Laboratorij za tehnologije znanja (KTLab)
Zavod za elektroniku, mikrolektroniku, računalne i inteligentne sustave
Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu

Interni dokument

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 291

Sintaktički analizator hrvatskoga jezika temeljen na nenadziranom strojnom učenju

Josip Saratlija

Zagreb, lipanj 2011.

SADRŽAJ

1. Uvod

Prirodni jezik je sustav simbola kojima kodiramo informacije te ih prenosimo drugim ljudima, pa kao takav čini okosnicu ljudske komunikacije. Svaki prirodni jezik posjeduje konačan skup pravila koja određuju načine kako se pojedine informacije preslikavaju u dobro oblikovane elemente jezika. Takav skup pravila nazivamo gramatika. Postojanje ovakvog konačnog skupa pravila jest nužno jer da bi pojedinac mogao komunicirati mora biti u mogućnosti razumjeti te proizvesti velik broj rečenica nekog jezika tijekom svog života. Kad gramatika ne bi postojala, bilo bi potrebno da se zasebno nauči svaka rečenica s pripadajućim značenjem odnosno informacijom koju nosi. Gramatika se često dijeli na komponente s obzirom na aspekte jezika koje pravilima određuje. Neke od tih komponenti su:

- *fonetika* – obuhvaća pravila koja određuju strukturu zvukova,
- *morfologija* – obuhvaća pravila koja određuju strukturu riječi,
- *sintaksa* – obuhvaća pravila koja određuju strukturu rečenice,
- *semantika* – obuhvaća preslikavanje informacija u elemente jezika.

Navedene komponente međusobno se uvelike isprepleću te ih nije moguće detaljnije proučavati bez konteksta ostalih komponenti. Sintaksa prirodnog jezika objedinjuje sva gramatička pravila koja određuju načine na koje se rečenice grade od manjih jezičnih elemenata. Osim samih pravila ovaj naziv obuhvaća i znanstvenu disciplinu koja se bavi njihovim izučavanjem. Sintaktička analiza ili parsanje jest postupak određivanja strukture rečenica prirodnog jezika u odnosu na određeni skup pravila odnosno formalnu gramatiku. Gramatička struktura rečenice jest izuzetno vrijedna informacija o samoj rečenici koja uvelike pomaže u obradi prirodnog jezika i to u postupcima kao što su: semantička analiza, ekstrakcija vremenskih oznaka, ekstrakcija trojki subjekt-predikat-objekt, ekstrakcija imenovanih entiteta, strojno prevođenje prirodnog jezika, odgovaranje na pitanja, razumijevanje prirodnog jezika itd. Sustav za sintaktičku analizu prirodnog jezika moguće je izgraditi na više načina. S obzirom na način na koji se dolazi do znanja odnosno pravila potrebnih za sintaktičku analizu, sustave možemo po-

dijeliti u dvije skupine: sustave u kojima su pravila direktno zapisana od strane eksperta te sustave u kojima su pravila naučena na temelju primjera. Sustavi u kojima su pravila direktno zapisana od strane eksperta su u biti ekspertni sustavi namijenjeni sintaktičkoj analizi prirodnog jezika. Izgradnja takvih sustava izuzetno je skupa te složena. Zbog velike ekspresivnosti prirodnog jezika takvi sustavi najčešće nisu namijenjeni obradi cjelokupnog jezika već određenog kontroliranog podskupa. Sustavi u kojima su pravila naučena na temelju primjera najčešće se primjenjuju na cjelokupnom jeziku te je uglavnom riječ o statističkim modelima. Takve sustave s obzirom na primjere na kojima su naučeni možemo podijeliti na sustave naučene nadziranim te sustave naučene nenadziranim učenjem. Kod nadziranog učenja primjeri su rečenice s označenom sintaktičkom strukturom te se takav sustav uči da sa što većom točnošću oponaša način na koji ekspert rečenicama dodjeljuje njihovu sintaktičku strukturu. Prednost ovog pristupa je velika točnost, a najveća je mana potreba za velikim brojem označenih rečenica odnosno bankom stabala (engl. *treebank*). Kod nenadziranog učenja primjeri su samo rečenice. Takav sustav treba na temelju dovoljnog broja rečenica uočiti određene pravilnosti te ih iskoristiti pri sintaktičkoj analizi. Najveća prednost ovakvog sustava jest u tome što on prilikom učenja koristi neoznačane rečenice koje je moguće pribaviti u praktički neograničenim količinama.

U ovom radu napravljen je sintaktički analizator hrvatskog jezika temeljen na nenadziranom učenju. Nenadzirano učenje odabrano je zbog nedostupnosti dovoljno velikog broja označenih rečenica. Statistički model koji je korišten za izgradnju sintaktičkog analizatora zove se *parsanje temeljeno na podacima* (engl. *Data-Oriented Parsing – DOP*). Taj model u nenadziranom pristupu prvo konstruira banku stabala tako da generira sve moguće načine sintaktičkog označavanja dostupnih rečenica, a potom koristi fragmente sintaktičkih struktura tih rečenica za analizu novih rečenica. U svrhu evaluacije, načinjena je manja banka stabala od sto označenih rečenica.

U sljedećem poglavlju objašnjeni su temeljni pojmovi vezani uz sintaktičku analizu prirodnog jezika te je dan pregled nekoliko metoda koje su korištene u nenadziranim pristupima sintaktičkoj analizi. Poglavlje 3 daje detaljan teoretski opis parsanja temeljnog na podacima. Poglavlje 4 daje opis svih algoritama, metoda i postupaka koji su korišteni prilikom implementacije parsanja temeljnog na podacima. U poglavlju 5 su dani rezultati koje je model postigao na načinjenoj banci stabala. Poglavlje 6 zaključuje rad te daje nekoliko smjernica za budući rad na ovom modelu.

2. Sintaktička analiza prirodnog jezika

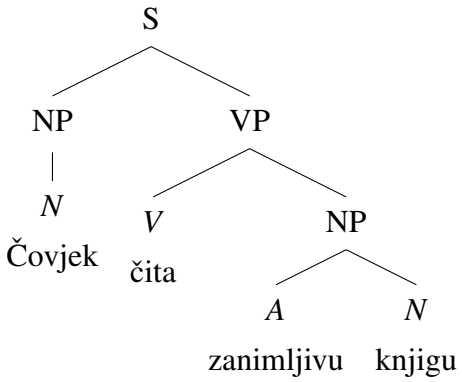
2.1. Vrste strukturnih opisa

Sintaktička struktura neke rečenice prirodnog jezika najčešće se prikazuje u obliku stabla. Određivanje sintaktičkog stabla stoga je osnovni cilj sintaktičke analize. Danas se skoro pa dominantno za prikaz sintaktičke strukture koriste dvije vrste stabala: sastavna stabla (engl. *constituency trees*) te ovisnosna stabla (engl. *dependency trees*).

2.1.1. Sastavna struktura

Sastavna struktura (engl. *constituency structure*) označava strukturu rečenice opisanu sastavnim stablom. Gramatika koja sintaksu rečenice prikazuje sastavnom strukturom naziva se sastavna gramatika (engl. *constituency grammar*). Sastavna gramatika jest kao formalizam za opis sintakse prirodnog jezika najviše afirmirana teorijama kao što su GB (engl. *Government and Binding theory*), (Chomsky, 1986), LFG (engl. *Lexical Functional Grammar*), (Kaplan i Bresnan, 1982) te HPSG (engl. *Head-driven Phrase Structure Grammar*), (Pollard i Sag, 1994). Temeljna ideja na kojoj se zasnivaju sastavne gramatike jest da se rečenice prirodnog jezika sastoje od segmenata (engl. *constituent*) koji predstavljaju lingvistički koherentne jedinice. U kontekstu sintaktičke analize prirodnog jezika ti se segmenti nazivaju sintaktičke jedinke. Svaka sintaktička jedinka ima pridijeljenu sintaktičku kategoriju. Sintaktičke kategorije koje se mogu rastaviti na manje sintaktičke kategorije nazivamo frazalne kategorije, a one koje su nedjeljive nazivamo leksičke kategorije.

Sastavno stablo predstavlja grafički prikaz hijerarhije sintaktičkih jedinki. Fraze su unutarnji čvorovi stabla koji se u kontekstu sintaktičke analize zovu nezavršni čvorovi, a njihova kategorija naziva se nezavršni znak. Leksičke jedinke su listovi sastavnog stabla koji se u kontekstu sintaktičke analize zovu završni čvorovi, a njihova kategorija naziva se završni znak. Na sastavnom stablu prikazanom na slici 2.1 možemo identificirati dvije imenične fraze označene s NP (engl. *Noun Phrase*) koje obuhvaćaju



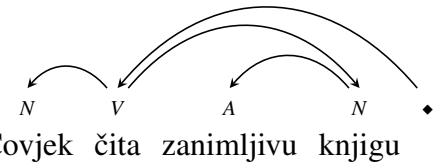
Slika 2.1: Sastavno stablo za rečenicu: *"Čovjek čita zanimljivu knjigu."*

segmente "Čovjek" odnosno "zanimljivu knjigu", potom glagolsku frazu označenu s VP (engl. *Verb Phrase*) koja obuhvaća segment "čita zanimljivu knjigu" te frazu označenu sa S (engl. *Sentence*) koja obuhvaća cijelu rečenicu. Također možemo uočiti četiri leksičke jedinke koje odgovaraju riječima iz rečenice. Za leksičke kategorije najčešće se uzimaju oznake vrste riječi (engl. *part-of-speech*), no moguće je u njih uključiti i same riječi. Taj postupak naziva se leksikalizacija.

2.1.2. Ovisnosna struktura

Ovisnosna struktura (engl. *dependency structure*) označava strukturu rečenice opisanu ovisnosnim stablom. Gramatika koja sintaksu rečenice prikazuje ovisnosnom strukturom naziva se ovisnosna gramatika (engl. *dependency grammar*). Takav opis sintaktičke strukture rečenice idejno je začet u (Tesnière, 1959), no mnogi lingvisti smatraju da su slični manje formalizirani opisi bili prisutni još od Srednjeg vijeka. Temeljna ideja na kojoj se temelje ovisnosne gramatike jest pretpostavka da u rečenici sve riječi osim jedne ovise o nekoj drugoj. Riječ koja ne ovisi ni o kojoj drugoj zove se korijen rečenice. Svaka ovisnost u rečenici jest usmjerena te se sastoji od glave (engl. *head, governor*) te ovisnog člana (engl. *complement, dependent*). Ovisnosti predstavljaju istovremenu sintaktičku te semantičku vezu između riječi. Svaka ovisnost nosi dio informacije koje je sadržana u rečenici, a ostvaruje se na način da ovisni član specificira općenit koncept koji predstavlja glava. Primjerice glagol "čitati" u rečenici prikazanoj na slici 2.2 specificiraju dva ovisna člana čije su gramatičke funkcije subjekt te objekt. Svaki od ovisnih članova može biti glava nekim drugim ovisnostima koje specificiraju njihova značenja.

Korijen rečenice je najčešće glagol odnosno predikat jer glagoli kao vrsta riječi najviše zahtijevaju druge riječi odnosno argumente da specificiraju njihovo značenje. To



Slika 2.2: Ovisnosno stablo za rečenicu: *"Čovjek čita zanimljivu knjigu."*

svojstvo glagola se naziva *valencija*. Većina informacije koju rečenica sadrži nalazi se u ovisnostima kojima je glava korijen rečenice.

2.2. Formalna gramatika

Formalna gramatika bilo je kakav matematički precizno definiran skup pravila kojim je definirana neka komponenta jezika, najčešće sintaksa. Primjeri gramatika koje nisu formalne bili bi perskriptivna gramatika kojom se ustanovljavaju određene norme u jeziku te deskriptivna gramatika kojom se opisuje sama upotreba jezika. Formalne gramatike se s obzirom na način korištenja dijele na generativne te analitičke. Generativna gramatika predstavlja generalizaciju algoritma kojim se generiraju ispravne rečenice nekog jezika, dok analitička gramatika predstavlja skup pravila na temelju kojeg se za danu rečenicu može reći pripada li ona određenom jeziku. Pojam formalne gramatike najčešće se poistovjećuje s matematičkim modelom predloženim u (Chomsky, 1956). Taj model gramatiku definira kao uređenu četvorku $< V, T, R, S >$ pri čemu je:

- V – konačan skup nezavršnih znakova;
- T – konačan skup završnih znakova;
- R – konačan skup pravila $\alpha \longrightarrow \beta$, gdje su α i β proizvoljni nizovi završnih i nezavršnih znakova;
- S – početni nezavršni znak.

Za pravila ove gramatike još se koristi i naziv produkcije. Elementi jezika dobivaju se na način da se počevši od početnog nezavršnog znaka primjenjuju produkcije gramatike sve dok dobiveni niz ne bude sadržavao isključivo završne znakove. Slijed primjena produkcija zove se derivacija. Uvođenjem određenih ograničenja na produkcije gramatike dobivaju se klase jezika različite ekspresivnosti. Tablica 2.1 prikazuje hijerarhiju klasa jezika na temelju restrikcija u produkcijama opisanog modela formalne gramatike.

Tablica 2.1: Chomskyjeva hijerarhija jezika

Gramatika	Klasa jezika	Oblik produkcija
Tip 0	Rekurzivno prebrojivi	$\alpha \longrightarrow \beta$
Tip 1	Konteksno ovisni	$\alpha A \beta \longrightarrow \alpha \gamma \beta$
Tip 2	Konteksno neovisni	$A \longrightarrow \gamma$
Tip 3	Regularni	$A \longrightarrow a$ ili $A \longrightarrow aB$

$$\alpha, \beta, \gamma \in (V \cup T)^*$$
$$A, B \in V$$
$$a \in T$$

Od navedenih klasa u sintaktičkoj analizi prirodnog jezika najviše se koristi klasa kontekstno neovisnih jezika. Kontekstno neovisna gramatika te njene brojne modifikacije jedan su od temeljnih načina zapisa pravila odnosno znanja o sintaktičkoj strukturi nekog jezika. Jedna od modifikacija koja se uvelike koristi u statističkim modelima jest vjerojatnosna konteksno neovisna gramatika (engl. *Probabilistic Context-Free Grammar - PCFG*) koja dodatno definira vjerojatnosnu funkciju $P : R \longrightarrow [0, 1]$, pri čemu vrijedi

$$\forall A \in V : \sum_{r \in R, left(r) = A} P(r) = 1. \tag{2.1}$$

2.3. Nenadzirani pristupi sintaktičkoj analizi

Postupci sintaktičke analize prirodnog jezika već su stoljećima bili predmetom istraživanja u lingvistici i filozofiji. Razvojem računala ta problematika polako ulazi i u računarsku znanost. Posljednjih dvadesetak godina dolazi do naglog intenziviranja te problematike najviše zbog nastojanja da se unaprijedi interakcija čovjeka i računala. To unaprjeđivanje u pravilu uključuje i određenu količinu razumijevanja prirodnog jezika od strane računala čemu je sintaktička analiza neizostavan dio.

Postupke sintaktičke analize možemo podijeliti s obzirom na različite kriterije. S obzirom na sintaktičku strukturu postoje sastavni i ovisnosni sintaktički analizatori, s obzirom na način pribavljanja znanja (engl. *knowledge acquisition*) postoje analizatori s eksplicitno definiranim znanjem te strojno naučeni analizatori, a strojno naučene analizatore možemo podijeliti na nadzirano i nenadzirano strojno naučene analizatore. U centru znanstvenog interesa dugo su vrijeme bili sastavni analizatori temeljeni na

nadziranom učenju koji su uglavnom bili namijenjeni engleskom jeziku. U novije vrijeme dolazi do sve više afirmacije ostalih jezika koja kao posljedicu ima povećavanje interesa za nenadzirano učenje zbog nedostatka označenih korpusa – banaka stabala (engl. *treebank*) te ovisnosnu strukturu zbog nemogućnosti modeliranja sastavnom strukturom nekih jezičnih fenomena prisutnih u različitim jezicima. Pošto je tema ovog rada sintaktička analiza temeljena na nenadziranom učenju, u ovom će poglavlju biti obrađeni samo takvi pristupi.

2.3.1. Pristupi temeljeni na maksimizaciji očekivanja

Jedna od temeljnih metoda nenadziranog učenja koja je našla široku primjenu i u sintaktičkoj analizi temeljnoj na nenadziranom učenju jest metoda maksimizacije očekivanja (engl. *Expectation Maximization – EM*) (Dempster et al., 1977). Metoda maksimizacije očekivanja koristi se kada želimo izračunati skup parametara modela Θ koji opisuju neku skrivenu razdiobu vjerojatnosti pri čemu je moguće promatrati samo dio atributa skupa raspoloživih m instanci $\{x_1, \ldots, x_m\}$, dok ostale atribute koje nazivamo latentne varijable $\{z_1, \ldots, z_m\}$ nije moguće direktno odrediti. Funkcija izglednosti (engl. *likelihood function*) definira se kao:

$$L(X; \Theta) := f(x_1; \Theta) \cdots f(x_m; \Theta) \tag{2.2}$$

pri čemu $f(x; \Theta)$ označava funkciju gustoće slučajne varijable X kojom modeliramo vidljive atribute. Za procjenu parametara Θ uzimamo onu vrijednost Θ' za koju funkcija izglednosti poprima globalni maksimum. Globalni maksimum u većini slučajeva nije moguće naći u razumnom vremenu pa metoda maksimizacije očekivanja koristi iterativan algoritam koji konvergira nekom lokalnom optimumu.

Algoritam "unutar-izvana"

Metoda maksimizacije očekivanja temelj je za izgradnju algoritma "unutar-izvana" (engl. *inside-ouside algorithm*) (Baker, 1982). Algoritam "unutar-izvana" je iterativan algoritam koji podešava vjerojatnosti produkcija vjerojatnosne kontekstno neovisne gramatike (engl. *Probabilistic Contex-Free Grammar – PCFG*) na način da izglednost korpusa bude što veća. Sastavno stablo prikazano na slici 2.1 parsano je sljedećim skupom produkcija:

$$S \longrightarrow NP\ VP$$

$$VP \longrightarrow V\ NP$$

$$NP \longrightarrow N$$

$$NP \longrightarrow A\ N$$

Vjerojatnost produkcije $A \longrightarrow \alpha$ označava se s $p(A \longrightarrow \alpha; \Theta)$, pri čemu je Θ skup parametara modela te vrijedi:

$$\sum_{\alpha} p(A \longrightarrow \alpha; \Theta) = 1 \tag{2.3}$$

Ako sastavno stablo prikazano na slici 2.1 označimo s t_1, a pripadajuću rečenicu sa s_1, vjerojatnost parsanja te rečenice takvim stablom uz skup parametara modela Θ možemo izraziti kao:

$$\begin{aligned} P(s_1, t_1; \Theta) \;\; = \;\; & p(S \longrightarrow NP\ VP; \Theta) \cdot p(VP \longrightarrow V\ NP; \Theta) \cdot \\ & p(NP \longrightarrow N; \Theta) \cdot p(NP \longrightarrow A\ N; \Theta) \end{aligned}$$

Vjerojatnost rečenice s_i jednaka je sumi vjerojatnosti svih sastavnih stabala kojima se ta rečenica može parsati na temelju definiranih produkcija odnosno:

$$P(s_i; \Theta) = \sum_{j} P(s_i, t_j; \Theta) \tag{2.4}$$

Izglednost korpusa S definira se kao umnožak vjerojatnosti svih njegovih rečenica odnosno:

$$L(S; \Theta) := P(s_1; \Theta) \cdots P(s_n; \Theta) \tag{2.5}$$

Pošto je skup rečenica nekog jezika prebrojiv, korpus je diskretna slučajna varijabla pa funkciju gustoće iz formule (2.2) možemo zamijeniti diskretnom vjerojatnosti iz čega slijedi ekvivalencija izraza (2.2) i (2.5). Metoda maksimizacije očekivanja je u algoritmu "unutar-izvana" primijenjena na način da rečenice nekog korpusa predstavljaju vidljive atribute dok je njihova sintaktička struktura uvjetovana skrivenim parametrima. Skriveni parametri svoj utjecaj ostvaruju preko vjerojatnosti produkcija. Iterativno izračunavanje vjerojatnosti ostvaruje se pomoću tzv. unutrašnjih i vanjskih vjerojatnosti čiji daljnji izvod ovdje neće biti naveden.

Najraniji pokušaji nenadziranog učenja izgradnje sintaktičke strukture rečenica pomoću algoritma "unutar-izvana" bili su obeshrabrujući. U (Lari i Young, 1990) pokušalo se korištenjem algoritma "unutar-izvana" rekonstruirati gramatiku koja bi parsala jezik koji se sastoji od nizova znakova koji su palindromi. Njihovi rezultati su pokazali da je algoritam iznimno osjetljiv na početnu raspodjelu vjerojatnosti produkcija

te redundancije u samoj gramatici. U (Amaya et al., 1999) algoritam "unutar-izvana" testiran je na Pennovoj[1] banci stabala gdje je dao nešto bolje rezultate.

CCM

U (Klein i Manning, 2002) predložen je generativni sastavni kontekstni model (engl. *Constituent-Contex Model – CCM*) koji se zasniva na dvama kriterijima za identifikaciju sintaktičkih jedinki predloženima u (Radford, 1988):

1. *Vanjska distribucija*: Sintaktička jedinka je niz riječi koje se pojavljuju na različitim strukturnim pozicijama (unutar većih sintaktičkih jedinki);

2. *Zamjenjivost*: Sintaktička jedinka je niz riječi koje mogu biti zamijenjene jednostavnim varijantama tog niza.

Prvi kriterij identificira sintaktičku jedinku kao niz riječi koji se u nepromijenjenom obliku može pojavljivati na različitim mjestima unutar većih sintaktičkih jedinki, dok drugi kriterij govori o tome kada dvije različite sintaktičke jedinke imaju istu sintaktičku kategoriju. Model CCM identificira dvije vrste rečeničnih podnizova: one koji predstavljaju sintaktičke jedinke (engl. *constituents*) te one koji to nisu (engl. *distituents*). Sastavna struktura prikazuje se matricom B koja za svaki podniz α sadrži istinosnu vrijednost je li to sintaktička jedinka ili ne. Svaki podniz α nalazi se u kontekstu x. Primjerice podniz *A N* sa slike 2.1 nalazi se u kontekstu *V*-$\diamond$, gdje $\diamond$ označava kraj rečenice.

Vjerojatnost da rečenica s ima sastavnu strukturu B uz skup parametara modela Θ jest:

$$P(s, B; \Theta) = P(B)P(s|B; \Theta) \tag{2.6}$$

pri čemu je

$$\begin{aligned} P(s|B;\Theta) &= \prod_{<i,j> \in \, podnizovi(s)} P(\alpha_{i,j}, x_{i,j}|B_{i,j};\Theta) \\ &= \prod_{<i,j> \in \, podnizovi(s)} P(\alpha_{i,j}|B_{i,j};\Theta)P(x_{i,j}|B_{i,j};\Theta) \end{aligned}$$

Marginalna vjerojatnost rečenice tada je:

[1] `http://www.cis.upenn.edu/~treebank/`

$$P(s;\Theta) = \sum_{B} P(B)P(s|B;\Theta) \tag{2.7}$$

Na ovako definirane vjerojatnosti metodom maksimizacije očekivanja podese se vjerojatnosti $P(\alpha, B)$ i $P(x, B)$, dok je $P(B)$ namješteno tako da su sva binarna stabla jednako vjerojatna, a sve ostale sastavne strukture imaju vjerojatnost 0.

DMV

U (Klein, 2004) predstavljen je model indukcije ovisnosne gramatike nazvan ovisnosni model s valencijom (engl. *Dependency Model with Valence – DMV*). Model započinje od korijena rečenice koji predstavlja početnu glavu. Svaka glava prvo generira niz ne-$STOP$ ovisnih članova s jedne strane, pa zatim $STOP$ na toj strani, pa niz ne-$STOP$ ovisnih članova s druge strane te zatim $STOP$ na toj strani. Odluka generira li se znak $STOP$ određena je distribucijom vjerojatnosti $P_{STOP}(STOP|h, dir, adj)$ gdje je h glava, dir strana (l ili r) te adj istinosna vrijednost koja predstavlja je li na toj strani već generiran ovisni član. Ako se na toj strani ne generira $STOP$ onda se ovisni član a bira s distribucijom vjerojatnosti $P_{CHOOSE}(a|h, dir)$.

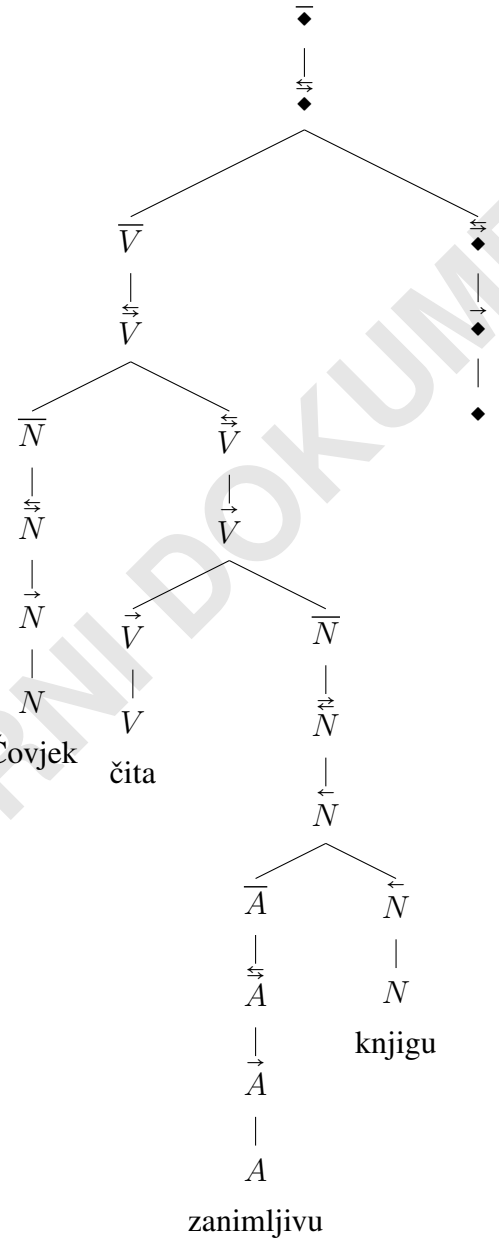
Formalno, za ovisnosnu strukturu D, gdje svaka glava h ima s lijeve strane ovisne članove $deps_D(h, l)$, te s desne strane ovisne članove $deps_D(h, r)$, sljedeća rekurzija definira vjerojatnost podstabla $D(h)$ čiji je korijen u h:

$$P(D(h)) = \prod_{dir\in\{l,r\}} \left(\prod_{a\in deps_D(h,dir)} P_{STOP}(\neg STOP|h, dir, adj) P_{CHOOSE}(a|h, dir) P(D(a)) \right) \cdot P_{STOP}(STOP|h, dir, adj)$$

Izgradnja ovisnosnog stabla ovim modelom može se opisati skupom produkcija kontekstno neovisne gramatike. Za svaki završni znak $w \in W \cup \{\blacklozenge\}$ postoje sljedeći nezavršni znakovi: $\vec{w}$, $\overleftarrow{w}$, $\vec{\overleftarrow{w}}$, $\overleftrightarrow{w}$, $\overline{w}$. Produkcije su:

Izbor glave (nadesno)	$\vec{w} \longrightarrow w$
Izbor glave (nalijevo)	$\overleftarrow{w} \longrightarrow w$
Nadesno desno pridjeljivanje	$\vec{h} \longrightarrow \vec{h}\,\overline{a}$
Nadesno desni stop	$\overleftrightarrow{h} \longrightarrow \vec{h}$
Nadesno lijevo pridjeljivanje	$\overleftrightarrow{h} \longrightarrow \overline{a}\,\overleftrightarrow{h}$
Nadesno zaključavanje	$\overline{h} \longrightarrow \overleftrightarrow{h}$
Nalijevo lijevo pridjeljivanje	$\overleftarrow{h} \longrightarrow \overline{a}\,\overleftarrow{h}$
Nalijevo lijevi stop	$\overrightarrow{\overleftarrow{h}} \longrightarrow \overleftarrow{h}$
Nalijevo desno pridjeljivanje	$\overrightarrow{\overleftarrow{h}} \longrightarrow \overrightarrow{\overleftarrow{h}}\,\overline{a}$
Nalijevo zaključavanje	$\overline{h} \longrightarrow \overrightarrow{\overleftarrow{h}}$

Za svaku rečenicu uzima se da završava znakom $\blacklozenge$, a početni nezavršni znak gramatike je $\overline{\blacklozenge}$. Slika 2.3 prikazuje sintaktičko stablo za potpuno artikulirani DMV-model ovisnosnog stabla prikazanog na slici 2.2.



Slika 2.3: Sintaktičko stablo za potpuno artikulirani DMV-model rečenice *"Čovjek čita zanimljivu knjigu."*

3. Parsanje temeljeno na podacima

Parsanje temeljeno na podacima (engl. *Data-Oriented Parsing – DOP*) statistički je model za sintaktičku analizu prirodnog jezika predložen u (Scha, 1990), a formaliziran u (Bod, 1992). Osnovni model DOP-a temeljen je na nadziranom učenju te namijenjen parsanju rečenica u sastavnoj strukturi. Svaki model namijenjen automatskoj akviziciji gramatike prirodnog jezika mora imati definirane sljedeće elemente:

1. *Prikaz znanja* – gramatički formalizam, način zapisa pravila koja uvjetuju strukturu prirodnog jezika;
2. *Akvizicija znanja* – dobivanje pravila u zadanom formalizmu iz dostupnih primjera;
3. *Parsanje* – definicija (najbolje) sintaktičke strukture rečenice na temelju dostupnih pravila te sam algoritam za njeno određivanje.

Statistički modeli dodatno unutar akvizicije znanja imaju i:

2'. *Podešavanje vjerojatnosti pravila* – određivanje i podešavanje vjerojatnosti pravila na temelju dostupnih primjera.

U nastavku će biti prikazano na koji način DOP definira elemente modela automatske akvizicije gramatike prirodnog jezika.

3.1. Prikaz znanja

DOP za prikaz znanja o sintaktičkoj strukturi prirodnog jezika koristi gramatički formalizam nazvan *vjerojatnosna gramatika supstitucije stabala* (engl. *Stochastic Tree-Substitution Grammar – STSG*). STSG je uređena petorka $< V, T, R, S, P >$ pri čemu je:

V – konačan skup nezavršnih znakova;

T – konačan skup završnih znakova;

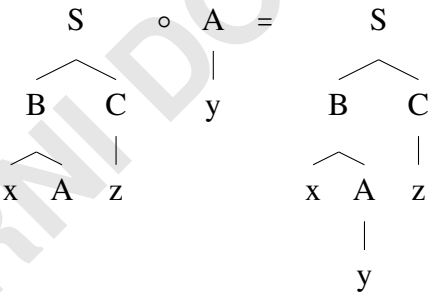
R – konačan skup pravila pri čemu je svako pravilo stablo čiji su unutrašnji čvorovi nezavršni znakovi, a listovi nezavršni ili završni znakovi;

S – početni nezavršni znak;

P - vjerojatnosna funkcija $P : R \longrightarrow [0, 1]$, pri čemu vrijedi

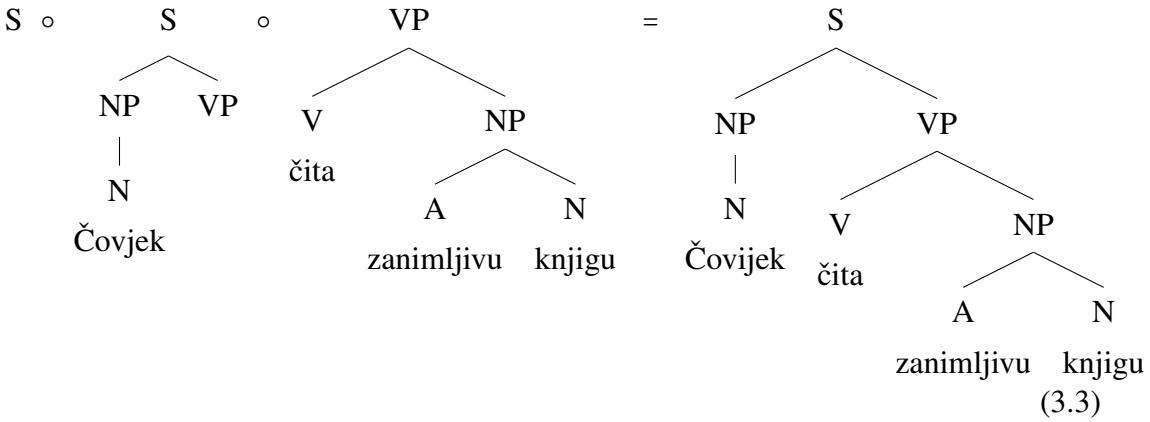
$$\forall A \in V : \sum_{r \in R, root(r)=A} P(r) = 1. \tag{3.1}$$

STSG-derivacija izvodi se kombiniranjem stabala pomoću supstitucijskog operatora $\circ$. Svako stablo kojemu je barem jedan list nezavršni znak zove se parcijalno stablo. Ako je r_1 parcijalno stablo čiji je najljeviji nezavršni znak A, a r_2 stablo s korijenom A, onda je rezultat operacije $(r_1 \circ r_2)$ stablo nastalo nadodavanjem stabla r_2 stablu r_1 na mjestu nezavršnog znaka A, (3.2).



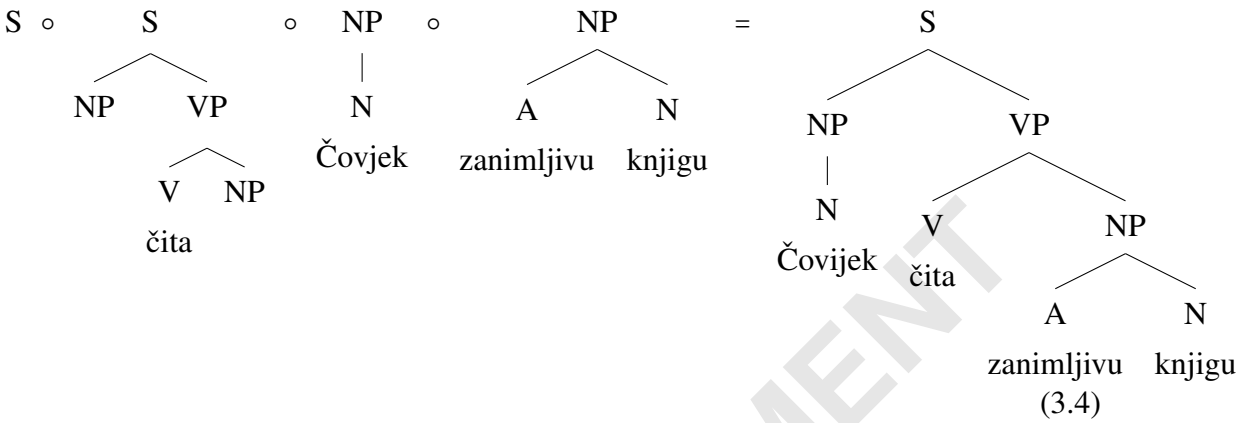
(3.2)

Operator $\circ$ je lijevo asocijativan pa niz uzastopnih supstitucija $(((r_1 \circ r_2) \circ r_3) \circ \cdots \circ r_n)$ možemo kraće zapisati $r_1 \circ r_2 \circ r_3 \circ \cdots \circ r_n$. Za razliku od formalizma CFG, isto sintaktičko stablo može se proizvesti na temelju više različitih derivacija[1], (3.3), (3.4).



(3.3)

[1]Podrazumijeva se da su CFG-derivacije determinirane odnosno uvedena je neka strategija odabira sljedećeg nezavršnog znaka za zamjenu, npr. zamjena najljevijeg nezavršnog znaka



(3.4)

STSG definira sljedeće vjerojatnosti:

Vjerojatnost derivacije (engl. derivation probability*):* Za derivaciju $d = r_1 \circ r_2 \circ r_3 \circ \cdots \circ r_n$ definira se vjerojatnost

$$P(d) = \prod_{i=1}^{n} P(r_i). \tag{3.5}$$

Izračun vjerojatnosti pretpostavlja da su primjene supstitucije nezavisne. Definicija vjerojatnosne funkcije $P : R \longrightarrow [0, 1]$ osigurava da će suma vjerojatnosti svih mogućih derivacija biti 1.

Vjerojatnost sintaktičkog stabla (engl. parse probability*):* Neka je D_t skup svih derivacija koje proizvode stablo t. Vjerojatnost stabla t je

$$P(t) = \sum_{d \epsilon D_t} P(d). \tag{3.6}$$

Vjerojatnost rečenice (engl. utterance probablility*):* Neka je T_s skup svih sintaktičkih stabala čiji je niz završnih znakova jednak nizu znakova izvedenog iz rečenice s. Tada je vjerojatnost rečenice s dana izrazom

$$P(s) = \sum_{tr \epsilon T_s} P(t). \tag{3.7}$$

3.2. Akvizicija znanja

3.2.1. Pristupi akviziciji znanja

DOP-model omogućava akviziciju znanja o sintaktičkoj strukturi prirodnog jezika i u nadziranom i u nenadziranom obliku. Primjeri potrebni za nadzirani pristup su rečenice s označenim sintaktičkim stablima, dok nenadzirani pristup zahtjeva samo rečenice.

Oba pristupa koriste metodu ekstrakcije svih podstabala (engl. *all-subtrees approach*). U nastavku oba će oblika akvizicije znanja biti detaljnije objašnjeni.

Nadzirani pristup

Nadzirani pristup je originalni model DOP-a pa se najčešće pod nazivom DOP on podrazumijeva, iako je pojavom nenadziranog pristupa preimenovan u S-DOP (engl. *Supervised Data-Oriented Parsing*). Akvizicija znanja odnosno pravila STSG-formalizma vrlo je jednostavna, a sastoji se od toga da svako različito podstablo iz skupa svih stabala neke banke stabala predstavlja jedno pravilo, a njegova vjerojatnost proporcionalna je broju njegovih pojavljivanja.

$$P(r) = \frac{|r|}{\sum_{r':root(r')=root(r)} |r'|} \tag{3.8}$$

Nenadzirani pristup

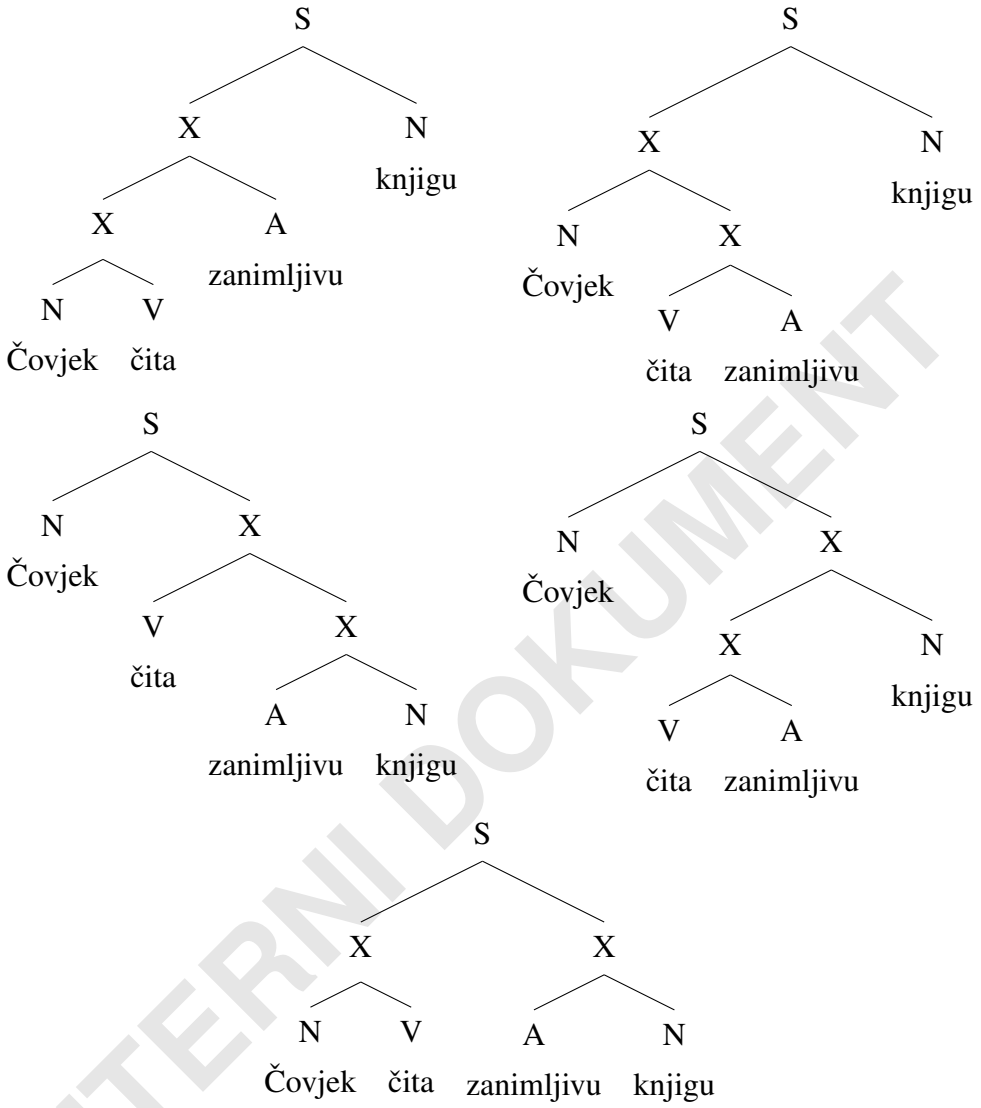
DOP-model koji koristi nenadzirani pristup akviziciji znanja zove se U-DOP (engl. *Unsupervised Data-Oriented Parsing*). Model U-DOP pri akviziciji znanja koristi rečenice bez označene sintaktičke strukture. Zbog toga što ne zna kakvu sintaktičku strukturu rečenica ima, U-DOP pretpostavlja da su sve moguće sintaktičke strukture te rečenice jednako vjerojatne. Ipak, U-DOP se ograničava na binarna stabla te zanemaruje razlikovanje frazalnih kategorija odnosno označava sve nezavršne znakove s X, osim početnog nezavršnog znaka koji ostaje S. U-DOP se stoga orijentira na identifikaciju elemenata rečenice bez njihovog kategoriziranja. Slika 3.1 prikazuje sva binarna stabla za rečenicu *"Čovjek čita zanimljivu knigu"*. Broj binarnih stabala rečenice od n riječi jednak je n-tom Catalanovom broju te vrijedi:

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!n!} \tag{3.9}$$

Asimptotski Catalanovi brojevi rastu kao:

$$C_n \sim \frac{4^n}{n^{\frac{3}{2}}\sqrt{\pi}} \tag{3.10}$$

Nenadzirani pristup dakle prvo načini banku stabala u kojoj se nalaze sva moguća binarna stabla za sve rečenice, a zatim nastavi s ekstrakcijom svih podstabala na način identičan nadziranom pristupu. Iako na takav način načinjena banka stabala ima mnoštvo pogrešnih sintaktičkih struktura rečenica, vrlo vjerojatno i puno više nego ispravnih, očekuje se da će model pri parsanju ipak preferirati ispravna sintaktička stabla.



Slika 3.1: Sva binarna stabla za rečenicu *"Čovjek čita zanimljivu knjigu"*

To se temelji na hipotezi da bi ispravne sintaktičke strukture trebale dijeliti ista svojstva jer su u stvarnosti posljedica istog skupa pravila[2] koji definira neki prirodni jezik. Na temelju statistički reprezentativnog skupa primjera onaj skup pravila u odabranom formalizmu koji proizvodi ispravna sintaktička stabla trebao bi "isplivati".

3.2.2. Podešavanje vjerojatnosti pravila

Podešavanje parametara modela odnosno vjerojatnosti pojedinih pravila jedan je od najvećih problema DOP-a. Do sada su konstruirana tri različita procjenitelja koji su

[2]Teoretski formalizam kojim bi se mogla izraziti pravila kojima bi se u potpunosti opisali svi prirodni jezici zove se *univerzalna gramatika*, (Chomsky, 1957)

pristrani na način da favoriziraju manja odnosno veća stabla ili dolazi do prenaučenosti (engl. *overfitting*). U nastavku su objašnjeni do sada konstruirani procjenitelji.

Procjenitelj relativnom frekvencijom

Izraz 3.8 predstavlja prvi procjelitelj parametara DOP-modela, a nalaže da je vjerojatnost pravila jednaka relativnoj frekvenciji podstabla koje to pravilo reprezentira među svim podstablima s tim korjenom u banci stabala. U (Johnson, 1998) je pokazano da je procjenitelj relativnom frekvencijom pristran[3] i nekonzistentan[4]. Pristranost nije nužno loša karakteristika procjenitelja, ponekad je bolje da je ona veća jer time se smanjuje varijanca što u konačnici može smanjiti ukupnu grešku procjenitelja. Nekonzistentnost procjenitelja se, ipak, općenito smatra lošom.

Bonnemov procjenitelj

U (Bonnema et al., 1999) predložen je nov procjenitelj parametara modela. Taj procjenitelj promatra svako stablo iz banke stabala kao skup svih različitih derivacija kojima ono može biti sastavljeno. Pošto se ne zna koja je od tih derivacija lingvistički najopravdanija, procjenitelj pretpostavlja njihovu uniformnu razdiobu. Za vjerojatnost pojedinog pravila vrijedi:

$$P'(r) = 2^{-N(r)} P(r) \tag{3.11}$$

gdje je $N(r)$ broj nezavršnih znakova u podstablu koje pravilo predstavlja ne računajući korijen ni nezavršne znakove koji su listovi podstabla, a $P(r)$ je vjerojatnost izračunata prema (3.8).

Procjenitelj temeljen na maksimizaciji očekivanja

U (Bod, 2000a) predložen je način izračuna parametara temeljen na maksimizaciji očekivanja. Ovakav način podešavanja parametara je nepristran te konzistentan. Inicijalizacija koja predstavlja jedan od temeljnih problema maksimizacije očekivanja rješena je na način da su početne vjerojatnosti inicijalizirane na vrijednosti relativne frekvencije odnosno prema (3.8). Ipak, kao velika mana ovog pristupa pokazala se pojava prenaučenosti odnosno model dobro parsira ona stabla koja postoje u banci stabala na

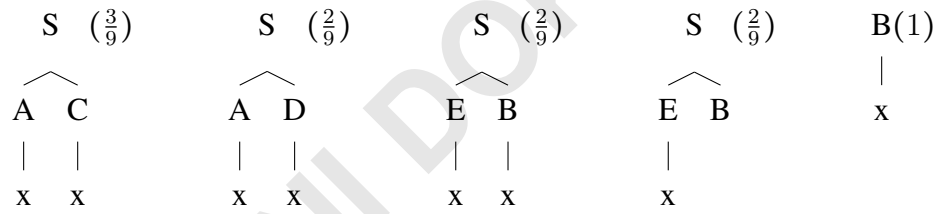
[3]Za procjenitelja Θ parametra ϑ kažemo da je *nepristran* ako je $E(\Theta) = \vartheta$, gdje $E(\cdot)$ označava očekivanje procjelitelja.

[4]Za procjenitelja $\Theta_n = \Theta(X_1, X_2, \ldots, X_n)$ parametra ϑ kažemo da je *konzistentan* ako za svaki $\epsilon > 0$ slučajna varijabla Θ_n konvergira prema ϑ po vjerojatnosti, odnosno $\lim_{n\to\infty} P(|\Theta_n - \vartheta| < \epsilon) \to 1$

kojoj je učen, dok mu je mogućnost generalizacije na neviđenim primjerima relativno mala.

3.3. Parsanje

Parsanje je postupak određivanja optimalne sintaktičke strukture neke rečenice na temelju stečenog znanja koje je prikazano u odabranom formalizmu. Parsanje obuhvaća dva povezana elementa: definicija optimalne sintaktičke strukture te algoritam za njeno određivanje. Definicija optimalne sintaktičke strukture najčešće se prikazuje u obliku kriterijske funkcije koja svakom sintaktičkom stablu daje odgovarajuću ocjenu, a za optimalno stablo uzima se ono s najboljom ocjenom. U nastavku su objašnjene često korištene definicije optimalne sintaktičke strukture, a korišteni algoritam je objašnjen u poglavlju 4.1.5.



Slika 3.2: Primjer jednostavne gramatike u STSG-formalizmu, preuzeto iz (Goodman et al., 2003)

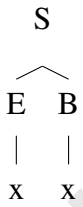
3.3.1. Najvjerojatnija derivacija

Slika 3.3: Najvjerojatnija derivacija, $p = \frac{3}{9}$

Kriterij najvjerojatnije derivacije (engl. *most probable derivation*) optimalnu sintaktičku strukturu neke rečenice definira kao sintaktičko stablo koje nastaje najvjerojatnijom derivacijom koja generira tu rečenicu. Vjerojatnost derivacije računa se prema definiciji iz STSG-formalizma prikazanoj u (3.5). Optimalna sintaktička struktura rečenice *xx* na temelju gramatike prikazane u 3.2 je 3.3. Ocjena koju kriterijska funkcija

ove definicije daje svakom stablu u biti je vjerojatnost najvjerojatnije derivacije kojom se to stablo može dobiti. Kao što je prikazano na slikama 3.3 i 3.4, isto sintaktičko stablo se može dobiti na temelju više različitih derivacija u STSG-formalizmu. Štoviše, taj broj mogućih različitih derivacija raste eksponencijalno s porastom duljine rečenice pa se odabir samo jedne od njih za ocjenjivanje cijelog stabla ne čini jako informativan. Općenito u praksi ova definicija sintaktičke strukture daje najlošije rezultate.

3.3.2. Najvjerojatnije sintaktičko stablo



Slika 3.4: Najvjerojatnije sintaktičko stablo, $p = \frac{4}{9}$

Kriterij najvjerojatnijeg sintaktičkog stabla (engl. *most probable parse*) optimalnu sintaktičku strukturu neke rečenice definira kao sintaktičko stablo najveće vjerojatnosti definirane u (3.6). Ta vjerojatnost u biti je kriterijska funkcija kojom se ocjenjuju sintaktička stabla. Slika 3.4 prikazuje optimalnu sintaktičku strukturu rečenice *xx* na temelju kriterija najvjerojatnijeg sintaktičkog stabla za gramatiku prikazanu u 3.2. Za razliku od kriterija najvjerojatnije derivacije koji u obzir uzima samo jednu derivaciju sintaktičkog stabla, kriterij najvjerojatnijeg sintaktičkog stabla uzima u obzir sve derivacije. Loša strana ovog pristupa jest u tome što mnoge od tih derivacija nisu lingvistički utemeljene, a ipak se pojavljuju zbog razlike u ekspresivnosti prirodnog jezika i STSG-formalizma. Drugi znatan problem ovog kriterija jest njegova složenost. U (Sima'an, 1996) pokazano je da je problem pronalaženja najvjerojatnijeg sintaktičkog stabla na temelju gramatike zadane u STSG-formalizmu NP-težak.

3.3.3. Najviše sintaktičkih jedinki

Većina mjera kojima se evaluira sastavna struktura rečenice u obzir uzima broj točno određenih sintaktičkih jedinki sintaktičkog stabla. Kriterijska funkcija koju koristi kriterij najviše sintaktičkih jedinki jest očekivan broj ispravnih sintaktičkih jedinki (engl. *maximum constituent parse*). Za vjerojatnost da se sintaktička jedinka s kategorijom X koja obuhvaća rečenični podniz $w_s \dots w_t$ nalazi u ispravnom sintaktičkom stablu vrijedi:

Slika 3.5: Stablo s najviše očekivanih ispravnih sintaktičkih jedinki, 2

$$P(X \overset{*}{\Rightarrow} w_s \dots w_t | S \overset{*}{\Rightarrow} w_1 \dots w_n) = \frac{P(S \overset{*}{\Rightarrow} w_1 \dots w_{s-1} X w_{t+1} \dots w_n) P(X \overset{*}{\Rightarrow} w_s \dots w_t)}{P(S \overset{*}{\Rightarrow} w_1 \dots w_n)} \tag{3.12}$$

Slika 3.5 prikazuje stablo s najviše očekivanih ispravnih sintaktičkih jedinki. U tom stablu je s vjerojatnošću 1 ispravna sintaktička jedinka S, sintaktička jedinka A je ispravna s vjerojatnošću $\frac{5}{9}$, a sintaktička jedinka B s vjerojatnošću $\frac{4}{9}$. Također valja uočiti da se sintaktičko stablo prikazano na slici 3.5 ne može dobiti na temelju niti jedne derivacije iz gramatike 3.2.

3.3.4. Jednostavnosni kriterij

Slika 3.6: Najkraća derivacija, $rang$ = 1

U (Bod, 2000b) predložen je nov kriterij za određivanje optimalne sintaktičke strukture nazvan jednostavnosni kriterij (engl. *simplicity criterion*) koji kao optimalnu strukturu uzima najkraću derivaciju. Sva pravila s istim korjenom sortirana su prema svojoj vjerojatnosti te je najvjerojatnijem dan rang 1, drugom najvjerojatnijem rang 2 itd. Rang pojedine derivacije računa se kao suma rangova svih korištenih pravila. Najbolje rangirana derivacija neke rečenice njena je optimalna sintaktička struktura.

4. Implementacija

4.1. Opis implementacije

Implementacija DOP-modela načinjena u ovom radu logički se može podijeliti u dvije cjeline: akvizicija znanja te parsanje. U nastavku je dana logička dekompozicija ostvarene implementacije.

1. Akvizicija znanja:
   (a) generiranje nizova završnih znakova,
   (b) generiranje skupa binarnih stabala,
   (c) generiranje skupa pravila,
   (d) minimizacija skupa pravila.
2. Parsanje:
   (a) generiranje nizova završnih znakova,
   (b) parsanje nizova završnih znakova,
   (c) odabir niza s najvjerojatnijom sintaktičkom strukturom.

U nastavku poglavlja detaljnije su objašnjeni navedeni koraci.

4.1.1. Generiranje nizova završnih znakova

Generiranje nizova završnih znakova prvi je korak i akvizicije znanja i parsanja, a sastoji se od toga da se svaka riječ iz rečenice zamijeni sa završnim znakom iz odgovarajućeg skupa završnih znakova na temelju odgovarajuće funkcije preslikavanja – terminalizatora. Ovaj korak pretpostavlja da je svaku riječ moguće zamijeniti s više od jednim znakom, pa za svaku rečenicu može biti generirano više nizova završnih znakova, po jedan za svaki element Kartezijevog produkta skupova mogućih završnih

znakova svake riječi. Ovo je jedini dio cijelog modela koji zavisi o konkretnom jeziku na kojem se model izvodi. Cijeli postupak moguće je izvesti na bilo kojem drugom jeziku uz adekvatnu implementaciju ovog koraka.

Završni znakovi kojima se zamjenjuju riječi trebali bi oslikavati sintaktičku ulogu riječi u rečenici. Ugrubo govoreći, svaku pojavu neke riječi u nekoj rečenici trebalo bi se moći zamijeniti s nekom drugom riječi koja se preslikava u isti završni znak, bez gubitka gramatičke ispravnosti te rečenice. Jedan terminalizator koji sigurno zadovoljava ovaj kriterij jest da završni znakovi budu same riječi. Osim što takav izbor znatno povećava složenost zbog velikog broja završnih znakova, on dosta umanjuje generalizacijsku moć modela. Glavni razlog tome jest to što je morfologija jezika pritom zanemarena, odnosno zanemarena je činjenica da određena pravila utječu na sam oblik riječi i da je onaj oblik u kojem se riječ pojavila u rečenici dijelom posljedica sintaktičke uloge te riječi. Drugi bitan razlog je u tome što velika većina korpusa ili banaka stabala na kojima se model može učiti predstavlja uglavnom smislene rečenice izvučene iz različitih instanci ljudske komunikacije. Ekspresivnost prirodnog jezika omogućuje izgradnju gramatički ispravnih rečenica koje su sa semantičkog aspekta besmislene.[1] Zbog svoje gramatičke ispravnosti te rečenice su sa sintaktičkog aspekta jednakovrijedne kao sve druge rečenice koje su mnogo češće element ljudske komunikacije.

U ovom radu za preslikavanje riječi u završne znakove koristi se uvriježeni princip upotrebe oznaka vrste riječi (POS) odnosno morfosintaktičkih opisnika (MSD). Ovisno o vrsti riječi, za svaki oblik riječi postoje dodatne informacije koje ga određuju. Tako npr. imenice možemo dodatno specificirati na temelju roda, broja, padeža, glagole na temelju lica itd. Količina informacija kojom se specificira pojedina vrsta riječi utječe na količinu završnih znakova te na samu moć generalizacije modela. U ovom radu iskušano je nekoliko različitih razina informacija o vrsti riječi. Jedan od najvećih problema preslikavanja riječi u oznake vrste riječi je u tome što različite riječi mogu imati neke svoje oblike jednake. Taj se problem može ublažiti korištenjem komponenti kao što su POS/MSD označivač, što ovdje nije učinjeno. U ovom radu riječ se preslikava u sve oznake koje može poprimiti, a očekuje se da će parser preferirati ispravni niz oznaka. Ovo očekivanje jednako je utemeljeno kao i opravdanje nenadziranog pristupa DOP-modela koje je izneseno u poglavlju 3.2.1. Stoga, ova implementacija DOP-modela ujedno predstavlja i POS/MSD označivač.

U ovom radu ukupno su implementirana tri terminalizatora čiji su opisi dani u

[1] Jedan od najpoznatijih primjera takve rečenice je: *"Colorless green ideas sleep furiously."* iz (Chomsky, 1957)

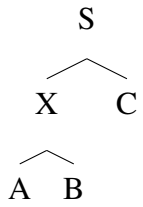
tablici 4.1.

Tablica 4.1: Implementirani terminalizatori

	Vrsta riječi									
Ter.	Imenica	Glagol	Pridjev	Zamjen.	Broj	Prilog	Prijedlog	Veznik	Čestica	Uzvik
MSD	oznaka, rod, broj, padež	oznaka, tip, vrijeme, lice, broj	oznaka, rod, broj, padež	oznaka, tip, lice, broj, padež	oznaka, tip, rod, broj, padež	oznaka	oznaka, padež	oznaka	oznaka	oznaka
POS-C	oznaka, padež	oznaka	oznaka, padež	oznaka	oznaka	oznaka	oznaka	oznaka	oznaka	oznaka
POS	oznaka	oznaka	oznaka	oznaka	oznaka	oznaka	oznaka	oznaka	oznaka	oznaka

4.1.2. Generiranje skupa binarnih stabala

Generiranje skupa binarnih stabala korak je koji služi za izgradnju banke stabala za nenadzirani pristup kakav se koristi u ovom radu. Za prikaz stabla možemo definirati zapis u kojem čvor s oznakom X koji ima ima dva djeteta, podstabla m i n, možemo prikazati ${}_X(m\ n)$. Za binarno stablo sa slike 4.1 odgovarajući zapis bio bi ${}_S({}_X(A\ B)\ C)$



Slika 4.1: Jednostavno binarno stablo

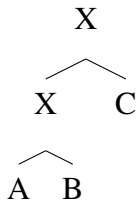
Sva binarna stabla rečenice $w_1 \ldots w_n$ možemo dobiti algoritmom 1.
Nenadzirani model DOP-a za označavanje unutarnjih čvorova sintaktičkog stabla predlaže da se svi unutarnji čvorovi označe nezavršnim znakom X, osim korijena stabla koji se označava nezavršnim znakom S. Takva notacija ima nekoliko nedostataka. Prvi

Algoritam 1 $MakeAllBinaryTrees(w_s, \ldots, w_t)$

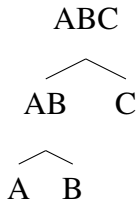
```
if s == t then
    return w_s
else
    T = {}
    for i = s to t − 1 do
        M = MakeAllBinaryTrees(w_s, ..., w_i)
        N = MakeAllBinaryTrees(w_{i+1}, ..., w_t)
        for all m ∈ M do
            for all n ∈ N do
                T ← _X(m n)
            end for
        end for
    end for
    return T
end if
```


nedostatak je u tome što se pravi distinkcija između cijele rečenice kao sintaktičke jedinke i dijela rečenice kao sintaktičke jedinke. Iako takva distinkcija naravno postoji, vrlo čest je slučaj da neki dio rečenice može stajati zasebno kao samostalna rečenica te da on u rečenici koje je dio čini sintaktičku jedinku. Neki od najčešćih slučaja tome su složene rečenice koje nastaju spajanjem jednostavnijih. Drugi nedostatak je u tome što se ne pravi distinkcija između različitih sintaktičkih kategorija kao što su imenična ili glagolska fraza. Prvi i drugi nedostatak međusobno su suprotni te sam DOP-model ne nudi način za njihovo rješavanje. U ovoj implementaciji DOP-a pokušalo se na nekoliko načina ublažiti navedene nedostatke. Osim predloženog označavanja sintaktičkog stabla kakav je prikazan na slici 4.1 i koji ćemo nazvati *S-X*, u ovom radu se predlaže označavanje svih unutarnjih čvorova istim znakom, slika 4.2, što ćemo nazvati *all-X*, a što bi trebalo ublažiti prvi nedostatak, no ujedno i naglasiti drugi. Drugi prijedlog označavanje je unutarnjih čvorova na način da nezavršni znak bude jednak konkatenaciji znakova lijevog i desnog djeteta, slika 4.3, što ćemo nazvati *concat*, a što bi trebalo ublažiti drugi nedostatak, no ujedno i naglasiti prvi.

Osim navedenih načina označavanja čvorova sintaktičkog stabla pokušalo se još s jednim načinom u kojem se svaki čvor označava oznakom jednog od djeteta. Taj način za svako stablo uvodi više od jednog načina označavanja jer za svaki unutarnji čvor imamo po dva izbora. Ovim načinom označavanja može se sastavnim stablom simu-



Slika 4.2: Označavanje svih unutarnjih čvorova istim znakom



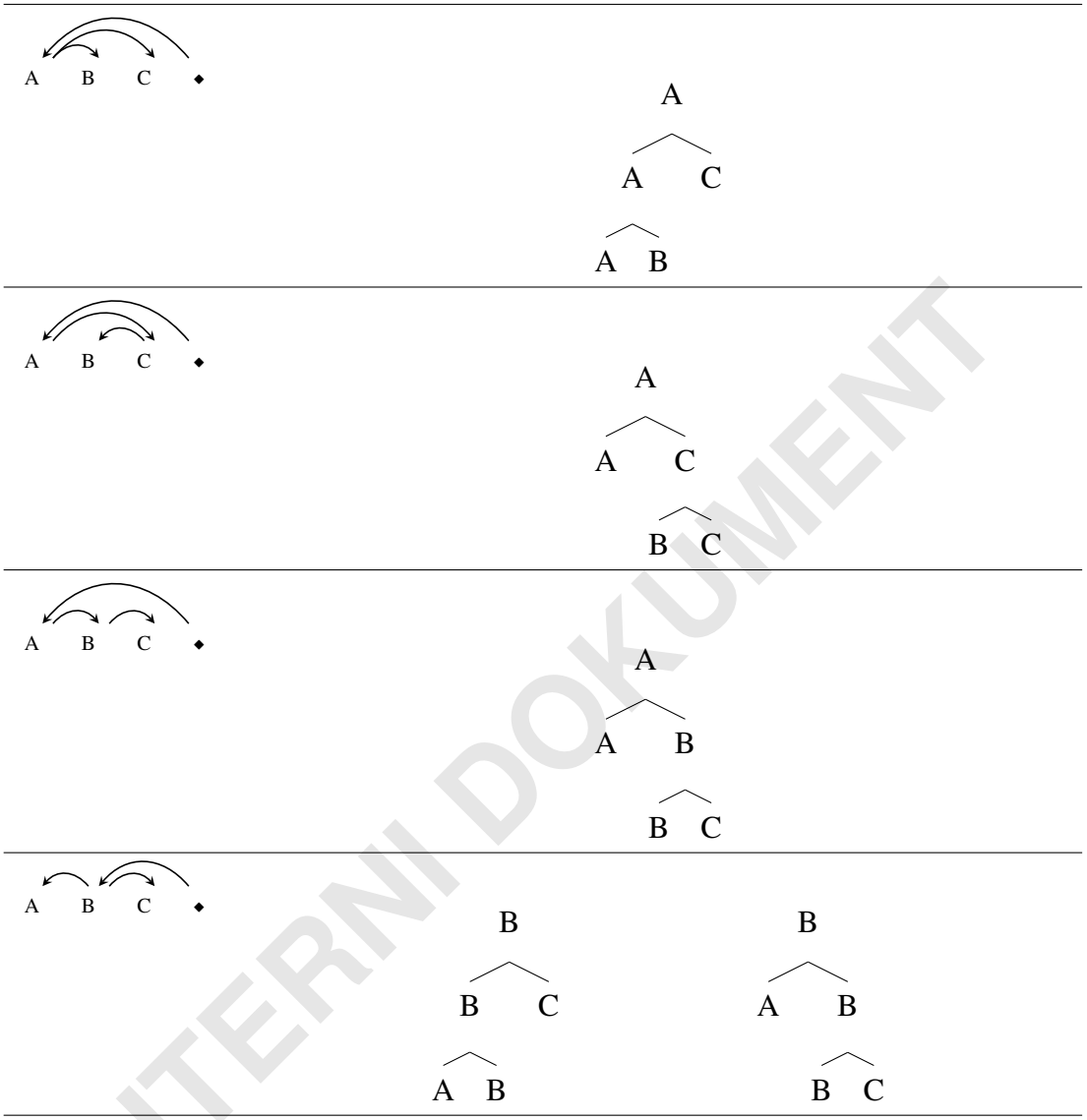
Slika 4.3: Označavanje svih unutarnjih čvorova konketenacijom znakova djece

lirati ovisnosno stablo. Ovisnosna stabla u kojima postoji neka glava koja ima ovisne članove na obje strane na ovaj način može se prikazati pomoću više sastavnih stabala koja uzimaju u obzir kojim redoslijedom glava uzima svoje ovisne članove. Preslikavanje nekoliko ovisnosnih struktura rečenice duljine tri riječi prikazano je na slici 4.4. Ovakav način imenovanja untarnjih čvorova nazvat ćemo dep. Iako je ovim preslikavanjem moguće modelirati ovisnosnu strukturu rečenice, intuicija na kojoj se temelji DOP-model nije direktno primjenjiva za prikazivanje strukture rečenice ovisnosnim stablom jer je usmjerena k identifikaciji rečeničnih elemenata, a ne ovisnosti među riječima, pa će se na ovakva stabla u ovom radu gledati iz perspektive sintaktičke strukture sastavnog oblika.

4.1.3. Generiranje skupa pravila

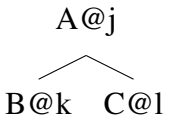
Generiranje skupa pravila korak je u kojem se iz banke binarnih stabala generiranih u prethodnom koraku ekstrahiraju sva pravila i pripadajuće vjerojatnosti. Svako pravilo predstavlja određeno podstablo, a njegova vjerojatnost računa se na temelju odgovarajućeg procjenitelja. Broj svih podstabala može biti vrlo velik jer u općenitom slučaju binarno stablo ima eksponencijalan broj podstabala u odnosu na broj završnih znakova odnosno riječi u rečenici. U (Goodman et al., 2003) predložena je redukcija STSG-gramatike u PCFG-gramatiku čiji je broj pravila u linearnoj ovisnosti s brojem riječi u rečenici odnosno cijelim korpusom što predstavlja znatno smanjenje cjelokupne složenosti. U nastavku će biti objašnjena navedena redukcija.

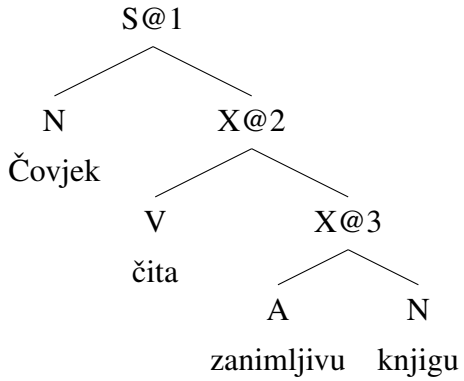
Svakom unutarnjem čvoru svakog stabla u banci stabala dodijeljen je jedinstven



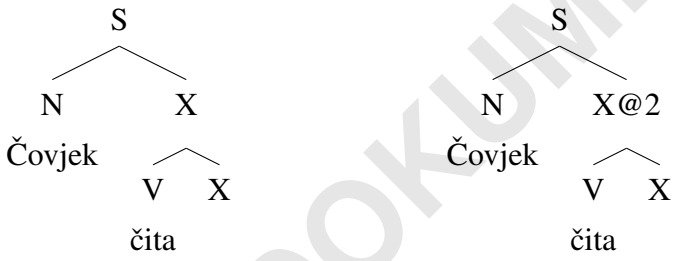
Slika 4.4: Preslikavanje ovisnosnih u sastavna stabla

broj, adresa. Primjer jednog takvog stabla dan je na slici 4.5. Oznaka $A@k$ označava čvor s adresom k koji je označen nezavršnim znakom A. Svaki unutarnji čvor iz banke stabala ima točno jedan nezavršni znak u rezultirajućoj PCFG-gramatici. Za čvor $A@k$ odgovarajući nezavršni znak će biti A_k. Takvi nezavršni znakovi koji predstavljaju adresirane čvorove zovu se "interni" nezavršni znakovi, dok se originalni nezavršni znakovi zovu "eksterni" nezavršni znakovi. Neka je a_j broj netrivijalnih podstabala kojima je korijen čvor $A@j$. Neka je a broj netrivijalnih podstabala kojima je korijen nezavršni znak A. Tada vrijedi $a = \sum_j a_j$. Promotrimo čvor $A@j$:





Slika 4.5: Primjer adresiranog stabla

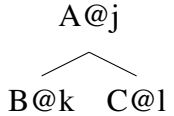


Slika 4.6: STSG-podstablo i njemu homomorfna PCFG-podderivacija

Postoji b_k netrivijalnih podstabala kojima je korijen čvor $B@k$ te c_l netrivijalnih podstabala kojima je korijen čvor $C@l$. Svako podstablo kojemu je korijen čvor $A@j$ može kao svoje lijevo podstablo imati bilo koje podstablo kojem je $B@k$ korijen ili u trivijalnom slučaju samo jedan čvor, B što je ukupno b_k+1 mogućnost. Slično za desno podstablo imamo $c_l + 1$ mogućnost. Dakle, za broj netrivijalnih podstabala kojima je čvor $A@j$ korijen vrijedi rekurzivna relacija $a_j = (b_k + 1)(c_l + 1)$.

Za PCFG-podderivaciju možemo reći da je homomorfna STSG-podstablu ako ima jednake čvorove kao podstablo pri čemu su korijen i svi nezavršni znakovi među listovima eksterni dok su svi unutarnji čvorovi interni. Slika 4.6 prikazuje STSG-podstablo i njemu homomorfnu PCFG-podderivaciju.

Neka $\overline{a}$ označava broj pojavljivanja nezavršnog znaka A u korpusu. Da bi napravili ciljnu PCFG-gramatiku potrebno je za svaki čvor oblika



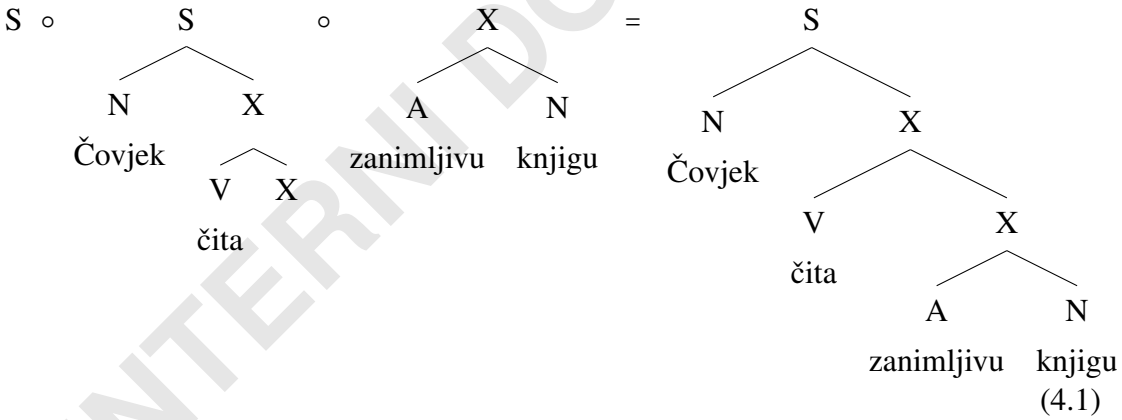
generirati osam PCFG-produkcija prikazanih u tablici 4.2.

Izvod za vjerojatnosti može se naći u (Goodman et al., 2003). Za PCFG-derivaciju kažemo da je homomorfna STSG-derivaciji ako za svaku supstituciju STSG-podstabla

Tablica 4.2: Produkcije PCFG gramatike s pripadajućim vjerojatnostima

Produkcija	Procjenitelj relativnom frekvencijom	Bonnemov procjenitelj
$A_j \longrightarrow B\ C$	$1/a_j$	$1/4$
$A_j \longrightarrow B_k\ C$	b_k/a_j	$1/4$
$A_j \longrightarrow B\ C_l$	c_l/a_j	$1/4$
$A_j \longrightarrow B_k\ C_l$	$b_k c_l/a_j$	$1/4$
$A \longrightarrow B\ C$	$1/a$	$1/4\overline{a}$
$A \longrightarrow B_k\ C$	b_k/a	$1/4\overline{a}$
$A \longrightarrow B\ C_l$	c_l/a	$1/4\overline{a}$
$A \longrightarrow B_k\ C_l$	$b_k c_l/a$	$1/4\overline{a}$

postoji odgovarajuća PCFG-podderivacija. Primjerice za STSG-derivaciju:

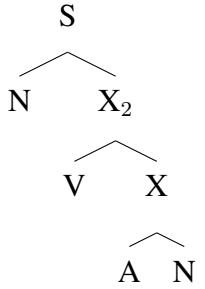


(4.1)

odgovarajuća homomorfna PCFG-derivacija je:

$$\underline{S} \Longrightarrow N\underline{X_2} \Longrightarrow NV\underline{X} \Longrightarrow NVAN \tag{4.2}$$

čije je odgovarajuće sastavno stablo:



(4.3)

Za sastavno stablo PCFG-a kažemo da je homomorfno sastavnom stablu STSG-a ako

su identični ako zamjenom internih nezavršnih znakova PCFG-stabla eksternim dobijemo odgovarajuće STSG-stablo. Rezultirajuće stablo iz STSG-derivacije (4.1) i PCFG-stablo prikazano u (4.3) su homomorfni. Svaka STSG-derivacija može imati više homomorfnih PCFG-derivacija. Vjerojatnost STSG-derivacije će onda biti jednaka sumi svih njoj homomorfnih PCFG-derivacija. Razlog postojanja više homomorfnih PCFG-derivacija jednoj STSG-derivaciji je u tome što se svakom čvoru u korpusu daje jedinstvena adresa iako je moguće da su neka podstabla, ako ne i cijela stabla, u korpusu jednaki. To je temelj za postupak minimizacije skupa pravila koji je predložen u ovom radu.

4.1.4. Minimizacija skupa pravila

Minimizacija skupa pravila korak je u kojem se smanjuje ukupan broj pravila bez utjecaja na iznos vjerojatnosti STSG-derivacija. Sam postupak temelji se na eliminaciji redundancija među pravilima.

Algoritam 2 $GrammarMinimization$

repeat
  for all $rule\ pair\ (r_1, r_2)\ where\ r_1 \equiv r_2 \equiv (X \longrightarrow \alpha)$ **do**
    $p(r_1)\ +=\ p(r_2)$
    $delete\ r_2$
  end for
  for all $rule\ pair\ (q_1, q_2)\ where\ q_1 \equiv (Y_i \longrightarrow \beta)\ and\ q_2 \equiv (Y_j \longrightarrow \beta)$ **do**
    $apply\ substitution\ \{Y_i/Y_j\}\ to\ entire\ grammar$
    $delete\ q_2$
  end for
until $no\ changes\ made$

Algoritam 2 je algoritam za minimizaciju gramatike gdje su r_i, q_i PCFG-produkcije, X eksterni nezavršni znak, Y_i interni nezavršni znak, te α, β nizovi završnih i nezavršnih znakova (internih i eksternih). Algoritam gramatiku minimizira na način da svaka STSG-derivacija ima točno jednu homomorfnu PCFG-derivaciju. Originalni način izgradnje PCFG-gramatike svaki unutarnji čvor stabla adresira jedinstvenom adresom, pa su i međusobno jednaka podstabla u adresiranom obliku različita. Sumacija njihovih vjerojatnosti stoga se ne događa u postupku ekstrakcije pravila već prilikom parsanja. Prikazani algoritam minimizacije identificira homomorfna pravila te ih eliminira uz podešavanje njihovih vjerojatnosti. Možemo razlikovati dvije vrste pravila: ona ko-

jima se s lijeve strane nalazi eksterni nezavršni znak, te ona kojima je s lijeve strane interni nezavršni znak. Primjena nekog pravila kojem je s lijeve strane eksterni nezavršni znak predstavlja početak dodavanja novog STSG-podstabla, dok primjena nekog pravila kojem je s lijeve strane interni nezavršni znak predstavlja nastavak izgradnje već započetog STSG-podstabla. Ukoliko algoritam detektira dva jednaka pravila koja s lijeve strane imaju jednak eksterni nezavršni znak, onda će im zbrojiti vjerojatnosti te ih zamijeniti jednim od njih. Ukoliko algoritam detektira dva pravila kojima su desne strane jednake te su im interni nezavršni znakovi jednaki do na adresu, onda će eliminirati jednog od njih te će zamijeniti adresu svih čvorova u gramatici s adresom jednakom onoj čvora koji se eliminira adresom čvora koji ostaje. Vjerojatnosti tih pravila na temelju izvoda moraju biti jednake te se vjerojatnost pravila koji ostaje ne mijenja.

Minimizacija skupa pravila nema utjecaja na sam model, no s implementacijske strane vrlo je bitna jer uvelike smanjuje prostornu i vremensku složenost. S povećanjem banke stabala na kojoj se model uči povećava se i faktor smanjenja gramatike jer se ujedno povećava vjerojatnost da su neki nizovi završnih znakova već ranije viđeni. Pošto se faktor smanjenja povećava s povećanjem rečenica u banci stabala možemo ugrubo reći da je veličina minimiziranog skupa pravila u logaritamskoj ovisnosti u odnosu na veličinu banke stabala što predstavlja znatno poboljšanje u odnosu na linearan odnos koji uvodi sama PCFG-redukcija. Strogo matematički gledano, ukoliko pretpostavimo da postoji ograničenje na duljinu rečenice, onda složenost zasigurno nije logaritamska jer postoji konačan broj različitih nizova završnih znakova pa nakon tog broja sigurno dodavanjem novih rečenica broj pravila ostaje konstantan, samo dolazi do mijenjanja njihovih vjerojatnosti.

4.1.5. Parsanje nizova završnih znakova

Parsanje nizova završnih znakova učinjeno je prema kriteriju najvjerojatnijeg sintaktičkog stabla uz pomoć algoritma Cocke–Younger–Kasami (CYK), predloženog u (Cocke i Schwartz, 1970; Younger, 1967; Kasami, 1965). Algoritam CYK je algoritam koji uz pomoć dinamičkog programiranja obavlja parsanje od dna prema vrhu (engl. *bottom-up parsing*) u kojem se sintaktičko stablo gradi od listova prema korijenu. Gramatika koju algoritam koristi treba biti u Chromskyjevom normalnom obliku[2] što PCFG-

[2]Za konteksno neovisnu gramatiku kažemo da je u Chromskyjevom normalnom obliku ako su sve njene produkcije u jednom od oblika: $A \longrightarrow BC$, $A \longrightarrow a$ ili $S \longrightarrow \epsilon$, gdje su A, B i C nezavršni znakovi, S je početni nezavršni znak, a je završni znak, a ϵ je prazni niz.

redukcija STSG-formalizma zadovoljava[3].

Algoritam 3 *Modified CYK algorithm*

R = *set of all symbols in grammar* {terminals, ext. and int. non-terminals}
$r = |R|$
$S = w_1 \ldots w_n$, *input terminal sequence*, $w_i \in R$
$P[n, n, r]$ = *array of lists of tree nodes*
for $i = 1 \rightarrow n$ **do**
  Node $x = w_i$
  $p(x) = 1$
  add x *to* $P[i, 1, w_i]$
end for
for $i = 2 \rightarrow n$ **do**
  for $j = 1 \rightarrow (n - i + 1)$ **do**
    for $k = 1 \rightarrow (i - 1)$ **do**
      for all $rule = (R_A \longrightarrow R_B\ R_C)$ *in grammar* **do**
        Nodes $L = P[j, k, R_B]$
        Nodes $R = P[j + k, i - k, R_C]$
        for all $NodePair\ (l, r) \in L \times R$ **do**
          Node $x = {}_{R_A}(l\ r)$
          $p(x) = p(l) * p(r) * p(rule)$
          add x *to* $P[j, i, R_A]$
        end for
      end for
    end for
  end for
end for
return *all* $P[1, n, X]$ *where* X *is external non – terminal*

Algoritam 3 predstavlja modificiran oblik originalnog algoritma prilagođen PCFG-redukciji STSG-formalizma. Algoritam na temelju zadane PCFG-gramatike i ulaznog niza završnih znakova obavlja parsiranje te vraća sva sintaktička stabla kojima je ko-

[3]Strogo gledano gramatika dobivena navedenom PCFG-redukcijom nije u Chromskyjevom normalnom obliku već u modifikaciji tog oblika gdje produkcije oblika $A \longrightarrow a$ ne postoje već su primjenjene na desne strane produkcija oblika $A \longrightarrow BC$, a produkcije oblika $S \longrightarrow \epsilon$ nema. Može se uočiti da se ovakvom gramatikom ne može parsati rečenica od jedne riječi, no to je i za očekivati jer za takvu rečenicu nije moglo biti načinjeno binarno stablo

rijen eksterni nezavršni znak. Algoritam prvo izgrađuje sva sintaktička stabla za podnizove duljine 2, potom podnizove duljine 3 itd., sve do duljine cijelog ulaznog niza. Prilikom izgradnje sintaktičkog stabla niza određene duljine koriste se već izgrađena sintaktička stabla nizova manjih duljina. Izgrađena sintaktička stabla zapisuju se u strukturu podataka $P[i_1, i_2, i_3]$ dimenzija $n \times n \times r$. Ta struktura je trodimenzionalno polje lista sintaktičkih stabala. Prva dva indeksa određuju podniz ulaznog niza, pri čemu prvi indeks označava redni broj prvog elementa, a drugi duljinu promatranog podniza. Treći indeks predstavlja nezavršni znak iz kojega se taj podniz izvodi. Element tog polja je lista koja sadrži sva sintaktička stabla kojima je promatrani nezavršni znak korijen, a završni znakovi promatrani podniz. Osnovna varijanta algoritma u kojoj se ne pamte pojedinačna sintaktička stabla već samo istinosna vrijednost može li se iz nekog nezavršnog znaka izvesti određeni podniz ima složenost $O(n^3 \cdot |G|)$, gdje G označava broj produkcija PCFG-gramatike. To jednostavno slijedi na temelju prve četiri ugniježđene petlje. Ova modificirana verzija ima i petu ugniježđenu petlju koja se izvodi onoliko puta koliki je umnožak broja podstabala iz kojih se gradi novi čvor. Taj broj ovisi o samoj gramatici te je u najgorem slučaju u eksponencijalnoj ovisnosti s duljinom promatranih podnizova što je ujedno i razlog činjenici da se problem pronalaženja optimalne sintaktičke strukture prema kriteriju najvjerojatnijeg sintaktičkog stabla svrstava u klasu NP-teških problema. Jedan od načina zaobilaženja tog problema je da se u strukturi $P[i_1, i_2, i_3]$ ne pamte sva podstabla, već k najvjerojatnijih za svaku kombinaciju vrijednosti i_1, i_2 i i_3. U tom slučaju algoritam ima složenost $O(n^3 \cdot |G| \cdot k^2)$, no u tom obliku više ne daje egzaktno rješenje.

4.1.6. Odabir niza s najvjerojatnijom sintaktičkom strukturom

Algoritam CYK predstavljen u prethodnom poglavlju vraća skup PCFG-derivacija parsanih na temelju ulaznog niza završnih znakova. Tim PCFG-derivacijama potom se uklanjaju adrese s nezavršnih znakova čime se one pretvaraju u homomorfne STSG-derivacije. STSG-derivacije grupiraju se u sintaktička stabla pri čemu im se vjerojatnosti zbrajaju. Ukoliko se pri izgradnji binarnih stabala koristio pristup da se korijen stabla označi nezavršnim znakom S, onda se iz dobivenog skupa sintaktičkih stabala uklanjaju ona koja ne započinju tim znakom, dok se u ostalim slučajevima to ne čini. Ono sintaktičko stablo koje nakon tog postupka ima najveću vjerojatnost predstavlja optimalnu sintaktičku strukturu ulaznog niza završnih znakova.

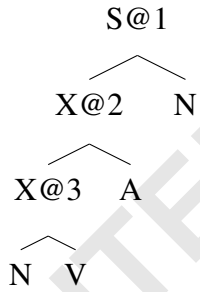
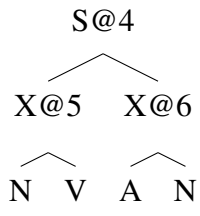
U poglavlju 4.1.1 rečeno je da se za rečenicu koja se parsa generiraju svi mogući nizovi završnih znakova koji pokrivaju sve mogućnosti klasa riječi koje neka riječ

rečenice može biti. Kao ispravan niz završnih znakova te kao optimalna sintaktička struktura cijele rečenice uzima se onaj niz završnih znakova čija je sintaktička struktura najvjerojatnija.

4.2. Primjer postupka

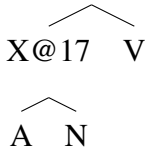
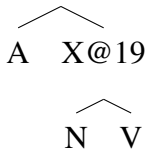
Pretpostavimo da model želimo naučiti na korpusu koji se sastoji od dvije rečenice: "Čovjek čita zanimljivu knjigu" i "Malena djevojčica plače". Za terminalizator ćemo uzeti POS terminalizator. Na temelju tog odabira dobivamo dva niza završnih znakova i to: "N V A N" i "A N V". Unutrašnji čvorovi su označeni na način predložen u originalnom modelu, S-X. Naučena gramatika prikazana je u tablicama 4.3 i 4.4.

Tablica 4.3: Dio gramatike naučen na temelju niza "N V A N"

Stablo	CFG pravila	RF vjerojatnosti	Bonn vjerojatnosti
	$X_3 \longrightarrow N\ V$	1	0.25
	$X \longrightarrow N\ V$	0.0625	0.02083
	$X_2 \longrightarrow X\ A$	0.5	0.25
	$X_2 \longrightarrow X_3\ A$	0.5	0.25
	$X \longrightarrow X\ A$	0.0625	0.02083
	$X \longrightarrow X_3\ A$	0.0625	0.02083
	$S_1 \longrightarrow X\ N$	0.33	0.25
	$S_1 \longrightarrow X_2\ N$	0.66	0.25
	$S \longrightarrow X\ N$	0.05	0.03571
	$S \longrightarrow X_2\ N$	0.1	0.03571
	$X_5 \longrightarrow N\ V$	1	0.25
	$X \longrightarrow N\ V$	0.0625	0.02083
	$X_6 \longrightarrow A\ N$	1	0.25
	$X \longrightarrow A\ N$	0.0625	0.02083
	$S_4 \longrightarrow X\ X$	0.25	0.25
	$S_4 \longrightarrow X_5\ X$	0.25	0.25
	$S_4 \longrightarrow X\ X_6$	0.25	0.25
	$S_4 \longrightarrow X_5\ X_6$	0.25	0.25
	$S \longrightarrow X\ X$	0.05	0.03571
	$S \longrightarrow X_5\ X$	0.05	0.03571
	$S \longrightarrow X\ X_6$	0.05	0.03571

	$S \longrightarrow X_5\ X_6$	0.05	0.03571
S@7 X@8 N N X@9 V A	$X_9 \longrightarrow V\ A$	1	0.25
	$X \longrightarrow V\ A$	0.0625	0.02083
	$X_8 \longrightarrow N\ X$	0.5	0.25
	$X_8 \longrightarrow N\ X_9$	0.5	0.25
	$X \longrightarrow N\ X$	0.0625	0.02083
	$X \longrightarrow N\ X_9$	0.0625	0.02083
	$S_7 \longrightarrow X\ N$	0.33	0.25
	$S_7 \longrightarrow X_8\ N$	0.66	0.25
	$S \longrightarrow X\ N$	0.05	0.03571
	$S \longrightarrow X_8\ N$	0.1	0.03571
S@10 N X@11 X@12 N V A	$X_{12} \longrightarrow V\ A$	1	0.25
	$X \longrightarrow V\ A$	0.0625	0.02083
	$X_{11} \longrightarrow X\ N$	0.5	0.25
	$X_{11} \longrightarrow X_{12}\ N$	0.5	0.25
	$X \longrightarrow X\ N$	0.0625	0.02083
	$X \longrightarrow X_{12}\ N$	0.0625	0.02083
	$S_{10} \longrightarrow N\ X$	0.33	0.25
	$S_{10} \longrightarrow N\ X_{11}$	0.66	0.25
	$S \longrightarrow N\ X$	0.05	0.03571
	$S \longrightarrow N\ X_{11}$	0.1	0.03571
S@13 N X@14 V X@15 A N	$X_{15} \longrightarrow A\ N$	1	0.25
	$X \longrightarrow A\ N$	0.0625	0.02083
	$X_{14} \longrightarrow V\ X$	0.5	0.25
	$X_{14} \longrightarrow V\ X_{15}$	0.5	0.25
	$X \longrightarrow V\ X$	0.0625	0.02083
	$X \longrightarrow V\ X_{15}$	0.0625	0.02083
	$S_{13} \longrightarrow N\ X$	0.33	0.25
	$S_{13} \longrightarrow N\ X_{14}$	0.66	0.25
	$S \longrightarrow N\ X$	0.05	0.03571
	$S \longrightarrow N\ X_{14}$	0.1	0.03571

Tablica 4.4: Dio gramatike naučen na temelju niza "A N V"

Stablo	CFG pravila	RF vjerojatnosti	Bonn vjerojatnosti
	$X_{17} \longrightarrow A\ N$	1	0.25
	$X \longrightarrow A\ N$	0.0625	0.02083
	$S_{16} \longrightarrow X\ V$	0.5	0.25
	$S_{16} \longrightarrow X_{17}\ V$	0.5	0.25
	$S \longrightarrow X\ V$	0.05	0.03571
	$S \longrightarrow X_{17}\ V$	0.05	0.03571
	$X_{19} \longrightarrow N\ V$	1	0.25
	$X \longrightarrow N\ V$	0.0625	0.02083
	$S_{18} \longrightarrow A\ X$	0.5	0.25
	$S_{18} \longrightarrow A\ X_{19}$	0.5	0.25
	$S \longrightarrow A\ X$	0.05	0.03571
	$S \longrightarrow A\ X_{19}$	0.05	0.03571

Na temelju zadanog korpusa generirana je PCFG-gramatika koja se sastoji od 64 produkcije. Na toj gramatici pokrenut je algoritam za minimizaciju koji ju je smanjio na veličinu od 50 producija prikazanih u tablici 4.5.

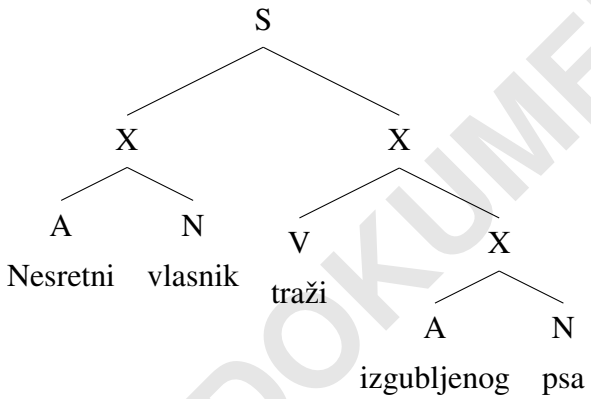
Tablica 4.5: Minimizirana gramatika

CFG pravilo	RF vjerojatnost	Bonn vjerojatnost
$X \longrightarrow N\ V$	0.1875	0.0625
$X_3 \longrightarrow N\ V$	1	0.25
$X \longrightarrow X\ A$	0.0625	0.0208
$X_2 \longrightarrow X\ A$	0.5	0.25
$X \longrightarrow X_3\ A$	0.0625	0.0208
$X_2 \longrightarrow X_3\ A$	0.5	0.25
$S \longrightarrow X\ N$	0.1	0.0714
$S_1 \longrightarrow X\ N$	0.33	0.25
$S \longrightarrow X_2\ N$	0.1	0.0357
$S_1 \longrightarrow X_2\ N$	0.66	0.25
$X \longrightarrow A\ N$	0.1875	0.625
$X_6 \longrightarrow A\ N$	1	0.25

$S \longrightarrow X\ X$	0.05	0.0357
$S_4 \longrightarrow X\ X$	0.25	0.25
$S \longrightarrow X_3\ X$	0.05	0.0357
$S_4 \longrightarrow X_3\ X$	0.25	0.25
$S \longrightarrow X\ X_6$	0.05	0.0357
$S_4 \longrightarrow X\ X_6$	0.25	0.25
$S \longrightarrow X_3\ X_6$	0.05	0.0357
$S_4 \longrightarrow X_3\ X_6$	0.25	0.25
$X \longrightarrow V\ A$	0.125	0.0416
$X_9 \longrightarrow V\ A$	1	0.25
$X \longrightarrow N\ X$	0.0625	0.0208
$X_8 \longrightarrow N\ X$	0.5	0.25
$X \longrightarrow N\ X_9$	0.0625	0.0208
$X_8 \longrightarrow N\ X_9$	0.5	0.25
$S \longrightarrow X_8\ N$	0.1	0.0357
$S_1 \longrightarrow X_8\ N$	0.66	0.25
$X \longrightarrow X\ N$	0.0625	0.0208
$X_{11} \longrightarrow X\ N$	0.5	0.25
$X \longrightarrow X_9\ N$	0.0625	0.0208
$X_{11} \longrightarrow X_9\ N$	0.5	0.25
$S \longrightarrow N\ X$	0.1	0.0714
$S_{10} \longrightarrow N\ X$	0.33	0.25
$S \longrightarrow N\ X_{11}$	0.1	0.0357
$S_{10} \longrightarrow N\ X_{11}$	0.66	0.25
$X \longrightarrow V\ X$	0.0625	0.0208
$X_{14} \longrightarrow V\ X$	0.5	0.25
$X \longrightarrow V\ X_6$	0.0625	0.0208
$X_{14} \longrightarrow V\ X_6$	0.5	0.25
$S \longrightarrow N\ X_{14}$	0.1	0.0357
$S_{10} \longrightarrow N\ X_{14}$	0.66	0.25
$S \longrightarrow X\ V$	0.05	0.0357
$S_{16} \longrightarrow X\ V$	0.5	0.25
$S \longrightarrow X_6\ V$	0.05	0.0357
$S_{16} \longrightarrow X_6\ V$	0.5	0.25
$S \longrightarrow A\ X$	0.05	0.0357
$S_{18} \longrightarrow A\ X$	0.5	0.25

$S \longrightarrow A\ X_3$	0.05	0.0357
$S_{18} \longrightarrow A\ X_3$	0.5	0.25

Pretpostavimo da s naučenom gramatikom želimo parsati rečenicu “Nesretni vlasnik traži izgubljenog psa”. Primjenom terminalizatora dobivamo niz “A N V A N”. Oba procjenitelja za zadanu rečenicu daju isto sintaktičko stablo prikazano na slici 4.7. Ovo stablo ujedno je i ispravno sintaktičko stablo promatrane rečenice.



Slika 4.7: Sintaktičko stablo za rečenicu “Nesretni vlasnik traži izgubljenog psa”

$$p_{RF} = 6.96 \cdot 10^{-5}$$
$$p_{Bonn} = 1.45 \cdot 10^{-5}$$

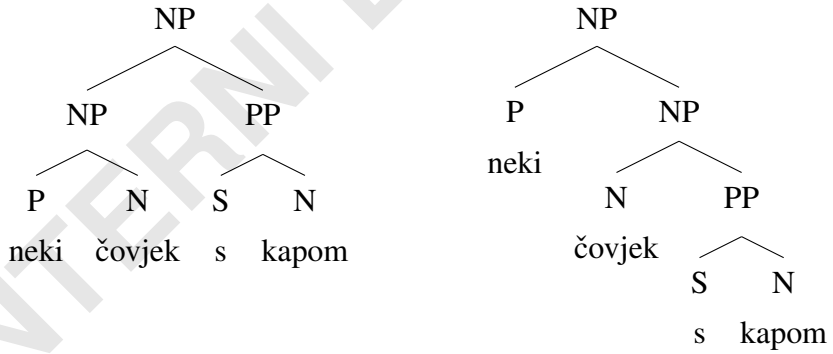
U tablici 4.6 prikazan je rad algoritma CYK na promatranoj rečenici. Svako polje predstavlja skup podstabala koja generiraju podniz koje to polje prekriva. Svako polje prekriva podniz čiji je početak u polju koje je vetikalo dolje u odnosu na promatrano polje, a kraj u polju koje je dolje desno u odnosu na promatrano polje. U ovom prikazu algoritmu je podešena vrijednost parametra $k = 1$, pa se u svakom polju bilježi najviše jedno podstablo za neki nezavršni znak i to ono najvjerojatnije. U najvišem polju koje prekriva cijeli niz samo ona stabla kojima je korijen eksterni nezavršni znak mogu predstavljati sintaktička stabla cijele rečenice. U ovom slučaju samo je jedno takvo, pa je to izlaz algoritma. To je ujedno i ispravno sintaktičko stablo prikazano na slici 4.7.

Tablica 4.6: Prikaz CYK algoritma za niz "A N V A N", k = 1

$\mathbf{S}(\mathbf{X}(\mathbf{A}\ \mathbf{N})\ \mathbf{X}(\mathbf{V}\ \mathbf{X}(\mathbf{A}\ \mathbf{N})))$ $S_{18}(A\ X(N\ X(V\ X_6(A\ N))))$ $S_4(X(A\ N)\ X(V\ X_6(A\ N)))$				
$S(A\ X(N\ X_9(V\ A)))$ $S_{18}(A\ X(N\ X_9(V\ A)))$ $S_4(X(A\ N)\ X(V\ A))$	$X(N\ X(V\ X_6(A\ N)))$ $X_8(N\ X(V\ X_6(A\ N)))$ $S(N\ X_{11}(X_9(V\ A)\ N))$ $S_{10}(N\ X_{11}(X_9(V\ A)\ N))$ $S_4(X_3(N\ V)\ X_6(A\ N))$ $S_1(X_2(X_3(N\ V)\ A)\ N)$			
$S(A\ X_3(N\ V)$ $S_{18}(A\ X_3(N\ V))$ $S_{16}(X_6(A\ N)\ V)$	$X(N\ X_9(V\ A))$ $X_8(N\ X_9(V\ A))$ $X_2(X_3(N\ V)\ A)$ $S(N\ X(V\ A))$ $S_{10}(N\ X(V\ A))$	$\mathbf{X}(\mathbf{V}\ \mathbf{X}(\mathbf{A}\ \mathbf{N}))$ $X_{14}(V\ X_{16}(A\ N))$ $X_{11}(X_9(V\ A)\ N)$ $S(X(V\ A)\ N)$ $S_1(X(V\ A)\ N)$		
$\mathbf{X}(\mathbf{A}\ \mathbf{N})$ $X_6(A\ N)$	$X(N\ V)$ $X_3(N\ V)$	$X(V\ A)$ $X_9(V\ A)$	$\mathbf{X}(\mathbf{A}\ \mathbf{N})$ $X_6(A\ N)$	
A	**N**	**V**	**A**	**N**

5. Evaluacija

Evaluacija sustava za sintaktičku analizu prirodnog jezika jest složen zadatak koji se u općenitom slučaju ne može izvesti na jedinstven način već ovisi o samoj namijeni za koju se takav sustav izgradio. Ipak, jedan od najčešćih pristupa evaluaciji je lingvistički pristup u kojem se evaluira lingvistička ispravnost sintaktičkih struktura koje je napravio sustav. Jedan od najvećih problema takvog pristupa je u tome što među lingvistima često ne postoji konsenzus u mnogim pitanjima. Primjerice za imensku frazu "neki čovjek s kapom" mogla bi se izgraditi dva različita sintaktička stabla, slika 5.1, koja se sa semantičkog aspekta ne razlikuju u informaciji koju prenose.

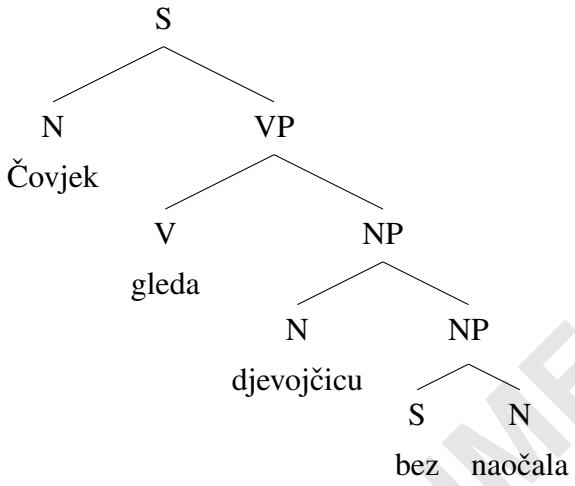


Slika 5.1: Sintaktičke strukture za imensku frazu "neki čovjek s kapom"

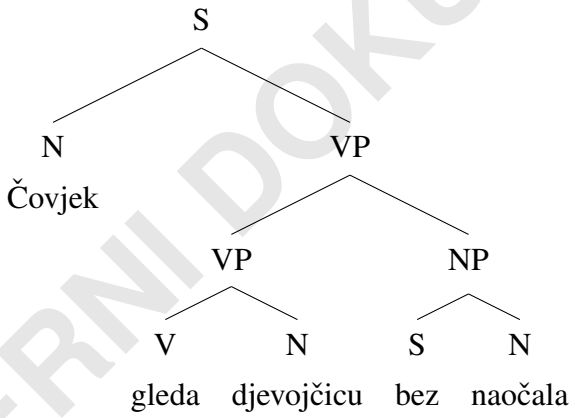
Često se sa svrhom eliminiranja višeznačnosti uvode određeni dogovori koji u velikom broju slučajeva sami sebi proturječe jer ih je teško formalizirati zbog činjenice da se sam prirodni jezik za sad još ne može formalizirati. Takvi dogovori ujedno predstavljaju problem nenadziranom učenju jer oni kao takvi nisu karakteristika jezika već su umjetno nadodani te ih se ne može inducirati samo na temelju neoznačenih primjera.

Osim višeznačnosti u sintaktičkoj strukturi koja sa semantičkog aspekta nema utjecaja, veći problem predstavlja sintaktička višeznačnost koja uzrokuje višestruka značenja iste rečenice. Primjerice za rečenicu "Čovjek gleda djevojčicu bez naočala." (slika 5.2), moguće je izgraditi dva različita sintaktička stabla od kojih prvo predstavlja slučaj u kojem čovjek gleda djevojčicu koja nema naočale, a u drugom čovjek ne koristeći

naočale promatra djevojčicu.



(a) djevojčica je bez naočala



(b) čovjek ne koristeći naočale gleda

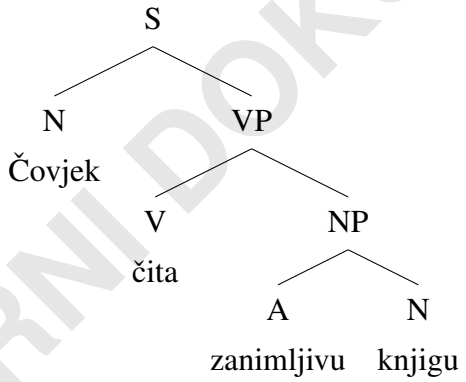
Slika 5.2: Sintaktičke strukture za rečenicu "Čovjek gleda djevojčicu bez naočala."

Ova višeznačnost posljedica je toga što zapis prirodnog jezika ne obuhvaća neke njegove karakteristike kao što je naglasak kojim prilikom govora identificiramo ispravno značenje rečenice. U sustavima koji sintaktičku analizu prirodnog jezika koriste za prevođenje rečenica iz jednog jezika u drugi ovo uglavnom ne predstavlja veći problem jer se višestruke varijante najčešće jednako prevode. U nekim slučajevima takva višeznačnost u nekom drugom jeziku ipak ima različite prijevode. Primjerice engleska rečenica: "I eat salad with tuna" ima čak tri različita značenja. Većina ljudi tu rečenicu shvaća u smislu da u salati ima tune, no osim tog značenja ova rečenica može značiti da se u društvu tune jede salata ili pak da se tuna kao pribor za jelo koristi da bi se salata jela. Prva dva značenja se na hrvatski prevode kao "Jedem salatu s tunom", pri čemu imamo sličan slučaj višeznačnosti kao u prethodnom primjeru, a treće značenje se prevodi kao "Jedem salatu tunom".

Svi navedeni problemi uvelike otežavaju evaluaciju te utječu na njen rezultat. U ovom pristupu koristit će se metoda evaluacije u kojoj svaka rečenica iz skupa za evaluaciju ima označenu točno jednu sintaktičku strukturu od strane eksperta i to onu koja predstavlja najvjerojatnije značenje promatrane rečenice. Za ocjenjivanje kvalitete stabla ono se uspoređuje s onim kojeg je načinio čovjek. Postoji više mjera koje služe za uspoređivanje dvaju stabala, a ona koja je odabrana u ovom pristupu je objašnjena u sljedećem odjeljku.

5.1. Metoda evaluacije

Za rečenicu: "$_0$Čovjek$_1$čita$_2$zanimljivu$_3$knjigu$_4$" sintaktička struktura dodijeljena od strane eksperta prikazana je na slici 5.3.



Slika 5.3: Sintaktička struktura za rečenicu "Čovjek čita zanimljivu knjigu"

Sintaktičko stablo t može se shvatiti kao skup sintaktičkih jedinki oblika $(x : i, j)$ gdje x označava sintaktičku kategoriju, a i i j su granice sintaktičke jedinke. Za stablo prikazano na slici 5.3 pripadajuće sintaktičke jedinke su:

Sintaktička jedinka	Rečenični dio
(NP: 2,4)	zanimljivu knjigu
(VP: 1,4)	čita zanimljivu knjigu
(S: 0,4)	Čovjek čita zanimljivu knjigu

Kod modela temeljenih na nenadziranom učenju kao što je i model predstavljen u ovom radu cilj sintaktičke analize najčešće nije identifikacija sintaktičkih kategorija, pa nas pri evaluaciji zanima samo jesu li granice sintaktičkih jedinki ispravne. Skup neoznačenih sintaktičkih jedinki – $(SNSJ)$ stabla t definira se kao:

$$SNSJ(t) = \{< i, j > |\ \exists x, (x : i, j) \in t\} \tag{5.1}$$

Za stablo prikazano na slici 5.3 pripadajući skup neoznačenih sintaktičkih jedinki je: $\{< 2, 4 >, < 1, 4 >, < 0, 4 >\}$

Ako je P skup sintaktičkih stabala p_i koje je odredio sustav, a G skup sintaktičkih stabala g_i koje je odredio ekspert, neoznačena preciznost (engl. *Unlabeled Precision – UP*) te neoznačeni odziv (engl. *Unlabeled Recall – UR*) definiraju se kao:

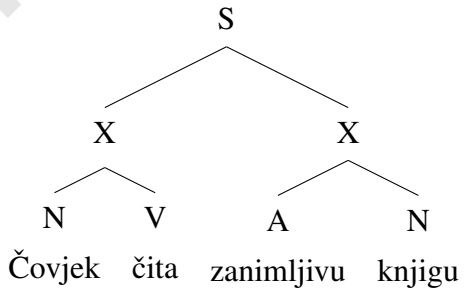
$$UP(P, G) \equiv \frac{\sum_i |SNSJ(p_i) \cap SNSJ(g_i)|}{\sum_i |SNSJ(p_i)|} \tag{5.2}$$

$$UR(P, G) \equiv \frac{\sum_i |SNSJ(p_i) \cap SNSJ(g_i)|}{\sum_i |SNSJ(g_i)|} \tag{5.3}$$

Neoznačena F_1 mjera jest njihova harmonijska sredina:

$$UF_1(P, G) = \frac{2}{UP(P, G)^{-1} + UR(P, G)^{-1}} \tag{5.4}$$

Primjerice, sintaktička struktura rečenice "Čovjek čita zanimljivu knjigu" prikazana na slici 5.4 ima pripadajući skup neoznačenih sintaktičkih jedinki $\{< 2, 4 >, < 0, 2 >, < 0, 4 >\}$. Kada se ona usporedi sa strukturom na slici 5.3 koju je dodijelio ekspert dobije se $UP = 2/3$, $UR = 2/3$ te $UF_1 = 2/3$.



Slika 5.4: Sintaktička struktura dodjeljena od strane modela

Pošto se u DOP-modelu koriste samo binarna stabla, broj sintaktičkih jedinki ovisi samo o broju riječi rečenice odnosno za 1 je manji od tog broja. Zbog toga će nazivnici u izrazima (5.2) i (5.3) uvijek biti jednaki pa će vrijednosti preciznosti i odziva, a s njima i UF_1 mjere, uvijek biti jednake.

5.2. Rezultati

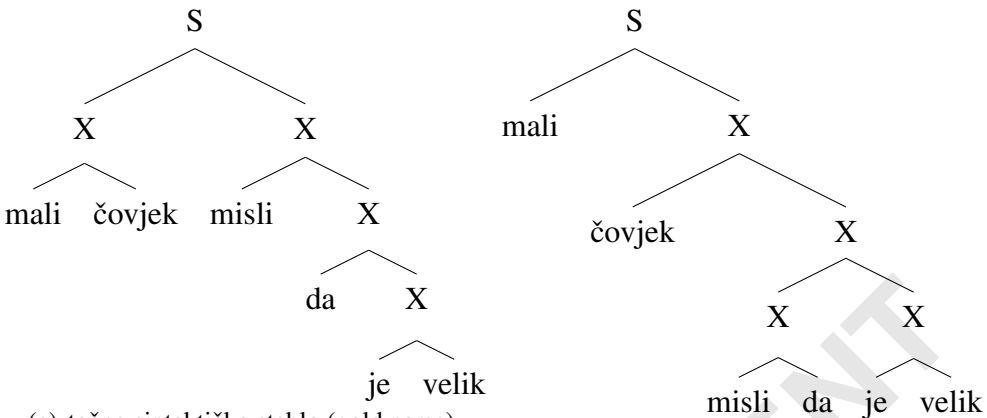
Model je naučen te testiran na Vjesnikovoj on-line arhivi koja obuhvaća vremensko razdoblje od 1999. do 2009. godine. Na raspolaganju je ukupno 4466178 rečenica. Za potrebe evaluacije ekspert je označio 100 rečenica pripadajućim sintaktičkim strukturama. Rezultati su prikazani u tablici 5.1.

Tablica 5.1: Rezultati

								vrijeme parsanja (s)			
Ter.	Proc.	Lab.	|K|	N_1	N_2	N_1/N_2	k	4	6	8	UF_1
POS	RF	concat	250	789408	72298	10.92	100	1.62	5.56	25.75	47.60
POS	Bonn	concat	250	789408	72298	10.92	100	1.77	6.44	25.50	46.07
POS	RF	concat	250	789408	72298	10.92	∞	1.61	5.64	23.75	48.90
POS	Bonn	concat	250	789408	72298	10.92	∞	1.46	5.36	24.23	46.07
POS	Bonn	concat	1000	10427110	654316	15.94	∞	61.85	170.72	644.5	38.86
POS	RF	S-X	2500	27374184	72218	379.05	∞	5.77	17.44	76.12	37.15
POS-C	Bonn	concat	250	9624788	598012	16.09	∞	214.31	1224.00	6070.50	42.14

Ter.	terminalizator
Proc.	procjenitelj
Lab.	način imenovanja nezavršnih znakova
|K|	veličina korpusa za učenje
N_1	broj generiranih pravila prije minimizacije
N_2	broj pravila nakon minimizacije
N_1/N_2	faktor smanjenja skupa pravila
k	parametar CYK algoritma
$4-8$	prosječno vrijeme parsanja rečenice prikazane duljine
UF_1	neoznačena F_1 mjera

Dobiveni rezultati lošiji su od rezultata koje je model postigao na engleskom jeziku (F_1 = 78.5) te njemačkom jeziku (F_1 = 65.4), no usporedivi su s rezultatima za kineski jezik (F_1 = 46.6). Pri izgradnji DOP-modela za te jezike je korišten POS/MSD označivač, pa zbog toga ti rezultati nisu u potpunosti usporedivi s onima dobivenima ovdje, no možemo pretpostaviti da korištenjem tog označivača ne bi dobili lošije rezultate. Neovisno o tome, uočeno je da ovaj model na hrvatskim rečenicama redovito griješi u određenim slučajevima.



(a) točno sintaktičko stablo (gold parse)

(b) dobiveno sintaktičko stablo

Slika 5.5: Usporedba točnog i dobivenog sintaktičkog stabla, $UF_1 = 50\%$

Na slici 5.5 prikazana je usporedba točnog i dobivenog od strane modela sintaktičkog stabla rečenice "mali čovjek misli da je velik". Model je napravio dvije greške u ovom slučaju. Prva greška je u tome da pridjev "mali" koji u ovoj rečenici ima ulogu atributa nije povezao u sintaktičku jediku s imenicom "čovjek" na koju se odnosi. Iako model u većini slučajeva poveže pridjev s imenicom na koju se odnosi, u nekima se to ipak ne dogodi. Razlog zbog kojeg se to dogodilo u ovom slučaju jest da dio rečenice "čovjek misli da je velik" može biti sam za sebe sintaktička jedinka, čak može stajati i kao samostalna rečenica. Stoga, dolazi do slučaja da određeni preklapajući dijelovi rečenice u drugim rečenicama mogu predstavljati sintaktičke jedike, pa model mora odabrati jednu od varijati te odabere krivu. Druga greška koju je model napravio u ovoj rečnici jest što je "misli da" povezao u sintaktičku jedinku. Rečenični fragment "misli da" načelo ne može biti sintaktička jedinka u nekoj rečenici, no u jeziku vrlo se često pojavljuje niz <glagol + "da"> pri čemu se prije glagola i nakon vezika "da" pojavljuju različite vrste riječi. Zbog toga u modelu takav niz poprimi veliku vjerojatnost pa prilikom parsanja to postane sintaktička jedinka.

6. Zaključak

U ovom radu predstavljen je sintaktički analizator hrvatskog jezika temeljen na nenadziranom učenju. Sam analizator temelji se na statističkom modelu nazvanom parsanje temeljeno na podacima (engl. *Data-Oriented Parsing – DOP*). Prema dostupnim informacijama, ovo je prvi pokušaj primjene ovog modela na hrvatski jezik. Korištenje strojnog učenja za izgradnju sustava za sintaktičku analizu prirodnog jezika mnogo je bolje od izgradnje takvog sustava direktnim zapisivanjem pravila od strane eksperta ako se želi izgraditi sustav za analizu cjelokupnog jezika. Ovisno o tome imaju li rečenice na kojima sustav uči označenu sintaktičku strukturu, strojno učenje može biti u nadziranom ili nenadziranom pristupu. Nadzirani pristup općenito postiže bolje rezultate, no nenadzirani pristup je vrlo privlačan jer ne zahtjeva izgradnju banke stabala.

Parsanje temeljeno na podacima je model koji u nenadziranom pristupu prvo izgradi banku stabala, a potom koristeći fragmente tih stabala gradi sintaktička stabla za rečenice koje parsa. U ovom je radu originalnom modelu predloženo nekoliko alternativnih načina označavanja čvorova stabla te je predložen algoritam za minimizaciju skupa pravila kojim je uveliko smanjeno vrijeme parsanja rečenica. Model na hrvatskom jeziku postiže neoznačenu F_1 mjeru nešto manju od 50%, što je manje od rezultata koji su dobiveni za engleski te njemački jezik, a približno jednako rezultatima za kineski jezik.

Performanse modela bi se vjerojatno mogle poboljšati upotrebom POS/MSD označivača, pa bi to bila najvažnija smjernica za eventualni budući rad na ovom modelu. Osim toga, moguće poboljšanje moglo bi se ostvariti nekim novim načinom označavanja unutarnjih čvorova sintaktičkih stabala. Također bilo bi dobro model naučiti te testirati na nekom korpusu koji je detaljno pregledan od strane čovjeka u kojem se nalaze samo ispravne rečenice s jasno vidljivom sintaksom.

LITERATURA

F. Amaya, J.M. Benedí, i J.A. Sánchez. Learning of stochastic context-free grammars from bracketed corpora by means of reestimation algorithms, 1999.

J. Baker. Trainable grammars for speech recognition. 1982.

Rens Bod. A computational model of language performance: Data oriented parsing. U *Proceedings of the 14th conference on Computational linguistics - Volume 3*, COLING '92, stranice 855–859, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/992383.992386. URL `http://dx.doi.org/10.3115/992383.992386`.

Rens Bod. Combining semantic and syntactic structure for language modeling. U *Proceedings ICSLP-2000*, 2000a.

Rens Bod. Parsing with the shortest derivation. U *Proceedings COLING-2000*, 2000b.

Remko Bonnema, Paul Buying, i Remko Scha. A new probability model for data oriented parsing, 1999.

N. Chomsky. Three models for the description of language. *IEEE Transactions on Information Theory*, 2(3):113–124, Rujan 1956. ISSN 0018-9448. doi: 10.1109/TIT.1956.1056813. URL `http://dx.doi.org/10.1109/TIT.1956.1056813`.

N. Chomsky. *Knowledge of Language: Its Nature, Origin, and Use*. Praeger, New York, 1986.

Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

John Cocke i Jacob T Schwartz. *Programming languages and their compilers : preliminary notes / John Cocke and J. T. Schwartz*. Courant Institute of Mathematical Sciences, New York University, New York :, 2d rev. version. izdanju, 1970.

A. P. Dempster, N. M. Laird, i D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

Joshua Goodman, Rens Bod, Remko Scha, R. Bod, i R. Scha. Efficient parsing of dop with pcfg-reductions, 2003.

Mark Johnson. The dop estimation method is biased and inconsistent. *Computational Linguistics*, 28:71–76, 1998.

R. M. Kaplan i J. Bresnan. *Lexical-Functional Grammar: A Formal System for Grammatical Representation*. MIT Press, Cambridge, MA, 1982.

T. Kasami. An efficient recognition and syntax analysis algorithm for context free languages. 1965.

Dan Klein. Corpus-based induction of syntactic structure: Models of dependency and constituency. U *In Proceedings of the 42nd Annual Meeting of the ACL*, stranice 479–486, 2004.

Dan Klein i Christopher D. Manning. A generative constituent-context model for improved grammar induction. U *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, stranice 128–135, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1073083.1073106. URL `http://dx.doi.org/10.3115/1073083.1073106`.

K. Lari i S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

C. Pollard i I. A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, 1994.

Andrew Radford. *Transformational grammar*. Cambridge University Press, 1988.

Remko Scha. Taaltheorie en taaltechnologie: competence en performance. U Q. A. M. de Kort i G. L. J. Leerdam, urednici, *Computertoepassingen in de Neerlandistiek*, LVVN-jaarboek, stranice 7–22. Landelijke Vereniging van Neerlandici, Almere, 1990.

Khalil Sima'an. Computational complexity of probabilistic disambiguation by means of tree-grammars. U *COLING*, stranice 1175–1180, 1996.

Lucien Tesnière. *Éleménts de syntaxe structurale*. Klincksieck, Paris, 1959.

D. Younger. Recognition and parsing of context-free languages in time n3*. *Information and Control*, 10(2):189–208, Veljača 1967. ISSN 00199958. doi: 10.1016/S0019-9958(67)80007-X. URL http://dx.doi.org/10.1016/S0019-9958(67)80007-X.

Sintaktički analizator hrvatskoga jezika temeljen na nenadziranom strojnom učenju

Sažetak

Sintaktička analiza ili parsanje jest postupak analize rečenica prirodnog jezika sa svrhom određivanja njihove strukture u odnosu na određeni skup pravila odnosno formalnu gramatiku. Strojna sintaktička analiza rečenice preduvjet je za više razine strojne obrade teksta, poput semantičke analize ili ekstrakcije informacija. U okviru diplomskog rada proučeni su pristupi parsanju temeljeni na nenadziranom strojnom učenju. Implementiran je model nazvan *parsanje temeljeno na podacima* (engl. *Data-Oriented Parsing – DOP*) koji rečenice parsa pridjeljujući im sastavna stabla (engl. *constituency tree*). Model je naučen te evaluiran na hrvatskom jeziku. U svrhu evaluacije načinjena je manja banka stabala od sto parsanih rečenica. Model na tom skupu postiže neoznačenu F_1 mjeru nešto manju od 50%.

Ključne riječi: sintaktička analiza, parsanje, nenadzirano učenje, hrvatski jezik

Unsupervised Parser for Croatian Language

Abstract

Syntactic analysis or parsing is the process of analysing sentences to determine their grammatical structure with respect to a given set of rules or formal grammar. Syntactic analysis is usually an important part of many natural language processing tasks such as semantic analysis or information extraction. In this thesis, a few approaches to unsupervised syntactic analysis are reviewed. A syntactical analyzer based on a model called *data oriented parsing – DOP* is implemented. This model parses sentences by assigning them constitency structure. The model has been trained and evaluated on Croatian language. The evaluaton is performed on a smaller treebank consisting of hundred expert-parsed sentences. The model achieves unlabeled F_1 measure slightly below 50%.

Keywords: syntactic analysis, parsing, unsupervised learning, Croatian language