je dodijeljen l-tom sloju za $l \in \{1, \ldots, L\}$. Slojeviti model uzima u obzir unigram (l, u, o) i prijelazna (l, w, v, u) svojstva na sličan način prije opisan u (3.2.1).

$$f_k(x_i, y_i, y_{i-1}, y_{i-2}) = \begin{cases} \mathbf{1}(y_i(l) = u, o \in x_i) & k \text{ za } (l, u, o) \\ \mathbf{1}(y_i(l) = u, y_{i-1}(l) = v, y_{i-2}(l) = w) & k \text{ za } (l, w, v, u) \end{cases} \tag{3.33}$$

U implementaciji za poljski jezik POS, padež i lice su u jednom sloju, a ostali atributi u drugom.

3.3.4. Rezultati

Za poljski jezik ovaj model pokazao se najuspješnijim do sada. S obzirom na da to jezik ima više od 1000 različitih oznaka rezultati nisu visoki kao kod najboljih implementacija označivača za engleski jezik, ali su jedni od boljih za visokoflektivne jezike. Usporedba ovog označivača s ostalim poljskim označivačima može se pronaći u (Pohl i Ziółko, 2013; Radziszewski, 2013). Rezultati koje je model postigao iznose 91.44% za MSD, a čak 59.19% za nepoznate riječi.

4. Označivač za hrvatski jezik

U okviru ovog rada razvijen je označivač za hrvatski jezik temeljen na modelu uvjetnih slučajnih polja s ograničenjima. Za vrijeme pisanja ovog rada najbolji rezultati za hrvatski jezik navedeni su u (Agić et al., 2013), gdje je za označavanje vrste riječi korišten HunPos označivač i postigao je na reduciranom skupu oznaka točnost od 97.13% za POS i 87.72% za MSD, rezultati na skupu oznaka koje su korištene u evaluaciji razvijenog označivača u okviru ovog radu su 97.04% za POS i 86.77% za MSD. Više o samoj implementaciji HunPosa bilo je rečeno u potpoglavlju 2.2.

4.1. Implementacija

Programski jezik korišten u razvoju označivača je Haskell.[1] Sam model uvjetnih slučajnih polja s ograničenjima razvijen je u potpunosti u Haskellu te je za prilagodbu na hrvatski jezik bilo potrebno koristiti programski jezik zbog razlike u zapisu morfosintaktičkih opisnika i morfosintaktičkih analizatora.[2] Haskell je čisti funkcijski jezik što znači da nije moguće imati „nuspojave" u programu. Uz navedenu prednost Haskell posjeduje i sposobnost zaključivanja o tipovima u programu, što je omogućilo brz razvoj i testiranje (poput korištenja skriptnog jezika) s dodatnom zaštitom tipova gdje postoji garancija da funkcija radi ono što želimo. Tijekom razvoja označivača za hrvatski jezik ni u jednom trenutku nije postojao trenutak nevidljivih grešaka u kodu (engl. *bug*), što je uvelike ubrzalo i olakšalo razvoj.

4.1.1. Morfosintaktički analizator

Model koji koristimo za označavanje zahtijeva korištenje kvalitetnog morfosintaktičkog analizatora (3.3.2) – sustav koji bi za riječ u rečenici morao izbaciti sve teoretski

[1] `http://www.haskell.org/`

[2] Implementacija za poljski jezik zvana *concraft-pl* prilagodba je opće implementacije *concraft*, a obje se mogu pronaći na
`http://hackage.haskell.org/packages/search?terms=concraft`

moguće morfosintaktičke opisnike. Za hrvatski jezik postoji razvijen analizator zvan HML,[3] ali za potrebe ovog rada nije bio korišten.

Izrada morfosintaktičkog analizatora

Postupak primijenjen za izradu morfosintaktičkog analizatora za hrvatski jezik bio je takav da se s podataka prisutnih na Internetu – konkretno stranici Wiktionary[4] – pobirala (engl. *crawling*) lista riječi hrvatskog jezika sa svojom POS-oznakom. Skoro svaka riječ u indeksu navedene web-stranice posjeduje svoju jedinstvenu stranicu na kojoj su prisutne deklinacijske tablice. Ovisno o tipu riječi svaka bi se stranica pobirala te bi se za određenu deklinaciju generirao morfosintaktički deskriptor. Stranica ima standardizirane tablice te su se uz pomoć dodatne biblioteke – za programski jezik Haskell – HandsomeSoup[5] s lakoćom dohvatile sve potrebne tablice i generirali svi mogući morfosintaktički deskriptori za imenice, glagole, pridjeve, brojeve itd. Stranica sadrži i dio na kojem objašnjavaju kako pobirati podatke te ne postoji ikakva ograničavajuća zaštita na prikupljenim podacima.

Nakon prikupljanja podataka morfosintaktički analizator konstruiran je uz pomoć podatkovne strukture *trie*. Napravljen je *trie* od svih prikupljenih riječi s naglaskom na sufikse. Što znači da je za danu riječ moguće vratiti, ako je ona prisutna u skupu prikupljenih podataka, točan skup mogućih oznaka ili, ako ona nije prisutna, skup mogućih oznaka koje su pridružene uz sufiks određene duljine. U konkretnoj implementaciji, za riječ duljine n gdje je $n > 3$, uzima se sufiks duljine $n - 1$ i $n - 2$.

Morfosintaktički analizator mogao se izgraditi i iz korpusa za hrvatski jezik, ali to bi implicitno poboljšalo rad označivača jer on ovisi o kvaliteti morfosintaktičkog analizatora, a to nije ono što se želi postići (želi se postići mogućnost generalizacije modela na neviđene podatke).

Trenutno analizator nije obradio slučaj živosti imenica i pridjeva i glagolskih pridjeva. Ako se to izvede trebao bi broj nepotpunih skupova – skup bez točnog opisnika – pasti na oko 3000. Sada je broj nepotpunih skupova, ovisno o postavkama, u rasponu od 5000-8000 (od ukupno 88000).

Morfosintaktički analizator izgrađen je od 225296 različitih oblika riječi, sadrži 956 različitih morfosintaktičkih opisnika. Sadrži morfosintaktičke opisnike za imenice, glagole, pridjeve, zamjenice, veznike, prijedloge, priloge, brojeve, čestice, interpunkciju i uzvike. Ne sadrži kratice i imenski ostatak (engl. *residual*). Broj različitih

[3] `http://hml.ffzg.hr/hml/`

[4] `http://en.wiktionary.org/`

[5] `http://hackage.haskell.org/package/HandsomeSoup`

sufiksa koji se mogu formirati do najmanje duljine 1 je 343037, a broj sufiksa do najmanje duljine 3 – koji se koristi u implementaciji – iznosi 342472.

4.1.2. Prilagodba uvjetnih slučajnih polja s ograničenjima

Prilagodba modela uvjetnih slučajnih polja s ograničenjima sastojala se od dva koraka. Bilo je potrebno definirati skup oznaka preko mogućih vrijednosti atributa morfosintaktičkih opisnika i osnovnih POS-oznaka. Primjer definicije može se vidjeti u dodatku A. Za atribute oko kojih su uglate zagrade nije potrebno definirati vrijednost u opisniku, ali kako je standard pozicijski orijentiran postoji mogućnost oznake '-', koja također predstavlja odsutnost vrijednosti atributa. Definicija služi sustavu za kreiranje unutrašnje strukture oznaka koja podržava lako rastavljanje atributa u slojeve i izvlačenje bitnih svojstava. U isto vrijeme provjerava se da su sve predane oznake, u skupu za učenje ili u skupovima predanim od strane morfosintaktičkog analizatora, valjane.

Za završetak prilagodbe potrebno je povezati morfosintaktički analizator s ostatkom modela. Obično su morfosintaktički analizatori odvojeni od označivača i njima treba pristupiti ili preko komunikacijskog kanala ili preko komandne linije. U ovom sustavu morfosintaktički analizator implementiran je direktno u Haskellu te je cijeli *trie* učitan u memoriju koda. Samim pokretanjem prilagodbe analizator je prisutan i spreman za korištenje.

4.2. Vrednovanje uspješnosti

Razvijen označivač u potpunosti je funkcionalan i moguće je koristeći korpus ocijeniti njegovu uspješnost u označavanju vrste riječi. Također, za potrebe ovog rada vrednovat će se i razvijeni morfosintaktički analizator jer uspješnost označivača ovisi o uspješnosti morfosintaktičkog analizatora.

4.2.1. Korpus

Korpus korišten za učenje modela zvan SETimes.HR[6] korišten je za vrednovanje. Označen je revidiranom verzijom 4 standarda MULTEXT-East. Sadrži ukupno 3995 rečenica koje gradi 89128 ručno označenih riječi, interpunkcijski znakova i brojeva.

[6] `http://nlp.ffzg.hr/resources/corpora/setimes-hr/`

4.2.2. Korištene mjere uspješnosti

Za označivača vrste riječi korištena je mjera točnosti koja glasi

$$\frac{b_t}{|T|}$$

gdje je b_t broj točno označenih riječi (potpuno slaganje s morfosintaktičkim opisnikom), a $|T|$ broj riječi u evaluacijskom skupu.

Za evaluaciju morfosintaktičkog analizatora koristi se mjera preciznosti i odziva

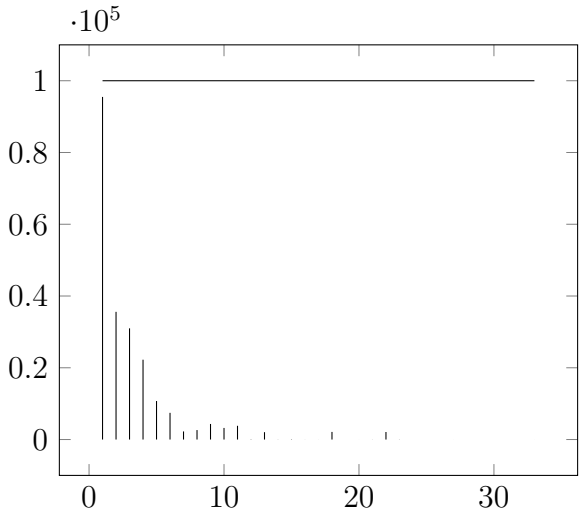
$$P = \frac{G \cap S}{S} \qquad\qquad R = \frac{G \cap S}{G},$$

gdje je G skup bitnih oznaka (u ovom slučaju to je točna oznaka), a S skup dohvaćenih oznaka (u ovom slučaju oznake ponuđene od strane morfosintaktičkog analizatora). Na kraju je računata F_1 mjera

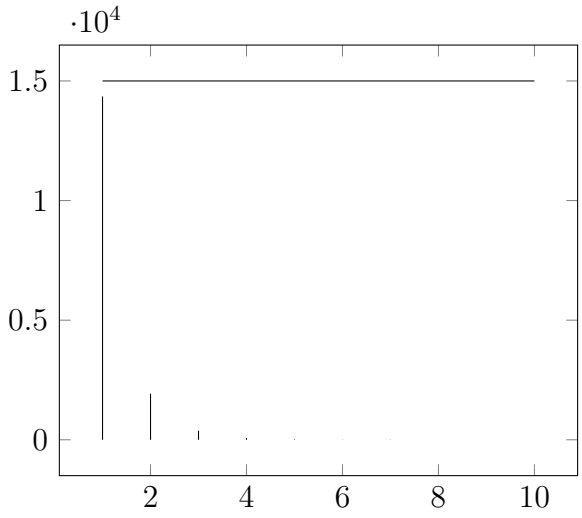
$$F = 2\frac{PR}{P+R}$$

za svaku riječ posebno i usrednjena preko rečenica.

4.2.3. Analiza



(a) Wiktionary



(b) Korpus

Slika 4.1: Prikaz raspodjele višeznačnosti različitih izvora.

Slika 4.1 prikazuje raspodjelu veličine skupa mogućih oznaka za riječi koje su pokupljene s Wiktionaryja, a vidljiva je raspodjela u korpusu. Riječi u korpusu dosežu najveću veličinu višeznačnosti od 10, dok riječi pokupljene s Wiktionaryja 33. Uzimajući samo riječi koje su u rječniku prosječna veličina skupa mogućih oznaka iznosi 3.16 dok je u korpusu samo 1.18. Očito je da će morfosintaktički analizator raditi lošije nego što bi mogao na korpusu.

Za provjeru kako se mijenja višeznačnost s obzirom na parametre analizatora dodana je tablica 4.1 gdje se može vidjeti mana razvijenog morfološkog analizatora. Prikazane su vrijednosti višeznačnosti u korpusu, rječniku koji je korišten za izgradnju

analizatora i proizvedena višeznačnost s primjenom analizatora na korpus (MA Korpus). Veličina skupa mogućih oznaka nije puno udaljena od veličine u početnom rječniku. U drugom slučaju dodaje se i atribut za živost imenice (y ili n). Broj riječi za koje smo dali nepotpun skup se smanjio na 7403, ali se višeznačnost povećala. U trećem slučaju dodaju se i svi mogući nastavci na pridjeve (živost i određenost) da bi se pokrili pridjevi koji imaju te oznake. Sada prosječna višeznačnost postaje toliko velika da se za model razrješavanja višeznačnosti – koji je trigramski model – složenost množi faktorom od oko tisuću (11.73^3), a također dolazi i do troška izvlačenja velikog broja svojstava iz susjednih svojstvenih skupova. Morfološki analizator, ako samo radi s dva najdulja sufiksa i za riječi s velikim početnim slovom, dodaje imenske imenice (atribut p) na kraju daje nepotpun skup za 8837 riječi u korpusu gdje preciznost iznosi $P = 37.18\%$, a R je jednak iznosu 90.08%. Preciznost, zbog postojanja višeznačnosti u jeziku, ne može biti visoka.

Tablica 4.1: Prosječne veličine skupa mogućih oznaka

Podaci	|S|
Korpus	1.1853
Wiktionary	3.1632
MA Korpus 8837	4.0721
MA Korpus 7403	6.8483
MA Korpus 5976	11.7360

Neslaganja i pogreške

Primijećena je i potvrđena pogreška u skupu za učenje. U korpusu ne postoje upitne zamjenice. Prave upitne zamjenice umjesto upitnog tipa imaju neodređeni tip. Promjenom upitnih zamjenica u neodređene, riješen je problem neprepoznavanja oko 4000 zamjenica. Tijekom izgradnje morfosintaktičkog analizatora uzimalo se da akuzativ muškog roda ima svoj neživi i živi oblik, u korpusu SETimes.HR to nije uvijek istina. U korpusu se pojavljuju oznake koje ne mogu biti stvorene navedenom izgradnjom analizatora. Npr. nije moguće naći informaciju je li pridjev glagolski, a u korpusu ima preko 900 glagolskih pridjeva. Određenost pridjeva se također ne slaže s onime što je uzeto s Wiktionaryja, ali dodavanjem svih mogućih kombinacija određenosti i živosti pridjeva, broj pokrivenih riječi povećao se za samo 1400, a uvedena višeznačnost učinila je postupak učenja presporim.

4.2.4. Ostvarena uspješnost

Tablica 4.2: Realistični rezultati uspješnosti

Podaci - MA	POS	MSD	POSn	MSDn
WikiTest - MA7403	95.53	81.31	27.65	14.18
SETimesTest - MA7403	94.38	80.19	37.43	9.49
WikiTest - MA8837	95.58	79.71	23.40	10.64
SETimesTest - MA8837	94.47	78.97	38.55	9.50
WikiTest - MA7403S	94.73	80.72	90.78	64.54
SETimesTest - MA7403S	93.91	80.80	97.77	58.10
WikiTest - MA8837S	94.57	80.77	90.78	65.96
SETimesTest - MA8837S	93.95	78.97	97.76	56.98

HunPos	POS	MSD
WikiTest	94.30	80.46
SETimesTest	97.04	86.77

Tablica 4.3: Optimistični rezultati uspješnosti

Podaci - MA	POS	MSD	POSn	MSDn
WikiTest - MA7403	92.91	82.85	27.66	14.89
SETimesTest - MA7403	94.25	84.76	39.66	11.73
WikiTest - MA8837	92.59	81.94	24.11	12.76
SETimesTest - MA8837	94.29	84.10	39.10	10.61
WikiTest - MA7403S	97.60	86.58	90.78	65.96
SETimesTest - MA7403S	98.78	88.76	97.77	58.65
WikiTest - MA8837S	97.55	84.87	91.48	68.08
SETimesTest - MA8837S	98.73	87.24	97.76	58.10

HunPos	POS	MSD
WikiTest	94.30	80.46
SETimesTest	97.04	86.77

U tablici 4.2 prikazani su realistični, a u tablici 4.3 optimistični rezultati označavanja. Vrednovanje je izvršeno na dva ručno označena testna skupa koji su korišteni u (Agić et al., 2013). Jedan je skup tekstova s Wikipedije, a drugi sa SETimes-a (oba imaju 200 rečenica). Model je naučen na SETimes.HR korpusu i evaluiran na testnim skupovima, a u prvom stupcu tablice naznačen je morfosintaktički analizator koji je korišten u procesu učenja. Imena morfosintaktičkih analizatora koja sadrže slovo S na kraju koriste samo skup od 663 oznake prisutan u SETimes.HR kako bi se dodatno smanjila višeznačnost. Stupci čiji naslov sadrži slovo n na kraju označava rezultat označivača na nepoznatim riječima – riječi za koje morfosintaktički analizator nije izbacio ni jednu oznaku. Može se primijetiti da morfosintaktički analizatori koji koriste samo oznake iz skupa za učenje, a ne svih 956 koje su dohvaćene s Wiktionaryja imaju vrlo dobru uspješnost označavanja nepoznatih riječi. Zadnja tablica prikazuje rezultate HunPos označivača vrednovanog na ista dva navedena skupa s istim skupom definiranih oznaka.

Kod optimističnih rezultata uzeta je pretpostavka kvalitetnog morfosintaktičkog analizatora, gdje će one riječi za koje se izbaci skup svih mogućih oznaka u tom skupu imati točnu oznaku. Oznake riječi za koje je morfosintaktički analizator izbacio prazan skup nisu dodane nego je zadatak određivanja skupa prepušten označivaču. Ako bi se poboljšao rad morfosintaktičkog analizatora, onda bi optimistični rezultati bili najbolji rezultat za označavanje vrste riječi u hrvatskom jeziku.

Moguća poboljšanja

Od presudne bi važnosti bilo korištenje kvalitetnog morfosintaktičkog analizatora (poput HML-a). Bez toga model jednostavno ne može naučiti svojstva neviđenih riječi. U svakom slučaju potrebno je višeznačnost riječi pokušati smanjiti, za trenutni morfosintaktički analizator može se pokušati s izbacivanjem imenica u vokativu jer one često dovode do višeznačnosti, a nisu toliko česte u novinskim tekstovima.

Model ima mnoštvo parametara s kojima se može eksperimentirati. Trenutna implementacija za morfosintaktičko pogađanje i razrješavanje višeznačnosti koristi prefikse i sufikse duljine 1, 2 i 3, a ortografska svojstva promatraju se na dvije prethodne pozicije i jednu naprijed. Atributi morfosintaktičkih opisnika raspodijeljeni su u tri sloja prikazani u dodatku B. Prvi sloj sadrži POS-oznaku i sve tipske oznake vrsta riječi.

Moglo bi se, uz eksperimentiranje s parametrima, testirati ponašanje modela s obzirom na eliminaciju nekog atributa iz slojeva. Raspored i struktura slojeva je nešto što

uvelike može utjecati na uspješnost označivača. Tijekom razvoja sustava nije se obratila prevelika pažnja parametrima stohastičkog gradijentnog spusta, a kako postupak garantira globalni optimum vjerojatno bi bilo dobro pokrenuti postupak učenja s više iteracija.

Trenutni morfosintaktički analizator može se poboljšati spajanjem više riječi u jednu. Ako postoji skup riječi koje neovisno o duljini sufiksa vraćaju uvijek isti skup mogućih oznaka, onda se te riječi mogu spojiti te se samim time smanjuje memorijsko i vremensko opterećenje.

5. Zaključak

Označavanje vrste riječi jedan je od osnovnih zadataka u obradi prirodnog jezika i preduvjet za mnoge druge zadatke. U ovome radu razvio se morfosintaktički analizator i označivač vrste riječi baziran na modelu uvjetnih slučajnih polja s ograničenjima. Uvjetna slučajna polja s ograničenjima pokazala su se vrlo korisnima za poljski jezik, a s obzirom na optimistične rezultate moguće je zaključiti da bi sličnu uspješnost model mogao dostići i za hrvatski jezik.

Morfosintaktičko označavanje na prvi pogled može zvučati kao primitivan problem koji računala danas rješavaju bez poteškoća, ali u ovom radu je pokazano da korištenje naprednih inačica modela, poput uvjetnih slučajnih polja s ograničenjima, ne garantira uvijek uspješan ishod. Višeznačnost u jeziku može drastično usporiti postupak učenja i za bolje rezultate nije dovoljno pristupiti primitivnim rješenjima. Potrebno je, prije nego što se krivnja prebaci na model, provjeriti ostale uzroke slabim rezultatima i u ovom radu pokazala se snažna ovisnost označivača o morfosintaktičkom analizatoru. Pretpostavka da je pravilna oznaka prisutna u skupu svih mogućih daje obećavajuće rezultate i vjerojatno će model raditi dobro uz kvalitetnog pomoćnika.

Uz sam razvoj sustava došlo se i do spoznaja o samome jeziku. Hrvatski jezik pokazuje veliki stupanj višeznačnosti stoga bi budući trud trebao biti usmjeren na njeno smanjivanje.

LITERATURA

Szymon Acedański. A morphosyntactic Brill Tagger for inflectional languages. U *Advances in Natural Language Processing*, stranice 3–14. Springer, 2010.

Željko Agić, Nikola Ljubešić, i Danijela Merkler. Lemmatization and Morphosyntactic Tagging of Croatian and Serbian. U *Proceedings of ACL*, 2013.

Eric Brill. A simple rule-based part-of-speech tagger. U *HLT '91 Proceedings of the workshop on Speech and Natural Language*, 1992.

Eric Brill. Some advances in transformation-based part of speech tagging. *arXiv preprint cmp-lg/9406010*, 1994.

Tomaz Erjavec. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. U *LREC*, 2004.

W Nelson Francis i Henry Kucera. Brown corpus. *Department of Linguistics, Brown University, Providence, Rhode Island*, 1, 1964.

Ralph Grishman. The "label bias" problem: MEMMs and CRFs. `http://cs.nyu.edu/courses/spring13/CSCI-GA.2590-001/LabelBias.pptx`, 2014.

Péter Halácsy, András Kornai, i Csaba Oravecz. HunPos: an open source trigram tagger. U *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, stranice 209–212. Association for Computational Linguistics, 2007.

Daphne Koller i Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

John Lafferty, Andrew McCallum, i Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Andrew McCallum, Dayne Freitag, i Fernando CN Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. U *ICML*, stranice 591–598, 2000.

Hrvoje Peradin. Sintaktička analiza tekstova na hrvatskom jeziku temeljena na gramatici ograničenja. Magistarski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, matematički odsjek, 2012.

Hrvoje Peradin i Jan Šnajder. Towards a Constraint Grammar Based Morphological Tagger for Croatian. U *Text, Speech and Dialogue*, stranice 174–182. Springer, 2012.

Aleksander Pohl i Bartosz Ziółko. A Comparison of Polish Taggers in the Application for Automatic Speech Recognition. 2013.

Adam Radziszewski. Evaluation of lemmatisation accuracy of four Polish taggers. 2013.

Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, i Douglas D Edwards. *Artificial intelligence: A Modern Approach*, svezak 2. Prentice hall Englewood Cliffs, 1995.

Jan Šnajder, Bojana Dalbelo Bašić, i Marko Tadić. Automatic acquisition of inflectional lexica for morphological normalisation. *Information Processing & Management*, 44(5):1720–1731, 2008.

Nataliya Sokolovska, Thomas Lavergne, Olivier Cappé, i François Yvon. Efficient learning of sparse conditional random fields for supervised sequence labeling. *Selected Topics in Signal Processing, IEEE Journal of*, 4(6):953–964, 2010.

Sargur N Srihari. Machine Learning: Generative and Discriminative Models. `http://www.cedar.buffalo.edu/~srihari/CSE574/Discriminative-Generative.pdf`, 2014.

Charles Sutton i Andrew McCallum. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373, 2011.

Martin J Wainwright i Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Jakub Waszczuk. Harnessing the CRF complexity with domain-specific constraints. The case of morphosyntactic tagging of a highly inflected language. U *COLING*, stranice 2789–2804, 2012.

Dodatak A
Definiran skup oznaka

```
[ATTR]

case = n g d a v l i -
gender = m f n -
animate = n y -
number = s p -
person = 1 2 3 -
degree = p c s -

number1 = s p -
gender1 = m f n -

ntype    = c p -

vtype     = m a c -
vform     = n p r f m a e -
vnegative = n y -

atype         = g s p -
adefiniteness = n y -

ptype           = p d i s q r x -
pclitic         = n y -
preftype  = p s -
psyntype = n a -
```

```
rtype   = g r -

scase = g d a l i -

ctype      = c s -
cformation = s c -

mform    = d r l -
mtype    = c o m s -

qtype = z q o r -

xtype = f t p -

[RULE]

N = ntype gender number case [animate]
V = vtype vform [person] [number] [gender] [vnegative]
A = atype degree gender number case [adefiniteness] [animate]
P = ptype person gender number case number1 gender1 pclitic
    preftype psyntype [animate]
R = rtype [degree]
S = scase
C = ctype [cformation]
M = mform [mtype] [gender] [number] [case] [animate]
Q = qtype
I =
Y =
X = [xtype]
Z =
```

Dodatak B
Raspored po slojevima

```
tierDefaults :: [D.Tier]
tierDefaults =
    [tier1, tier2, tier3]
  where
    tier1 = D.Tier True $ S.fromList $
      map (T.pack . (:"type")) "nvaprcmqx"
    tier2 = D.Tier False $ S.fromList
      ["degree", "scase", "case", "person",
       "animate", "number", "number1", "person1"]
    tier3 = D.Tier False $ S.fromList
      ["cformation", "mform", "pclitic",
       "preferenttype", "psyntactictype",
       "adefiniteness", "vform", "vnegative"]
```

Označavanje vrste riječi u hrvatskome jeziku modelom uvjetnih slučajnih polja

Sažetak

Označavanje vrste riječi jedan je od osnovnih zadataka u obradi prirodnog jezika i preduvjet za mnoge druge zadatke. U ovome radu opisana je problematika označavanja vrste riječi i dan je pregled osnovnih i naprednih stohastičkih grafičkih modela te njihova primjena na označavanje vrste riječi visokoflektivnih jezika. Opisan je razvoj morfosintaktičkog označivača temeljen na modelu uvjetnih slučajnih polja s ograničenjima i pažljivo su analizirane sve poteškoće prisutne u razvoju.

Ključne riječi: obrada prirodnog jezika, označavanje vrste riječi, uvjetna slučajna polja, hrvatski jezik

Part-of-Speech Tagging for Croatian using Conditional Random Fields

Abstract

Part-of-speech tagging is one of the fundamental tasks in natural language processing and a prerequisite for many others. In this thesis the problem of POS and morphosyntactic tagging was described. Overview of basic and advanced stochastic graphical models was given and their application to the tagging problem of highly-inflectional languages. Description of the development of the morphosyntactic tagger based on constrained conditional random fields is provided and detailed analysis of all the problems encountered during development.

Keywords: natural language processing, morphosyntactic tagging, conditional random fields, Croatian