UNIVERZITET U BEOGRADU
FILOLOŠKI FAKULTET

Miloš V. Utvić

IZGRADNJA REFERENTNOG KORPUSA SAVREMENOG SRPSKOG JEZIKA

doktorska disertacija

Beograd, 2013.

UNIVERSITY OF BELGRADE
FACULTY OF PHILOLOGY

Miloš V. Utvić

THE CONSTRUCTION OF REFERENCE CORPUS OF CONTEMPORARY SERBIAN

doctoral dissertation

Belgrade, 2013

Mentor:
dr Cvetana Krstev, vanredni profesor, Univerzitet u Beogradu, Filološki fakultet

Članovi komisije:
dr Božo Ćorić, redovni profesor, Univerzitet u Beogradu, Filološki fakultet
dr Duško Vitas, vanredni profesor, Univerzitet u Beogradu, Matematički fakultet
dr Gordana Pavlović Lažetić, redovni profesor, Univerzitet u Beogradu, Matematički fakultet
dr Vesna Polovina, redovni profesor, Univerzitet u Beogradu, Filološki fakultet

Datum odbrane: _____

Naslov disertacije: Izgradnja referentnog korpusa savremenog srpskog jezika

Rezime: U ovom radu se razmatra problem metoda i alata za konstrukciju korpusa savremenog srpskog jezika kao referentnog jezičkog resursa. Rad se sastoji od tri dela.

U prvom delu rada se razmatraju opšta pitanja koja se odnose na definiciju, istorijat, parametre i klasifikaciju korpusa, kao i na korpusnu lingvistiku kao metodologiju istraživanja jezika. Posebna pažnja je posvećena pitanjima reprezentativnosti i balansiranosti korpusa kao uzorka jezika. Takođe je detaljno razmotren i uticaj Interneta, odnosno veba, na kritičko preispitivanje definicije korpusa. Kao parametri korpusa, posebno su analizirani nosač, domen i namena, obim (veličina), period, izvor/medijum, anotacija i višejezičnost. Na osnovu tih parametara su opisane moguće klasifikacije korpusa i posebno su izdvojeni nacionalni korpusi kao opšti, referentni korpusi koji pretenduju da reprezentuju jezik jedne zemlje. Detaljno su analizirani nacionalni korpusi slovenskih jezika. Poseban odeljak je posvećen istorijatu srpske korpusne lingvistike. Na kraju prvog dela rada su navedeni ciljevi rada: razmatranje mogućnosti izgradnje opšteg korpusa srpskog jezika koji bi bio elektronski, dinamički, sinhroni, balansiran, anotiran (morfološki, strukturno, bibliografski), kao i mogućnosti izgradnje pratećih višejezičnih paralelnih korpusa u kojima je srpski izvorni ili ciljni jezik.

U drugom delu rada se opisuju opšte metode i faze u okviru prethodne obrade i analize korpusa. Razmatraju se sledeće radnje neophodne za prethodnu obradu korpusa: prikupljanje, digitalizacija i klasifikacija tekstova za korpus, konverzija korpusnih tekstova u odgovarajući format elektronskog teksta, lingvistička obrada i anotacija elektronskih tekstova za korpus, kao i indeksiranje i kompresija tekstova korpusa. Kada je u pitanju analiza korpusa, detaljno su razmotreni mehanizmi pretrage korpusa, posebno formalizam regularnih izraza, potom konkordance kao metod za vizuelizaciju podataka iz korpusa koji odgovaraju korisnikovom upitu i na kraju osnove statističke analize korpusa. Posebno poglavlje je posvećeno uporednom pregledu različitih sistema integrisanih korpusnih alata, pri čemu su odvojeno analizirani korpusni alati za veb. Sistemi integrisanih korpusnih alata su upoređivani po sledećim parametrima: licenca, platforma, klijent-server arhitektura i veb, ažurnost i

podrška, proširivost, jezički resursi, tipovi pretrage i raspoložive statističke funkcije.

U trećem delu rada je opisan proces konstrukcije Korpusa savremenog srpskog jezika (SrpKor), sa posebnim akcentom na aktuelnu verziju SrpKor2013. Izložen je istorijat SrpKor-a i analizirani njegovi parametri (nosač, domen i namena, obim (veličina), period, izvor/medijum, anotacija, mogućnosti pretrage). Detaljno su opisane faze prethodne obrade SrpKor-a (prikupljanje tekstova, konverzija tekstova u format za čuvanje i indeksiranje, anotacija i indeksiranje). Sve navedene faze su posebno razmotrene za paralelne korpuse kod kojih je izvorni ili ciljni jezik srpski. Takođe su prikazane mogućnosti jednostavne i napredne pretrage korpusa preko upitnog jezika i veb-sučelja.

Na osnovu rezultata izloženih u ovom radu, može se zaključiti da je najveći deo postavljenih ciljeva ispunjen, kao i da su na konkretan način opisani postupci za njihovo ostvarenje. Izgrađen je elektronski sinhroni korpus savremenog srpskog jezika, veličine 122 miliona reči, anotiran bibliografskim informacijama (korpusni tekstovi) i morfološkim informacijama (vrsta reči i lema korpusnih tokena). Korpus je dinamički i tek treba da postigne balansiranost. S obzirom da su utvrđene operativne smernice izgradnje korpusa i razvijeni alati za podršku izgradnji, u budućnosti se može postići i balansiranost ažuriranjem korpusnih tekstova, tj. dodavanjem novih i zamenom postojećih korpusnih tekstova.

Ključne reči: korpusna lingvistika, srpski jezik, računarska lingvistika, obrada prirodnog jezika, morfosintaksička anotacija, referentni korpus

Naučna oblast: _____

Uža naučna oblast: korpusna lingvistika

UDK broj: 81'322(043.3)

Disertation title: The construction of reference corpus of contemporary Serbian

Abstract: The problem regarding the methods and tools to construct a corpus of contemporary Serbian as a reference language resource is considered in this thesis. The thesis consists of three parts.

General questions related to definition, history, parameters and classification of corpora, as well as to corpus linguistics as a methodology in language research, are considered in the first part of the thesis. The special attention is paid to questions regarding representativeness and balance of corpus as a language sample. The affect of Internet/Web on critical review of corpus definition is considered in detail, too. Corpus parameters (storage medium, domain/purpose, size, time span, mode of communication, annotation and multilinguality) are particularly analysed. Possible classifications of corpora, based on these parameters, are described with emphasis on national corpora as general reference corpora which are supposed to represent the national language of a country. National corpora of Slavic languages are analysed exhaustively. A special section is dedicated to the history of Serbian corpus linguistics. The goals of thesis are listed in the end of the first part of the thesis: considering possibilities for construction of general, electronic, dynamic, synchronous, balanced, morphosyntactically and bibliographically-annotated corpus, as well as the possibilities for construction of multilingual parallel corpora with Serbian as source or target language.

General methods and phases of corpus preprocessing and analysis are described in the second part of the thesis. The following activities essential for corpus preprocessing are considered: collecting, digitizing and classifying corpus texts, text capture (conversion to appropriate e-text format), linguistic processing and annotation of corpus texts, as well as the indexing and compression of corpus texts. As for corpus analysis, search mechanisms are considered exhaustively, especially the formalism of regular expressions, then the concordances as a method of visualising the corpus data that correspond to user query, and finally, the basic statistical analysis of corpus. A particular chapter is dedicated to a comparative view of different systems of integrated corpus tools with separate analysis of web-based corpus tools. The criteria for comparison of systems of integrated corpus tools are: licence, plat-

form, client-server architecture and web, software updates and support, extensibility, language resources, types of search and available statistical functionalities.

The process of constructing the Corpus of contemporary Serbian (SrpKor) is described in the third part of the thesis with emphasis to the current version SrpKor2013. The history of SrpKor is presented and SrpKor corpus parameters (storage medium, domain/purpose, size, time span, mode of communication, annotation and search options) are analysed. The phases of SrpKor preprocessing (collecting texts, text conversion to storage format and indexing format, annotation and indexing) are described in detail. All mentioned phases are particularly considered in case of the parallel corpora with Serbian as source or target language.

Based on the results presented in this thesis, it can be concluded that the most part of thesis goals was achieved and that the concrete answers on how to do that were provided. Electronic synchronous corpus of contemporary Serbian is constructed annotated with bibliographical information (corpus texts) and morphological information (part of speech and lemma of corpus tokens). Corpus is dynamic (monitor corpus) and it still needs to become balanced. Considering the fact that operational corpus construction guidelines are determined, and that corpus tools to support the construction are developed, corpus balance can be achieved in the future by updating the corpus texts, i.e. adding new texts and replacing the existing corpus texts.

Keywords: corpus linguistics, Serbian, computational linguistics, natural language processing, morphosyntactic annotation, reference corpus

Research area: _____

Research subarea: corpus linguistics

UDC number: 81'322(043.3)

Sadržaj

Deo I

Uvod

1

Korpusna lingvistika kao metodologija

Leksema **korpus** potiče od latinske imenice *corpus, corporis, n.* sa značenjima *telo*, *celina*, *ukupnost*, *skup*, *stalež*, *zbornik*. U lingvistici se pod korpusom, u najširem smislu, podrazumeva empirijski materijal namenjen istraživanju jezika ([Vitas & Popović, 2003]), dok **korpusna lingvistika** (takođe u najširem smislu) obuhvata istraživanja jezika zasnovana na korpusu.

Termin *korpusna lingvistika* je skovan osamdesetih godina XX veka ([McEnery et al., 2006]) i obično se u literaturi vezuje za prvu knjigu posvećenu toj temi ([Aarts & Meijs, 1984]), mada se pojavljuje i ranije (npr. [Aarts & van den Heuvel, 1982]). Sam Aarts primećuje da je termin „skovan u žurbi" i da „ne predstavlja baš najbolje ime: čudna je to disciplina nazvana po svom glavnom istraživačkom alatu i izvoru podataka" ([Taylor, 2008]). Uprkos tome, međunarodni simpozijum britanskih, holandskih, švedskih i norveških lingvista, održan početkom avgusta 1991. godine u Stokholmu ([Svartvik, 1991]), označio je stvaranje nove zajednice istraživača, korpusnih lingvista, koja je tokom devedesetih godina prošlog veka uspela da učvrsti svoju poziciju mnogobrojnim publikacijama, a od 1996. godine izdaje i svoj časopis *The International Journal of Corpus Linguistics*. Pa ipak, među samim korpusnim lingvistima ne postoji dogovor šta korpusna lingvistika predstavlja. Autori definicija, pored toga što daju svoje mišljenje, uglavnom odbacuju ranije alternativne pokušaje definisanja, pa se tako korpusna lingvistika tretira kao „alat, metod, metodologija,

metodološki pristup, disciplina, teorija, teoretski pristup, paradigma (teorijska ili metodološka), ili kao kombinacija navedenog" ([Léon, 2005]). Korpusna lingvistika se danas pre smatra za metodologiju ili skup metodologija nego za posebnu teorijsku disciplinu u okviru lingvistike ([McEnery et al., 2006]), ali i dalje ostaje otvoreno pitanje da li predstavlja i nešto više ([Taylor, 2008]).

1.1 Istorijat korpusa

Kada se govori o istoriji nastanka i razvoja korpusa, obično se razmatraju vremenski periodi koje obeležavaju:

- Preelektronski korpusi
- Kritika Čomskog
- Elektronski korpusi I generacije
- Elektronski korpusi II generacije (savremeni elektronski korpusi)

Preelektronski korpusi

Iako je naziv *korpusna lingvistika* u upotrebi tek nekoliko decenija, primena korpusa kao metodologije ima dugu tradiciju u raznim oblastima lingvistike [McEnery et al., 2006]. Takođe, mada se danas se pod korpusom uglavnom podrazumevaju elektronski korpusi, korpusi su korišćeni u lingvistici i mnogo pre pojave računara.

Tako se za neke od najstarijih poznatih gramatika može smatrati da su zasnovane na korpusu ([Meyer, 2008: 3]):

- Paninijeva gramatika sanskritskog, nastala između VI i IV veka p.n.e, pored ostalog opisuje i jezik Veda kojim se u tom trenutku više ne govori, i koji je dostupan samo preko kolekcije sačuvanih tekstova.
- Aristonik, aleksandrijski filolog i komentator Ilijade iz I veka n.e, u šest knjiga pod nazivom *O negramatičkim rečima* izlaže svoj rad o neregularnim gramatičkim konstrukcijama na osnovu „korpusa" Homerovih tekstova.

Paninijeva gramatika nije usamljen primer upotrebe korpusa u izučavanju tzv. mrtvih jezika (jezika koji su prestali da predstavljaju svakodnevno sredstvo komunikacije u govornoj zajednici poput starogrčkog i latinskog danas), kao i za opis živih, prethodno nezapisanih ili nepoznatih jezika (npr. jezika američkih Indijanaca početkom XX veka). Sačuvane tekstove ili kolekcije tekstova, kao empirijski materijal za istraživanja, koristi istorijska (dijahrona) lingvistika za proučavanje pojedinih jezika i jezičkih grupa iz perspektive njihovog razvoja ([Malmkjær, 2001], [Bugarski, 1995: 11]). Posebno su u XIX veku razvijene tehnike za rekonstrukciju starih (mrtvih) jezika, kao i za prepoznavanje veza između različitih jezika, koje se koriste i danas ([Lüdeling & Kytö, 2008: vi]).

Pod **preelektronskim korpusima** se podrazumevaju korpusi nastali pre 1960. godine, tj. kako korpusi nastali pre pojave prvih elektronskih računara sredinom XX veka, tako i korpusi koji su nastali posle toga, a nisu u mašinski čitljivom obliku, tj. obliku koji omogućava obradu pomoću računara ([Kennedy, 1998]). Posebno obimna literatura postoji o preelektronskim korpusima engleskog jezika ([Francis, 1992; Kennedy, 1998; Malmkjær, 2001; McEnery & Wilson, 2001; Meyer, 2008; Stubbs, 2004]), gde se navode sledeća glavna polja lingvistike sa bogatom tradicijom analize zasnovane na korpusu:

1. gramatika
2. leksikografija
3. dijalektologija
4. izučavanje Biblije i književnosti
5. usvajanje jezika i jezičko obrazovanje (uključujući i maternji i strane jezike)
6. komparativna lingvistika.

Preelektronski korpusi su uglavnom kreirani u svrhu istraživačkih projekata sa specifičnim ciljem, tj. retko zarad proizvoljnih (opštih) lingvističkih istraživanja. Proces kreiranja preelektronskih korpusa je često zahtevao mukotrpni i dugotrajni rad. Navešćemo nekoliko primera.

Dominikanski fratri u XIII veku, potpomognuti brojnim pomoćnicima, ručno su indeksirali stranice Biblije, tj. alfabetski suređali reči iz Biblije i uz svaku navodili brojeve pasusa u kojima su se pojavile, omogućivši detaljnu pretragu reči i fraza ([Krstev & Gucul, 2007; McCarthy & O'Keeffe, 2010]). Ovo je samo jedan od primera izrade biblijskih konkordanci (v. odeljak 3.2) na latinskom (XIII vek), hebrejskom (XV vek) i engleskom jeziku (XV i XVIII vek), koje se ponekad smatraju za „... prve značajne deliće istraživanja vezanih za lingvistiku koja su zasnovana na korpusu..." ([Kennedy, 1998: 13]).

Tokom poslednje decenije XIX veka nemački istraživač J. V. Keding (J. W. Kaeding) je sproveo istraživanje radi prikupljanja statističkih podataka o korišćenju pojedininačnih slova i reči u nemačkom jeziku. Rezultate tog istraživanja je trebalo iskoristiti da se usavrši obuka stenografa koji su zapisivali diskusije tokom poslovnih i državnih sastanaka. Prikupljanje podataka za korpus od 11 miliona reči i njihova analiza trajali su godinama, a Kedingu je pomagalo preko pet hiljada saradnika ([Bongers, 1947]).

Jedan od najpoznatijih primera iz XX veka je, svakako, rad na monumentalnoj *Gramatici savremenog engleskog* (eng. *A Modern English Grammar on Historical Principles*) danskog profesora Ota Jespersena (Otto Jespersen) koji je trajao od 1909. do 1949. godine ([Lindquist, 2009]). Uz svaku razmatranu gramatičku konstrukciju Jespersen navodi autentične primere koji predstavljaju samo deo engleskih rečenica zabeleženih tokom njegovog intenzivnog proučavanja engleske književnosti. U svojoj autobiografiji on opisuje kako se njegova ogromna vila pored Kopenhagena postepeno popunjavala kutijama za cipele koje su sadržale stotine hiljada papirnih kartica („listića") na koje je beležio primere engleskih gramatičkih konstrukcija.

Pre pojave elektronskih računara, a čak i više decenija posle toga, ručno izvlačenje podataka iz teksta i njihovo beleženje na karticama predstavljalo je uobičajeno sredstvo za prikupljanje informacija u formi lingvističkih opisa. Taj metod je verovatno najduže opstao u leksikografiji. Kao ilustracija prvih primena korpusa u leksikografiji najčešće se navodi *Rečnik engleskog jezika* Samjuela Džonsona (Samuel Johnson), štampan u dva toma 1755. godine, čiji je „... uticaj na potonju leksikografiju neprevaziđen..." ([Kristal, 1996]), mada postoje potvrde da su još u XVI

veku engleski leksikografi koristili citate za ilustraciju upotrebe pojedinih leksema. Ono po čemu je Džonsonov rečnik specifičan u odnosu na prethodne jeste intenzivno korišćenje takvih citata, tj. korpusa od približno 150.000 primera jezičke upotrebe ([Francis, 1992]), pri čemu je polovina svih citata uzeta iz dela „najboljih" engleskih pisaca iz perioda od 1560. do 1660. godine (Šekspira (W. Shakespeare), Drajdena (J. Dryden), Miltona (J. Milton), Adisona (J. Addison), Bejkona (F. Bacon), Poupa (A. Pope)) i iz Biblije.

Drugi značajan primer primene preelektronskog korpusa u leksikografiji je korpus korišćen u izradi *Oksfordskog rečnika engleskog jezika* (eng. *Oxford English Dictionary*, skr. **OED**). Izrada ovog rečnika je trajala od 1857. godine punih sedam decenija, da bi poslednji, dvanaesti tom bio izdat 1928. godine ([Kennedy, 1998: 14–15]). Oko 2 hiljade volontera je prikupilo skoro 5 miliona primera ([Francis, 1992]), najvećim delom iz pisanih engleskih književnih tekstova, sa ciljem da se u rečniku nađe svaka reč upotrebljena u engleskom u periodu od 1250. do 1858. godine.

Upravo su primene korpusa u leksikografiji, a potom i rad američkih strukturalnih (deskriptivnih) lingvista, predvođenih Leonardom Blumfildom (Leonard Bloomfield) i njegovim sledbenicima, pre svih, Zeligom Harisom (Zellig Harris), doprineli da korpus dobije primarnu ulogu u lingvističkom istraživanju sredinom XX veka. Za deskriptivne lingviste korpus je „polazna tačka lingvističke analize" ([Karlsson, 2008]), „ne samo nezamenljivi alat već neophodan uslov za naučni opis" ([Malmkjær, 2001: 85]). Malmkjaer opisuje dva prelaza koji karakterišu ovaj period:

- „prelaz sa zatvorenog korpusa mrtvog jezika na zatvoreni i konačni korpus živog jezika", tj. prešlo se na izučavanje jezika u upotrebi kao sredstva za komunikaciju u govornoj zajednici.

- „prelaz sa pisanih tekstuelnih podataka o mrtvim jezicima na govorne tekstuelne podatke o živim, do tog trenutka nezapisanim jezicima".

Kritika Čomskog

Knjiga *Sintaksičke strukture* ([Chomsky, 1957]) predstavlja jednu od bitnih prekretnica u lingististici. Učenik Zeliga Harisa, u početku i sam zastupnik struktu-

ralizma, Noam Čomski (Noam Chomsky) u ovoj knjizi iznosi nove ideje koje će narednih godina preusmeriti pažnju lingvista sa empirizma na racionalizam i tako potisnuti deskriptivnu lingvistiku. Između ostalog, Čomski uvodi pojmove **jezička sposobnost** (eng. **linguistic competence**) i **govorna delatnost** (eng. **linguistic performance**). Jezička sposobnosti predstavlja „čovekovo interno znanje o jeziku", tj. sposobnost da razume i proizvede jezik, dok je govorna delatnost samo „bleda slika" (spoljašnja manifestacija) tog znanja. Za razliku od deskriptivnih lingvista koji se bave izučavanjem govorne delatnosti, Čomski smatra da je pravi zadatak lingvistike da istraži jezičku sposobnost.

Čomski strogo formalno zasniva **transformaciono-generativnu gramatiku** i formalni jezik kao skup rečenica koje se mogu generisati takvom gramatikom, a potom klasifikuje formalne jezike prema njihovoj generativnoj moći (hijerarhija Čomskog, v. npr. [Vitas, 2006]). Tom prilikom Čomski primećuje da struktura rečenica prirodnog jezika može biti rekurzivna, što neminovno dovodi do tvrđenja da postoji beskonačno mnogo takvih rečenica, suprotno dotadašnjim pretpostavkama deskriptivnih lingvista da je broj rečenica prirodnog jezika konačan i da se one mogu prikupiti i nabrojiti ([McEnery & Wilson, 2011]).

Čomski dalje primećuje da korpus, pošto je po svojoj prirodi konačan skup primera govorne delatnosti, ne može predstavljati osnovu za opis beskonačnih mogućnosti generisanja prirodnog jezika i potrebno je zameniti ga intuicijom govornika kao jedinim pouzdanim izvorom podataka o jeziku ([Kennedy, 1998]). „Neke rečenice se neće pojaviti jer su očigledne, druge zato što su neistinite, a ostale pak jer su nepristojne.", navodi Čomski 1958. godine u prilog svojoj tezi ([Leech, 1991]).

Naknadno je Čomski kritikovao i relevantnost statističke analize učestanosti pojavljivanja lingvističkih elemenata na osnovu korpusa, okarakterisavši činjenicu da rečenica „Ja živim u Njujorku" ima veću učestanost od rečenice „Ja živim u Dejtonu, Ohajo" kao nerelevantnu za lingvističku teoriju ili opis ([Chomsky, 1962]).

Elektronski korpusi I generacije

Od samog nastanka prvih elektronskih računara sredinom četrdesetih godina XX veka, ulažu se napori da se računar primeni u obradi prirodnih jezika. Već 1948.

godine u SAD je pokrenut projekat sa ciljem da se realizuje automatsko prevođenje s jednog prirodnog jezika na drugi. Iako taj prvi projekat nije dao zadovoljavajuće rezultate, njegov neostvareni cilj ni do danas nije prestao da daje snažan podsticaj razvoju računarske lingvistike[1].

Tokom pedesetih godina XX veka, dok su deskriptivni lingvisti još uvek imali glavnu reč, istraživači su pokušavali da razviju metode za automatsko „učenje" morfoloških, sintaksičkih i drugih lingvističkih pravila analizom korpusa. Za ispitivanje strukture jezika najčešće je korišćena metoda raspodele reči i fraza na osnovu sličnosti okruženja (konteksta) u kome se pojavljuju u korpusu ([Brill & Mooney, 1998]). Međutim, uticaj Čomskog posle *Sintaksičkih struktura* i njegova kritika korpusa (v. str. 5) dovode do masovnog odbacivanja korpusa u lingvistici i zamene intuicijom.

Iako su Čomski i njegovi sledbenici otkrili niz novih činjenica o jeziku, kada je lingvistička metodologija u pitanju, oni su zapravo zamenili jedno ekstremno stanovište drugim: dok su deskriptivni lingvisti, sledbenici Zeliga Harisa, smatrali da je „korpus dovoljan za sve", dotle su pristalice Čomskog odbacivale sve sem intuicije. Međutim, između dva suprotstavljena tabora, iako je uticaj Čomskog sve više rastao, ostali su (istina retki) lingvisti koji nisu potpuno odbacivali ni korpus ni intuiciju, već su pokušavali da iskoriste prednosti i jednog i drugog.

I upravo u takvim okolnostima je početkom šezdesetih godina XX veka u SAD nastao prvi elektronski korpus, *Standardni korpus savremenog američkog engleskog jezika Univerziteta Braun* (eng. *The Brown University Standard Corpus of Present-Day American English*), danas poznatiji pod skraćenim nazivom **Braunov korpus**. Tvorci Braunovog korpusa su Henri Kučera (Henry Kučera) i Vintrop Nelson Fransis (Winthrop Nelson Francis). Evo šta o tome kažu sami autori:

> „Kada sam 1962. godine bio u početnoj fazi prikupljanja Braunovog standardnog korpusa američkog engleskog, upoznao sam profesora Roberta Liza (Robert Lees) na jednoj lingvističkoj konferenciji. Na njegovo

[1] Kao što primećuje Vitas (2010, „termin *računarska lingvistika* nije adekvatan prevod za englesko *computational linguistics*. Naime, računarska lingvistika se bavi izgradnjom formalnih modela (konkretnih) prirodnih jezika što kao posledicu može, a ne mora, imati računarsku primenu. U tom svetlu, korektniji termin bi bio *izračunljiva lingvistika* ili *lingvistika izračunljivog*.")

pitanje o mojim interesovanjima odgovorio sam da sam dobio sredstva od Biroa za obrazovanje SAD da napravim korpus od milion reči savremenog američkog engleskog koji bi se koristio na računaru. Pogledao me je sa zaprepašćenjem i pitao, 'Za ime sveta, zašto to radite?'. Odgovorio sam nešto o pronalaženju pravih činjenica o gramatici engleskog. Nikad neću zaboraviti njegov odgovor: 'To je potpuno gubljenje Vašeg vremena i vladinog novca. Vi ste izvorni govornik engleskog; za deset minuta možete proizvesti više ilustracija ma kog dela engleske gramatike nego što ćete naći među milionima reči slučajnog teksta.'"([Francis, 1982])

„Friman Tvadel (Freeman Twadell) je predložio da bi bilo korisno da se kompilira reprezentativni računarski korpus savremenog američkog engleskog jezika. Nelson Fransis je pribavio za to sredstva, i mi smo započeli sa pripremama [...]

Sa današnjeg stanovišta korpus od milion reči deluje kao skromna vežba. Međutim, početkom šezdesetih godina XX veka to je bio i tehnološki i lingvistički izazov. Računari tog vremena su bili ogromne mašine u klimatizovanim prostorijama, ali sa ograničenim mogućnostima izračunavanja i majušnom unutrašnjom memorijom. Računar sa kojim smo bili prinuđeni da radimo, IBM 7070, imao je memoriju kapaciteta 50 kilobajta. Sve podatke i programe trebalo je uneti ili na bušenim karticama ili na papirnoj traci. Tek posle toga su informacije mogle da se prenesu na magnetne trake radi dalje obrade. Bilo je potrebno više od godinu dana da se, pod energičnim vođstvom Nelsona Fransisa, otkucaju podaci, obavi celokupna provera i da se korpus sačuva na magnetnim trakama. Kada je došlo vreme da se odrede [...] svojstva korpusa, između ostalog i sortiranje milion slogova (zapisa), bilo nam je potrebno četrnaest sati neprekidnog predanog rada na milion dolara vrednom računaru sa šest magnetnih uređaja kako bismo konstruisali prvu listu reči.

Lingvistički izazov se sastojao u tome da korpus što više reprezen-

> tuje trenutni pisani američki engleski jezik. To je zahtevalo da se ustanovi ukupan broj uzoraka tekstova, njihove veličine i raspodele među raznim žanrovima pisanog teksta. Ove odluke su donete na maloj konferenciji lingvista na Univerzitetu Braun 1963. godine. Rezultat je bila baza sastavljena od 500 uzoraka teksta, svaki približne veličine 2.000 reči, podeljenih u 15 žanrova proporcionalno procenjenoj upotrebi [...] Pojedinačni uzorci su birani slučajnim metodom iz raspoloživih izvora tekstova koji su prvi put odštampani 1961. godine. U većini slučajeva, bila je neophodna dozvola nosioca autorskih prava, i to je bio zadatak koji je zahtevao od Nelsona Fransisa da se uključi u obimnu prepisku, kao i objašnjenja da je korpus naučni poduhvat i da nemamo sredstava da platimo naknadu. Lepo je primetiti da su teškoće ove prirode uglavnom prevaziđene, i da je ogromna većina kontaktiranih autora i izdavača dobrovoljno dala svoju dozvolu." ([Kučera, 2002: 307–308])

U detaljnom uputstvu ([Francis & Kučera, 1964]) se navode kriterijumi za odabir uzoraka tekstova: svaki uzorak počinje od početka rečenice teksta (ali ne nužno od prve rečenice pasusa ili neke veće strukturne jedinice teksta), a završava se na kraju prve rečenice teksta koja se završava posle 2.000 reči. U slučaju 18 uzoraka se odstupilo od tog kriterijuma, tako da su neki uzorci kraći od 2.000 reči, dok nekoliko uzoraka sadrži dodatne rečenice posle propisane granice. Prosečna dužina uzorka je $2.028,6$. Tabela 1.1 prikazuje raspodelu uzoraka tekstova za Braunov korpus.

Ovde treba spomenuti i poslednji veliki preelektronski korpus (britanskog) engleskog jezika koji je koncipiran na sličnim kriterijumima, *Istraživanje upotrebe engleskog* (eng. *Survey of English Usage*, skr. **SEU**). Rad na ovom korpusu ([SEU, 2010]), pod rukovodstvom Randolfa Kverka (Randolph Quirk), trajao je između 1955. i 1985. godine, a korpus je danas poznatiji pod nazivom svoje elektronske verzije, *Korpus London-Lund* (eng. *London-Lund Corpus*, skr. **LLC**), a u literaturi se na njega referiše i kao „Kverkov korpus". Veličina korpusa SEU je milion reči (200 uzoraka pisanog teksta i transkribovanog govora, svaki veličine 5.000 reči) Za razliku od Braunovog korpusa koji se uglavnom oslanja na pisane tekstove, korpus SEU obuhvata i govorne i pisane tekstove, približno u istoj količini. Govornim tekstovima su

Tabela 1.1: Sadržaj Braunovog korpusa ([Lindquist, 2009]

	Tip teksta	**Broj tekstova**	**Udeo u korpusu (%)**
A	Štampa: reportaže (politika, sport, društvo, finansije, kultura, sa lica mesta)	44	8,8
B	Štampa: uređivanje (uključujući pisma uredniku)	27	5,4
C	Štampa: pregledi (pozorište, knjige, muzika, ples)	17	3,4
D	Religija	17	3,4
E	Veštine i hobi	36	7,2
F	Tradicionalno znanje	48	9,6
G	Beletristika, biografije, memoari	75	15,0
H	Razno (uglavnom državni dokumenti)	30	6,0
J	Nauka (akademski tekstovi)	80	16,0
K	Opšta fikcija (romani i kratke priče)	29	5,8
L	Misterije i detektivska fikcija	24	4,8
M	Naučna fantastika	6	1,2
N	Avanturistička i vestern-fikcija	29	5,8
P	Romanse i ljubavne priče	29	5,8
R	Humor	9	1,8
	Ukupno nefikcije	374	75
	Ukupno fikcije	126	25
	Ukupno	500	100

obuhvaćeni i dijalozi i monolozi, dok se među pisanim tekstovima, pored štampanog materijala i rukopisa, nalaze i primeri engleskog jezika čitani naglas (radijske vesti i napisani govori).

Korpus LLC je, zapravo, rezultat dva projekta, SEU i projekta *Istraživanje govornog engleskog* (eng. *Survey of Spoken English*, skr. **SSE**). Veličina korpusa LLC je 500.000 reči, tj. on predstavlja polovinu korpusa SEU koju čini 100 uzoraka transkribovanog govora. Izrada korpusa LLC je započela 1975. godine, a prva dostupna verzija se pojavila početkom 1980. godine.

Braunov korpus je poslužio kao uzor mnogim potonjim elektronskim korpusima, pre svega, u pokušaju da se napravi balansiran korpus (v. odeljak 1.2), tj. korpus

koji bi uistinu odslikavao celinu jezika. Na pomenutoj „maloj konferenciji lingvista na Univerzitetu Braun 1963. godine" ([Kučera, 2002]) učestvovao je i Randolf Kverk (u svojstvu rukovodioca projekta SEU), kao i eminentni američki lingvisti, između ostalih urednik *Trećeg međunarodnog Vebsterovog rečnika* Filip B. Gov (Philip B. Gove).

Primenom istih kriterijuma, samo drugog materijala, napravljeni su *Korpus Lankaster-Oslo/Bergen* (eng. *The Lancaster-Oslo/Bergen Corpus*, skr. **LOB**) i tzv. *Frajburški korpusi* (eng. *Freiburg corpora*). LOB je britanska verzija Braunovog korpusa (vremenski okvir uzoraka je takođe 1961. godina), dok su Frajburške verzije Braunovog korpusa (FROWN) i korpusa LOB (FLOB) napravljene skoro trideset godina kasnije i zasnovane su na američkim, odnosno britanskim materijalima iz 1991. godine ([Lindquist, 2009]).

Elektronski korpusi druge generacije

Uprkos pojavi elektronskih korpusa, tokom šezdesetih i sedamdesetih godina XX veka, njihov uticaj na lingvistiku generalno nije bio veliki, pre svega zbog dominacije generativne lingvistike Čomskog, tako da je rad sa korpusom bio rezervisan za uske krugove lingvista koji su radili gotovo „u tajnim ćelijama" ([Lindquist, 2009]). Ipak, nekolicina projekata vezanih za istraživanje engleskog jezika, kreirala je sopstvene korpuse za posebne primene u raznim oblastima lingvistike: leksikografiji, usvajanju jezika (maternjeg i stranog), istorijskoj lingvistici, sociolingvistici, itd. ([Kennedy, 1998: 33–45]).

S druge strane, te decenije je obeležio snažan razvoj računarskih tehnologija, koji će osamdesetih godina omogućiti da računari postanu široko dostupni. Pojavili su se novi spoljašnji memorijski medijumi, optički diskovi (CD-ROM), koji su, u odnosu na svoje prethodnike, omogućili skladištenje mnogo više podataka po mnogo manjoj ceni. Takođe, početkom osamdesetih godina počinje masovna proizvodnja skanera koji, kao ulazni uređaji računara, omogućavaju jednostavniji i brži unos podataka (v. odeljak 2.1), a time i uslove za kreiranje korpusa veličine nekoliko desetina miliona reči.

Među prvim lingvistima koji su iskoristili pojavu elektronskih korpusa i nagli

razvoj računarskih tehnologija bili su leksikografi. Već šezdesetih godina XX veka američki leksikografi su pokušali da iskoriste Braunov korpus za izradu *Rečnika američkog nasleđa* (eng. *American Heritage Dictionary*, skr. **AHD**). Međutim, zbog svoje nedovoljne veličine, Braunov korpus nije mogao značajno da doprinese obradi većine odrednica: blizu polovine ukupnog broja reči predstavljaju hapax legomena, tj. reči koje se pojavljuju samo jednom, dok se oko 85% svih reči pojavljuje svega nekoliko puta ([Johansson, 2008]). Tokom 1980. godine započet je projekat *Međunarodna jezička baza Kolinsa i Univerziteta u Birmingemu* (eng. *Collins Birmingham University International Language Database*, skr. **COBUILD**). Cilj projekta COBUILD je bila izgradnja višemilionskog korpusa engleskog jezika, koji bi poslužio kao osnova za izradu novog rečnika engleskog jezika. Projekat je ostvaren kao saradnja privatne izdavavačke kuće Kolins i Odeljenja za engleski jezik Univerziteta u Birmingemu. Na čelu projekta je bio Džon Sinkler (John Sinclair), profesor savremenog engleskog jezika na Univerzitetu u Birmingemu. Do avgusta 1982. godine glavni deo korpusa je sadržao $7,3$ miliona reči, da bi, posle neprekidnog dopunjavanja, u trenutku izdavanja rečnika zasnovanog na korpusu (Collins Cobuild English Language Dictionary) 1987. godine, veličina korpusa bila oko 20 miliona reči ([Sinclair, 1987]).

Krajem osamdesetih godina XX veka započeta je izgradnja još jedne komercijalne baze podataka za potrebe leksikografije u organizaciji izdavačke kuće Longman i Univerziteta u Lankasteru. Izvršni rukovodioci projekta su bili Dela Samers (Della Summers), ispred kuće Longman, i Džefri Lič (Geoffrey Leech), profesor opšte lingvistike i savremenog engleskog jezika na Univerzitetu u Lankasteru, koji je prethodno bio među rukovodiocima izgradnje korpusa LOB (v. 1.1, str. 11). Rezultat su tri korpusa:

- *Korpus engleskog jezika Longman/Lankaster* (eng. *Longman/Lancaster English Language Corpus*, skr. **LLELC**), od približno 30 miliona reči;
- *Longmanov govorni korpus* (eng. *Longman Spoken Corpus*, skr. **LSC**), od približno 10 miliona reči;
- *Longmanov korpus engleskog kao nematernjeg jezika* (eng. *Longman Corpus*

of Learners' English, skr. **LCLE**), od približno pet miliona reči.

Ova dva poduhvata su predstavljala samo uvod u razvoj „mega-korpusa" ([Kennedy, 1998]). U periodu od 1991. do 1995. godine nastao je *Britanski nacionalni korpus* (eng. *British National Corpus*, skr. **BNC**), verovatno najuticajniji savremeni korpus. Iako je u pitanju korpus engleskog jezika od 100 miliona reči, principi na kojima je izgrađen (reprezentativnost, balansiranost, veličina 1.2, v. odeljak 1.2, str. 14) trebalo je da posluže „kao uzor sličnim poduhvatima izgradnje nacionalnih korpusa u drugim zemljama i za druge jezike" ([Kennedy, 1998]). Iza projekta su stale vodeće akademske institucije, privatni izdavači, državne institucije Velike Britanije, uključujući i britansku vladu koja je snosila polovinu ukupnih troškova. Naknadno nisu dodavani novi tekstovi, ali su se pojavile još dve verzije ovog korpusa, *BNC World* 2001. godine i *BNC XML Edition* 2007. godine ([Burnard, 2007]). Odvojeno su izdata i dva podkorpusa: *BNC Sampler* (kolekcije pisanih i govornih tekstova, svaka od po milion reči) i *BNC Baby* (četiri različita žanra, svaki predstavljen sa po milion reči).

Najveći deo korpusa BNC čine pisani tekstovi (90%), dok su govorni tekstovi preuzeti iz Longmanovog govornog korpusa (LSC). Za izbor pisanih tekstova, kao kriterijumi, korišeni su „domen" (tip sadržaja, tematika), „vreme" (kada je tekst proizveden) i „medijum" (tip publikacije: monografska, periodika, itd.). Pri odabiru govornih tekstova korišćena su dva kriterijuma, „demografski" i „vođen kontekstom"[2] ([Xiao, 2008]). Tekstovi prikupljeni po „demografskom" kriterijumu predstavljaju transkribovani govor snimljen prilikom neformalnih susreta svakog od 124 volontera, izabranih prema polu, starosti, geografskom regionu i socijalnoj grupi. U slučaju kriterijuma „vođenog kontekstom", biran je formalan govor snimljen tokom radioprenosa, stručnih predavanja, poslovnih sastanaka, itd. Raspodela pisanih i govornih tekstova je prikazana u tabelama 1.2 i 1.3 ([Aston & Burnard, 1998]). Trenutno, Britanski nacionalni korpus sadrži 4.124 teksta veličine do 40 hiljada reči.

Kao što je već spomenuto (str. 1), početkom avgusta 1991. godine u Stokholmu je održan skup na kome je zvanično promovisan rad nove zajednice istraživača, zajednice korpusnih lingvista ([Svartvik, 1991]). Inicijator skupa je bila *Međunarodna*

[2]U originalu „context-governed".

Tabela 1.2: Raspodela pisanih tekstova u Britanskom nacionalnom korpusu

Domen	%	**Datum**	%	**Medijum**	%
Fikcija	21,91	1960–74	2,26	Monograf. publikacije	58,58
Umetnost	8,08	1975–93	89,23	Periodika	31,08
Vera i misao	3,40	Neklasifikovano	8,49	Razno (objavljeno)	4,38
Trgovina/finansije	7,93			Razno (neobjavljeno)	4,00
Slobodno vreme	11,13			Govori	1,52
Prirodne/čiste nauke	4,18			Neklasifikovano	0,40
Primenjene nauke	8,21				
Društvene nauke	14,80				
Svetska politika	18,39				
Neklasifikovano	1,93				

računarska arhiva savremenog i srednjevekovnog engleskog jezika (eng. *International Computer Archive of Modern and Medieval English*, skr. **ICAME**). Nastala još 1979. godine, ova zajednica je tokom više od jedne decenije imala redovne godišnje, gotovo tajne sastanke, na kojima se diskutovalo o aktuelnim pitanjima primene (elektronskih) korpusa u istraživanju, pre svega, engleskog jezika.

Tabela 1.3: Raspodela govornih tekstova u Britanskom nacionalnom korpusu

Region	%	**Tip interakcije**	%	**Vođen kontekstom**	%
Jug	45,61	Monolog	18,64	Obrazovni/informativni	20,56
Centralni deo	23,33	Dijalog	74,87	Poslovni	21,47
Sever	25,43	Neklasifikovano	6,48	Institucionalni	21,86
Neklasifikovano	5,61			Slobodno vreme	23,71
				Neklasifikovano	12,38

Ubrzani razvoj informatičke tehnologije i pojava Interneta tokom devedesetih godina XX veka, bitno su uticali i na razvoj korpusne lingvistike. Pojavljuju se postepeno korpusi gotovo svih svetskih jezika, napravljeni po različitim kriterijumima i za različite namene, a dobar deo postaje javno dostupan preko Interneta, kao i razni programski paketi za kreiranje i pretraživanje korpusa. Nemoguće je navesti popis svih postojećih korpusa, a kamoli dati njihov detaljan pregled, tako da će biti spomenuti samo oni korpusi koji se neposredno tiču teme ovog rada. Međutim, pre toga neophodno je preciznije definisati korpus (u skladu sa savremenim stavovima korpusne lingvistike), detaljno objasniti njegove parametre, kao i tipove korpusa koji proizilaze iz različitih vrednosti tih parametara.

1.2 Aktuelne definicije i reprezentativnost korpusa

S obzirom da su uglavnom svi savremeni korpusi elektronski, korpusni lingvisti u svojim definicijama uglavnom poistovećuju *korpus* i *elektronski korpus*[3]. Najveći broj aktuelnih definicija korpusa slaže se da je u pitanju „kolekcija autentičnih mašinski čitljivih tekstova koji predstavljaju reprezentativni uzorak pojedinačnog jezika ili jezičkog varijeteta" ([McEnery et al., 2006]). Da bi prethodna definicija bila prihvaćena, mora se precizirati:

(R1) šta je ciljna populacija iz koje se bira uzorak, tj. tekstualni univerzum čiji elementi su kandidati za uključivanje u korpus;

(R2) šta znači „reprezentativni uzorak";

(R3) koje veličine treba da bude uzorak (korpus);

(R4) po kom metodu treba birati tekstove uzorka, tj.:

a) koliki broj tekstova uključiti u korpus;

b) kako birati pojedinačne tekstove koji će biti obuhvaćeni korpusom;

c) da li birati celovite tekstove ili odlomke iz tekstova (veličina jediničnog uzorka);

d) ako se biraju odlomci, kako ih birati (sa početka, kraja ili iz sredine teksta, jednake ili nejednake dužine, koje dužine, itd.).

Odgovori na ova pitanja određuju kriterijume za ocenu reprezentativnosti korpusa, i stoga su od izuzetnog značaja, pre svega za praktičnu konstrukciju korpusa. Zato ne iznenađuje što se na tu temu vodi višedecenijska polemika. Mogu se navesti primeri koji pokazuju da je reprezentativnost korpusa relativan, nepostojan koncept koji zavisi od „pitanja koja zanimaju istraživača u trenutku kada formira ili razmišlja da koristi korpus" ([Xiao, 2010: 151]), što se zapravo svodi na to da populacija koju korpus pretenduje da predstavi prvenstveno utiče na to koliko će on biti

[3]Stoga će i u ostatku ovog teksta pojmovi *korpus* i *elektronski korpus* biti tretirani kao sinonimi.

reprezentativan. Tako jedan te isti korpus, sastavljen, na primer, od izvesnog broja dela Ive Andrića, ali tako da obuhvata primerke tekstova svih žanrova i vremenskih perioda u kojima je pisac stvarao, makar bio i „idealno reprezentativan" u odnosu na celokupno piščevo stvaralaštvo, i dalje je samo reprezentativan za istraživače koje interesuje jezik Andrićevih dela, ali daleko od toga da može da reprezentuje jezik srpske književnosti XX veka. Upravo je namena većine kreiranih korpusa diktirala kako su njihovi tvorci odgovorili na pitanja (R1)–(R4).

Ovde se neizostavno mora spomenuti Daglas Bajber (eng. Douglas Biber), američki lingvista koji je među prvima pokušao da opiše operativni postupak za postizanje reprezentativnosti uzorka prilikom konstrukcije korpusa, detaljno razmatrajući pitanja (R1)–(R4). Detaljnost analize reprezentativnosti korpusa u Bajberovim radovima doprinela je da se svi kasniji radovi drugih istraživača na tu temu referišu na njegove radove (prevashodno [Biber, 1993]).

Definicija ciljne populacije

Odgovor na pitanje (R1) je preduslov za modeliranje korpusa jer se njime precizira *šta* korpus zapravo treba da predstavi. Prema Liču, „celokupni tekstuelni univerzum S-jezika[4] (...) jeste populacija iz koje se bira uzorak" ([Leech, 2007: 3]). Međutim, u praksi se najpre ta ogromna populacija ograničava tako što se formira **okvir uzorkovanja/okvir uzorka** (eng. **sampling frame**) iz koga se zatim biraju **jedinični uzorci/jedinice uzorka** (eng. **sampling unit**) kao elementi uzorka ([Biber, 1993]). Okvir uzorkovanja se svodi na konačnu numerisanu listu (popis, spisak) svih potencijalnih jediničnih uzoraka sa pridruženim identifikatorima. Tako je u slučaju Braunovog korpusa populacija ograničena na štampane materijale na američkom engleskom jeziku, prvi put izdate u SAD 1961. godine, a okvir uzorkovanja su zapravo materijali sa navedenim svojstvima koji su u vreme kreiranja korpusa bili dostupni u biblioteci Univerziteta Braun i Ateneumu Providensa (eng. Providence Athenaeum).

Međutim, ponekad je izuzetno teško definisati populaciju, a samim tim i formi-

[4]Skr. od *spoljašnji jezik*, termin koji Čomski koristi kao sinonim za govornu delatnost ([Chomsky, 1987: 45]).

rati odgovarajući okvir uzorkovanja. Kao primer Bajber (1993) navodi konstrukciju korpusa kojim bi se predstavili govorni tekstovi u jeziku „pošto ne postoje nikakvi katalozi ni bibliografije govornih tekstova, i pošto svi mi proširujemo univerzum govornih tekstova svakodnevnim razgovorima...". Naravno, to ne znači da se ne može izabrati uzorak govornih tekstova, ali je praktično nemoguće oceniti reprezentativnost takvog uzorka.

Reprezentativnost uzorka

Na pitanje (R2) se obično daje odgovor „pozajmljen" iz statistike, tj. uzorak se smatra reprezentativnim ako rezultati dobijeni na uzorku važe i na celoj populaciji ([Manning & Schütze, 1999: 119]). Međutim, pokazuje se da je daleko teži problem kako u praksi doći do tako definisanog reprezentativnog uzorka i to objektivno dokazati, usled čega se javlja i sumnja kako u reprezentativnost pojedinih korpusa, tako i u mogućnost da se određenom metodom može kreirati reprezentativni korpus. Tako pojedini autori otvaraju pitanje reprezentativnosti i elektronskih korpusa prve generacije, konkretnije, Braunovog korpusa i korpusa LOB:

> „[O jedinicama uzorka koji će predstavljati populaciju Braunovog korpusa je] ... odlučeno na sastanku stručnjaka koji su dizajnirali shemu po kojoj su različiti jezički varijeteti, nazvani *žanrovi*, zastupljeni u specifičnim srazmerama[...] mnogo napora je potrošeno da bi se osigurao slučajan izbor tekstova unutar svake [tekstuelne] kategorije, ali nije mi poznat nijedan javni argument kao opravdanje za pojedine srazmere između kategorija" ([Váradi, 2001: 589]).

Bajber (1993) smatra da se „reprezentativnost odnosi na obim u kojem uzorak obuhvata celokupan opseg raznovrsnosti unutar populacije". S tim u vezi on je izvršio niz studija za engleski jezik koje prikazuju mogućnost izuzetno finog strukturiranja korpusa prema funkcionalnim stilovima i žanrovima ([Biber, 1988; Biber et al., 1998; Biber & Finegan, 1991]). Time se problem reprezentativnosti svodi na dva pitanja koja će biti posebno razmotrena:

(A) opseg žanrova obuhvaćenih korpusom i njihova međusobna srazmernost, tj. **balansiranost** (eng. **balance**) korpusa.

(B) uzorkovanje u okviru pojedinačnih žanrova.

S obzirom da odgovori na oba pitanja zavise od toga šta se tačno podrazumeva pod *žanrom*, neophodno je najpre precizirati taj pojam.

Preciziranje pojmova *stil*, *funkcionalni stil*, *registar*, *žanr*

Različiti autori koji se bave teorijskim i praktičnim istraživanjima zasnovanim na korpusu, a pogotovo temom reprezentativnosti i balansiranosti korpusa, na različite načine koriste pojmove *stil* (eng. *style*), *funkcionalni stil* (eng. *functional style*), *registar* (eng. *register*), *žanr* (eng. *genre*), *tip teksta* (eng. *text type*): ponekad se isti par pojmova kod jedne grupe autora tretira kao par sinonima, dok druga grupa autora između njih uspostavlja odnos hiperonim/hiponim, pri čemu je isti pojam kod jednog autora nadređeni, a kod drugog podređeni. Takva neusklađenost terminologije ne samo da stvara konfuziju, već i dodatno otežava modeliranje korpusa, kao i ocenjivanje njegove reprezentativnosti. Stoga je neophodno da se navedeni pojmovi jasno razgraniče pre njihovog daljeg korišćenja.

Ekspertske savetodavne grupe za standarde jezičkog inženjerstva (eng. **Expert Advisory Groups on Language Engineering Standards**, skr. **EAGLES**) (v. str. 135) su pokušale da doprinesu standardizaciji terminologije tako što su izdale preporuku o tipologiji tekstova ([Sinclair & Ball, 1996]). U svojoj preporuci EAGLES koriste iste kriterijume kao i Bajber ([Biber, 1993]) i razlikuju unutrašnje (interne) i spoljašnje (eksterne) kriterijume klasifikacije korpusnih tekstova. Interni kriterijumi se odnose na lingvističke karakteristike teksta (leksičke, gramatičke), dok se spoljašnji kriterijumi oslanjaju na konvencije grupisanja tekstova (sociološke, kulturne, institucionalne, itd.) koje nisu lingvističke, već su vezane za upotrebu teksta, a ne njegovu formu i strukturu.

Opisujući pokušaj klasifikacije 4.124 teksta za BNC, Dejvid Li (David Lee) detaljno razmatra kako se različiti autori odnose prema navedenim pojmovima ([Lee, 2001]). U svom zaključku on razdvaja pojmove *stil*, *registar* i *žanr* na sledeći način:

- „... *stil* se u suštini odnosi na individualnu upotrebu jezika", tj. karakteriše pojedinca koji na određeni način upotrebljava jezik, pri čemu se razlikuju „formalni i neformalni stil u kombinaciji sa parametrima kao što su način pripreme (pripremljen/spontan), smer (jednosmerna ili interaktivna komunikacija), 'komunikacijsko grupisanje' (grupa u kojoj se vodi konverzacija; govornik/pisac i publika; posredna publika)" ([Lee, 2001]45).

- registar i žanr predstavljaju „dva različita pogleda na istu stvar. Registar se koristi kada na tekst gledamo kao na jezik (...) Žanr se koristi kada se na tekst gleda kao na član neke kategorije...". Preciznije, registri razlikuju upotrebljene lingvističke obrasce (u smislu rečnika, gramatike, značenja, diskursa, itd.) u određenim situacijama. U različitim situacijama biće upotrebljeni oni lingvistički obrasci koji su datoj situaciji „maksimalno 'funkcionalno prilagođeni"'. S druge strane, žanrovi su „kategorije uspostavljene kulturnim konsenzusom i time [...] vremenom [...] podložne promenama" ([Lee, 2001]46).

Li se na kraju odlučuje za klasifikaciju korpusnih tekstova prema žanrovima.

Pojedini domaći autori tretiraju *funkcionalni stil* i *registar* kao sinonime (npr. [Bugarski, 1995]). Pri tom najčešće razlikuju nekoliko vrsta funkcionalnih stilova koji nisu strogo i precizno razgraničeni: književnoumetnički, publicistički, naučni, administrativni, razgovorni ([Tošović, 2002], [Simić & Jovanović, 2002], [Klikovac, 2008: 111–134]).

S druge strane, Bajber se u svojim analizama radije bavi raspodelom *registara* umesto raspodelom *žanrova*, pri čemu te pojmove tretira kao „različita viđenja tekstuelnih varijeteta": u oba slučaja je „bitan opis namene i situacioni kontekst", ali se

- „u slučaju registra kombinuje analiza lingvističkih karakteristika koje su uobičajene u okviru tekstuelnog varijeteta sa analizom situacije u kojoj se varijetet upotrebljava..." ([Biber & Conrad, 2009: 2]),

- dok se, kada je žanr u pitanju, „... lingvistička analiza fokusira na konvencionalne strukture koje se koriste pri konstrukciji kompletnog teksta u okviru

varijeteta, npr. konvencionalan način da se započne i završi pismo." ([Biber & Conrad, 2009: 2])

Balansiranost

Kada se obrazlaže reprezentativnost najistaknutijih predstavnika elektronskih korpusa I i II generacije, poput Braunovog i Britanskog nacionalnog korpusa, najčešće se ističe da su oni balansirani. Prema Liču, korpus je **balansiran** (eng. **balanced corpus**) „kada je veličina njegovih potkorpusa (koji predstavljaju pojedinačne žanrove ili registre) proporcionalna relativnoj učestanosti pojavljivanja tih žanrova u celokupnom jezičkom tekstuelnom univerzumu", tj. „balansiranost je jednako proporcionalnost" ([Leech, 2007: str. 136]). Na istom mestu se primećuje da prilikom kreiranja Braunovog korpusa i korpusa BNC „nije bilo ozbiljnih pokušaja da se postigne balansiranost u tom smislu".

Osnovni problem sa pojmom balansiranosti je nemogućnost da se objektivno i precizno odrede relativne srazmere različitih registara i žanrova pre no što se pristupi uzorkovanju tekstova za korpus. Bajber primećuje da postoje tri elementa jezika koji mogu da se uzmu u obzir prilikom uzorkovanja:

(i) govornici i pisci —- inicijatori teksta;

(ii) slušaoci i čitaoci —- primaoci teksta, i

(iii) sami tekstovi.

Međutim, u svojoj kritici proporcionalnosti (a time i balansiranosti) u okviru korpusa ([Biber, 1993: str. 247–248]), Bajber se prvenstveno bavi samo inicijatorima teksta. Po njemu, svaki „proporcionalan" korpus mora biti zasnovan na demografskim kriterijumima, što za posledicu ima da se „oko 90% svih tekstova"[5] svodi samo na konverzacije, dok između svih ostalih registara treba raspodeliti preostalih 10%. Samim tim, „proporcionalan" korpus ne znači i reprezentativan u odnosu na uticaj i važnost pojedinih registara ili žanrova, jer pojedini registri i žanrovi, poput knjiga

[5]koji su uzorkovani

i vesti koje plasiraju mediji, imaju daleko veći uticaj nego što to sugerišu njihove relativne učestanosti u „proporcionalnom" korpusu.

Kao odgovor na Bajberovu kritiku, Lič predlaže da osnovna jedinica prilikom određivanja veličine datog tekstuelnog univerzuma ne bude sam tekst, već veza inicijator-tekst-primalac koju naziva **atomični komunikacijski događaj** (eng. **Atomic Communication Event**, skr. **ACE**) ([Leech, 2007: str. 138]). Kao primer za ACE, Lič navodi radio program koji slušaju milioni ljudi; iako taj radio program predstavlja samo jedan tekst, u pitanju je milion različitih ACE; zato bi takav tekst trebalo uključiti u korpus pre nego razgovor dva čoveka tokom kojeg u svakom trenutku postoji samo jedan slušalac. Lič dalje navodi da postoje slučajevi sa višestrukim inicijatorima, „npr. u horskom govoru ili koautorstvu pisanih tekstova", ali „ogromna većina tekstova" ima samo jednog inicijatora. Stoga predlaže da je dovoljno ispitati proporcionalnost prijema teksta, tj. koliko vremena je provedeno u slušanju različitih kategorija govora i čitanju različitih kategorija pisanog teksta, kako bi se modelirala „ACE-proporcionalnost", tj. balansiranost u smislu Ličove definicije. Ispitivanje proporcionalnosti prijema teksta bi moglo da se obavi primenom sociooloških metoda (korišćenje demografskog uzorka, ispitivanje pomoću dnevnika, upitnika, itd.). Kao ilustraciju, Lič navodi da je tokom modeliranja Češkog nacionalnog korpusa korišćeno istraživanje prijema jezika da bi se odredile proporcije različitih žanrova pisanog teksta ([Čermák & Schmiedtová, 2003: str. 212]).

Veličina uzorka (korpusa)

U pogledu pitanja (R3), jedino oko čega postoji slaganje jeste da što je korpus veći, to je i reprezentativniji. Oko svega ostalog, mišljenja su podeljena, pri čemu se jasno razlikuju dve suprotstavljene grupe korpusnih lingvista. Prva grupa (Kverk, Lič) zastupa da korpus treba kreirati tako da bude balansiran i više ne menjati njegovu veličinu (v. odeljak 1.3, statički korpus, str. 25). Druga grupa čiji je rodonačelnik Sinkler, smatra da korpus mora redovno da se uvećava, dodavanjem novih materijala (v. odeljak 1.3, dinamički korpus, str. 25), tj. „mora redovno da se ažurira, inače brzo postaje nereprezentativan" ([Hunston, 2002: 30]).

U prvoj deceniji XXI veka postoji tendencija da veličina korpusa predstavlja

njegov glavni parametar, pri čemu se često prećutno i neosnovano pretpostavlja da će korpus postati reprezentativan „sam od sebe" kada dostigne pozamašnu veličinu ([Xiao, 2010: 151]).

Metodi izbora uzorka

Kada se definiše populacija, odnosno okvir uzorkovanja, ostaje još da se utvrdi način na koji će biti birani jedinični uzorci, tako da reprezentativnost uzorka bude što veća. U statistici postoji više različitih metoda za odabiranje uzorka: prost slučajan uzorak, stratifikovan slučajan uzorak, sistematski uzorak, grupni uzorak, višeetapni uzorak, dvofazni uzorak ([Petrović, 2007]).

Najjednostavniji je metod **prostog slučajnog uzorka** (eng. **simple random sampling**) koji se zasniva na pretpostavci da svaki element populacije ima podjednaku verovatnoću da bude uključen u uzorak, pri čemu su izbori pojedinačnih elemenata međusobno nezavisni. U praksi se izbor konkretnog jediničnog uzorka iz okvira uzorka svodi na slučajno određivanje rednog broja pod kojim se jedinica uzorka nalazi na spisku (popisu) članova populacije. Slučajnost izbora se obezbeđuje primenom raspoloživih tablica slučajnih brojeva.

Međutim, ako struktura populacije nije homogena, npr. ako se jedinice uzorka značajno razlikuju po veličini (kao što je slučaj pri izboru celovitih tekstova za korpus) ili po učestalosti podvrste u populaciji kojoj pripadaju (kao što je slučaj sa novinskim i administrativnim tekstovima), prost slučajan uzorak nije verodostojan metod jer se verovatnoće izbora jediničnih uzoraka previše razlikuju. U takvim slučajevima se najčešće primenjuje **stratifikovan slučajan uzorak** (eng. **stratified sampling**). Ideja ovog metoda je da se najpre populacija podeli na **stratume** (lat. **stratum, strata, n.**, *grupa, klasa*), a onda se za svaki stratum pravi poseban okvir uzorka, tj. za svaki stratum se posebno vrši uzorkovanje. Prilikom uzorkovanja stratuma može se iskoristiti bilo metoda prostog slučajnog uzorka, bilo ponoviti metoda stratifikovanog uzorka.

Da li je reprezentativnost ostvariv cilj

Lič ([Leech, 2007]) opisuje jednu od najranijih diskusija o 'reprezentativnosti korpusa' vođenu u zborniku [Bergenholtz & Schaeder, 1979]. Upravo u tom zborniku je Fransis, jedan od tvoraca Braunovog korpusa, u svom članku ([Francis, 1979: 110]) definisao korpus kao „kolekciju tekstova za koju se *pretpostavlja* da je reprezentativna za dati jezik, dijalekt ili drugi podskup jezika, a koja se koristi za lingvističku analizu [dodat kurziv]". Dva autora u istom zborniku daju svoje negativno mišljenje:

- Riger ističe da se reprezentativnost korpusa kao uzorka može postići tek „kada budemo znali toliko o univerzumu iz kog (uzorak) dolazi da nam formiranje tog uzorka više neće biti potrebno" ([Rieger, 1979: 66])

- Bungarten smatra da ako već ne može da se ostvari reprezentativni korpus, postoji niži cilj kome treba težiti - „egzemplarni korpus". Po njemu, „egzemplarni korpus" je korpus čija se reprezentativnost ne može dokazati, ali se može tvrditi na osnovu „manje formalnih argumenata" poput „konsenzusa stručnjaka" ([Bungarten, 1979: 42–43]).

Intenzivne rasprave o reprezentativnosti korpusa su vođene tokom cele poslednje decenije XX veka. Početkom XXI veka, pre svega kao posledica uvođenja koncepta „veba kao korpusa" (v. odeljak 1.4, str. 31), je došlo do promene percepcije korpusa kao nužno reprezentativnog u odnosu na celokupan jezik, već je za zastupnike pomenutog koncepta dovoljno da bude reprezentativan u odnosu na onaj deo jezika za čije je istraživanje namenjen. Korpusne istraživače sve više interesuju informacije koje mogu da ekstrahuju iz korpusa, kao i pronalaženje novih oblasti u računarskoj lingvistici i obradi prirodnog jezika u kojima te informacije mogu da se iskoriste, a manje teorijski aspekti korpusne lingvistike. Stoga pitanje reprezentativnosti korpusa ostaje nerešeno i donekle potisnuto, ali uvek spremno da se nađe na dnevnom redu. Jer, kako primećuje Varadi, analizom korpusa, ne vodeći računa o njegovoj reprezentativnosti, se dobijaju rezultati koji važe samo za taj korpus i ne mogu se uopštiti ni na šta drugo ([Váradi, 2001]).

1.3 Klasifikacija korpusa

Korpusi se mogu klasifikovati na osnovu svojih parametara kao što su nosač, obim (veličina), domen, namena, period, izvor, način anotacije, broj uključenih jezika, itd. ([Vitas & Popović, 2003]).

Nosač

U zavisnosti od nosača, korpusi mogu biti preelektronski i elektronski (v. odeljak 1.1). Preelektronski korpusi su uglavnom bili papirni, a i sada postoje papirni ekvivalenti pojedinih elektronskih korpusa. Međutim, pojavom elektronskih računara, najpre njihova brzina obrade, a kasnije i lakoća sa kojom se manipuliše podacima (skladištenje, pretraživanje, izbor i ekstrakcija, sortiranje, formatiranje) doprineli su apsolutnoj dominaciji elektronskih korpusa, tako da se danas pod pojmom *korpus* uglavnom podrazumeva *elektronski korpus*. Dodatne prednosti upotrebe računara pri razvoju, održavanju i korišćenju korpusa jesu preciznost i konzistentnost automatske obrade podataka lišene ljudskih predrasuda[6], kao i mogućnost dopunjavanja tekstova metapodacima korisnim za analizu korpusa (v. podnaslov Anotacija ovog odeljka, str. 27).

Domen i namena

Prema domenu korpusi se mogu podeliti na **opšte** (eng. **general corpora, core corpora**) i **specijalizovane** (eng. **specialized corpora**).

Opšti korpusi predstavljaju osnovu za proizvoljno lingvističko istraživanje, te stoga mogu imati raznovrsne namene, tj. mogu se koristiti za leksikografska, gramatička, semantička, pragmatička, sociolingvistička, psiholingvistička i dr. istraživanja. Stoga se prilikom konstrukcije statičkog opšteg korpusa insistira na njegovoj balansiranosti, dok dinamički opšti korpus povećavanjem svoje veličine pokušava da postigne dovoljnu reprezentativnost.

[6]Izuzev onih koje je programer ugradio kao pravila u softver za obradu podataka.

Specijalizovani korpusi nastaju radi nekog specifičnog lingvističkog istraživanja (**leksikografski korpusi**, **gramatički korpusi**, **dijalekatski korpusi**, **regionalni korpusi**, **nestandardni korpusi**, **korpusi jezika kao nematernjeg**, itd.) ili kao pomoćno sredstvo u računarskoj lingvistici/obradi prirodnog jezika (korpusi za treniranje alata za automatsku morfosintaksičku analizu teksta, automatsko prepoznavanje i generisanje govora, itd.). Nestandardni korpusi se obično kreiraju kao kolekcije tekstova od izuzetnog kulturnog značaja ili statusa, tekstova koji „nisu pouzdani primeri savremenog [. . . jezika]; ne biraju se zbog svoje običnosti već zbog svoje izuzetnosti" ([Tognini-Bonelli & Sinclair, 2005: 210]).

Obim (veličina)

U zavisnosti od toga da li po kreiranju korpusa njegova veličina ostaje fiksirana ili se korpus neprekidno dopunjuje novim tekstovima, razlikujemo **statičke** (eng. **static corpora**) i **dinamičke** (eng. **dynamic corpora, monitor corpora**) korpuse. Tvorcem i jednim od vodećih zastupnika ideje o dinamičkom korpusu obično se smatra Džon Sinkler koji je prvi upotrebio izraz *monitor corpora* ([Sinclair, 1991: 24–26]). Upravo iz njegovog rada na projektu COBUILD (v. odeljak 1.1, str. 12) je 1991. godine ([Xiao, 2008: 394–395]) nastao svakako najpoznatiji dinamički korpus, **Banka engleskog** (eng. **Bank of English**, skr. **BoE**)[7]. Tokom dvadesetak godina postojanja ovaj korpus je neprekidnim dopunjavanjem dostigao 650 miliona reči (počev od 2012. godine).

Posebnu vrstu dinamičkih korpusa predstavljaju **oportunistički korpusi** (eng. **opportunistic corpora**). Takvi korpusi se konstantno dopunjavaju svim tekstovima do kojih kreatori korpusa mogu da dođu, a da pri tom dodati tekstovi zadovoljavaju osnovnu namenu korpusa.

Period

Na osnovu vremenskog perioda koji predstavlja, korpus se može svrstati u **sinhrone** (eng. **synchronic corpora**) ili **dijahrone korpuse** (eng. **diachronic cor-**

[7] http://www.collins.co.uk/

pora). Sinhronim korpusom se predstavlja jedan specifični vremenski period koji se može posmatrati kao samostalna celina (npr. realizam u srpskoj književnosti od 1870. do 1914. godine). Dijahroni korpus, po pravilu, ima za cilj da obuhvati duže vremenske periode (nekoliko vekova ili duže) tako da se na osnovu njega mogu izučavati sporije jezičke promene koje se ne mogu detektovati u sinhronom korpusu. Često se dijahroni korpusi poistovećuju sa **istorijskim korpusima** (eng. **history corpora**). U praksi se u istu svrhu mogu koristiti i sinhroni korpusi kojima su predstavljeni različiti vremenski periodi, ali su, u pogledu ostalih kriterijuma, modelirani na sličan način ([Lindquist, 2009: 167]), što je npr. slučaj sa Braunovim korpusom i korpusom LOB (tekstovi iz 1961. godine) i njihovim Frajburškim verzijama (FROWN i FLOB) sa tekstovima iz 1991. godine (v. odeljak 1.1, str. 11).

Izvor/medijum

Prema izvoru (medijumu) tekstova koji su obuhvaćeni korpusom mogu se razlikovati korpusi **pisanih** (eng. **written corpora**), **govornih** (eng. **spoken corpora**), **elektronskih** tekstova (npr. elektronska pošta, blogovi, forumi, društvene mreže), dok korpusi koji pretenduju da budu opšti kombinuju sve navedeno. Korpusi govornih tekstova su daleko manje zastupljeni od korpusa pisanih tektova, pre svega zbog toga što njihovo kreiranje zahteva i vreme i novčana sredstva kako bi se govor transkribovao (fonetski/fonološki i prozodijski) i predstavio kao mašinski čitljiv tekst (v. odeljak 2.1).

Posebno se izdvajaju **multimodalni korpusi** (eng. **multimodal corpora**). Mulitimodalni korpus u širem smislu može da se definiše kao „digitalizovana kolekcija materijala koji se odnosi na jezik i komunikaciju i koji se mogu iskoristiti pomoću više čulnih modaliteta" ([Allwood, 2008: 208]); u užem smislu to je kolekcija audiovizuelnog materijala (tekstuelni, zvučni i video zapisi) na kojem je istovremeno zabeleženo više čulnih modaliteta međuljudske komunikacije (govor, izrazi lica, pokreti usana, očiju i glave, gestikulacija, dodir), pri čemu kolekcija obično sadrži i transkribovanu i anotiranu verziju te komunikacije ([Allwood, 2008; Wittenburg, 2008; Xiao, 2010: 161]). Anotacija omogućava uspostavljanje veze između različitih modaliteta koji su korišćeni u određenom vremenskom trenutku, a obuhvata i infor-

macije o prostornom i vremenskom kontekstu snimljene komunikacije.

Ponekad se značenja pojmova *multimodalni* i *multimedijalni* preklapaju, te se zbog njihovog preciziranja ističe da je „'komunikacioni medijum' fizički nosilac multimodalne informacije[...tj.] medijum vida su talasi svetlosti, medijum sluha su zvučni talasi, medijum dodira je fizički pritisak, a medijum mirisa i ukusa su razni tipovi hemijskih molekula" ([Allwood, 2008: 208]).

Svojstvo multimodalnosti elektronskih dokumenata zahteva složeniju pretragu u odnosu na postojeću „jednomodalnu" ([Marković, 2009]), uglavnom tekstuelnu pretragu. Složenost tehnologije neophodne za izradu i pretragu multimodalnih korpusa glavni je razlog zašto su ovi korpusi tek u začetku u smislu zastupljenosti i razvijenosti, što će se sigurno izmeniti u bližoj budućnosti.

Anotacija

Anotacija korpusa je postupak kojim se dodatne informacije pridružuju delovima korpusa (tekstovi, logičke celine u okviru teksta, tokeni). U zavisnosti od vrste informacija, kao i od delova korpusa kojima se pridružuju, razlikujemo nekoliko nivoa anotacije ([Utvić, 2011]):

(1) Tekstu korpusa se mogu pridružiti odgovarajuća bibliografska referenca, podaci o kreiranju i ažuriranju elektronske verzije teksta, kao i statistički podaci o tekstu.

Bibliografska referenca se svodi na informacije o izvoru teksta poput naslova, imena i prezimena autora, godine izdanja, informacije o izdavaču, itd.

Podaci o kreiranju i ažuriranju elektronske verzije teksta se odnose na datum njenog nastanka, originalni oblik teksta (papirno izdanje, mašinski čitljiv tekst), eventualnu transformaciju teksta iz originalnog u mašinski čitljiv oblik (skaniranje i optičko prepoznavanje karaktera, prekucavanje), osobe odgovorne za kreiranje, korekciju, distribuiranje elektronske verzije, itd.

Pridruženi statistički podaci o tekstu su najčešće dužina teksta, izražena brojem tokena i tipova, odnosno brojem korpusnih reči i korpusnih tipova, mada se, u zavisnosti od namene korpusa, mogu pridružiti i drugi numerički podaci

kojima se opisuju raspodele određenih lingvističkih elemenata poput vrsta reči, koncepata značenja, itd.

(2) Ako se u okviru teksta uoči i obeleži njegova logička struktura (poglavlja, naslovi, pasusi, rečenice), realizuje se **strukturna anotacija** (eng. **structural annotation/markup**);

(3) Svakoj korpusnoj reči se može pridružiti informacija o:

(i) vrsti reči (imenica, pridev, glagol, itd.),

(ii) lemi (nominativ jednine imenice, infinitiv glagola, itd.),

(iii) vrednostima flektivnih kategorija (rod, broj, padež, glagolski oblik, glagolski vid, itd.), odnosno flektivnoj osnovi i nastavcima;

(iv) (tvorbenoj) osnovi, prefiksima, infiksima i sufiksima;

(v) načinu izgovora (akcenat);

(vi) granicama slogova;

Podtipovi anotacije (3i),(3ii),(3iii) imaju posebne nazive i to **etiketiranje vrstom reči** (eng. **Part of Speech tagging**, skr. **PoS tagging**), **lematizacija** (eng. **lemmatization**), **gramatička anotacija** (eng. **grammatical annotation**), tim redom.

(4) Ako se svakom tokenu pridruži oznaka odgovarajućeg značenja, u pitanju je **semantička anotacija** (eng. **semantic annotation**);

(5) Jednočlanom ili višečlanom nizu korpusnih reči može se pridružiti zajednička informacija o funkciji u rečenici (subjekat, predikat, objekat, glagolska odredba) ili se takav niz može obeležiti kao sintagma (imenička, pridevska, itd.). Takvo pridruživanje informacija se naziva **sintaksička anotacija** (eng. **parsing**).

(6) Na nivou diskursa se može realizovati **anotacija koreferencije** (eng. **coreference annotation**) kojom se u tekstu označavaju relacije koreferencije (anaforički i kataforički odnosi) između korpusnih reči;

(7) Obeležavanjem govornog čina u tekstu pragmatičkim, odnosno stilističkim informacijama, realizuju se **pragmatička anotacija** (eng. **pragmatic annotation**) i **stilistička anotacija** (eng. **stylistic annotation**) tim redom.

Iako većina autora za sva navedene oblike pridruživanja informacija korpusu koristi pojam **anotacija korpusa** (eng. **corpus annotation**), pojedini autori ([Xiao, 2010]) pod tim pojmom podrazumevaju samo pridruživanje lingvističkih informacija (lema, vrsta reči, itd.), dok se za ostale (nelingvističke) informacije (bibliografski podaci o tekstu, originalnom formatiranju teksta, itd.) koristi termin **označavanje korpusa** (eng. **corpus markup**).

Prednosti anotacije korpusa su:

- Pretraga korpusa se efikasnije realizuje zahvaljujući mogućnosti preciznijeg zadavanja upita.
- Anotacija korpusa u rezultatima pretrage nadoknađuje informacije koje su izgubljene tokom pripreme korpusa (eliminisane slike, tabele, fusnote, i sl.), kao i informacija koje nedostaju zbog nedovoljno širokog konteksta u kome su prikazani rezultati pretrage.
- Anotacija korpusa olakšava statističku analizu korpusa, tj. automatsko određivanje raspodele anotiranih lingvističkih svojstava.

Anotaciju korpusa karakterišu i nedostaci, zbog kojih postoji otpor prema samoj ideji anotacije. Stav prema anotaciji je jedna od suštinskih razlika između lingvista koji zastupaju **pristup zasnovan na korpusu** (eng. **corpus-based linguists**) i lingvista koji zastupaju **pristup vođen korpusom** (eng. **corpus-driven linguists**). Pristalice prvog pristupa se zalažu za što detaljniju anotaciju korpusa jer smatraju da je uloga korpusa testiranje, korigovanje i dopunjavanje postojećih teorija, pronalaženjem primera koji ih potvrđuju ili opovrgavaju. Zastupnici pristupa vođenog korpusom zastupaju stav da korpus treba analizirati bez unapred oformljenih teorija kako bi se isključivo na osnovu podataka dobijenih analizom postulirale lingvističke kategorije. Potonji pristup stoga tretira anotaciju korpusa kao nepotrebnu, jer njome anotatori zapravo izvode jednu konkretnu analizu korpusa, pa se

naknadnim analiziranjem anotiranog korpusa mogu samo ponoviti rezultati njihove prethodne analize korpusa ([Lindquist, 2009: 45]).

Višejezičnost

U okviru istog korpusa mogu se naći tekstovi na jednom ili više jezika, pa se tako mogu razlikovati **jednojezični** (eng. **monolingual corpora**) i **višejezični korpusi** (eng. **multilingual corpora**). Najčešći broj jezika kod višejezičnih korpusa je dva, pa se često u literaturi izjednačavaju sa **dvojezičnim korpusima** (eng. **bilingual corpora**)[8].

Daljom klasifikacijom višejezičnih korpusa razlikuju se **paralelizovani** (eng. **parallel corpora, aligned corpora, translation corpora**) i **uporedni** (eng. **comparable corpora**). Iako je bilo nedoslednosti i uzajamne razmene definicija ova dva pojma ([Xiao, 2010: 159]), danas je uglavnom preovladalo stanovište da su paralelizovani korpusi sastavljeni od tekstova na izvornom jeziku i njihovih prevoda na jedan ili više ciljnih jezika, dok su uporedni korpusi[9] u opštem slučaju sastavljeni od različitih tekstova na različitim jezicima, ali odabranih iz istog okvira uzorkovanja i po sličnim kriterijumima poput „tip teksta, formalnost, tema, vremenski raspon, itd." ([Aijmer, 2008: 276]).

U praksi se paralelizovani korpusi sa više od dva jezika najčešće realizuju kao skup dvojezičnih paralelizovanih korpusa. Osnovni element dvojezičnog paralelizovanog korpusa (kao kolekcije) nije tekst već **bitekst** (eng. **bitext**) ili **paralelizovani tekst** (eng. **parallel text, aligned text**) Pod paralelizovanim tekstom ili bitekstom se „obično podrazumevaju tekst i njegov(i) prevod(i) predstavljeni na takav način da je između elemenata njihovog logičkog izgleda uspostavljena eksplicitna veza" ([Vitas, 2010]). Veza se uspostavlja između jedinica pojedinih strukturnih nivoa teksta (tj. nivoa dokumenta, poglavlja, pasusa, rečenice ili reči) i ona omogućava analizu

[8]Ponekad se posebno izdvajaju *dvojezični korpusi*, a pojam *višejezični korpus* se koristi kada su tekstovi na više od dva jezika.

[9]Tonjini-Boneli i Sinkler daju uopšteniju definiciju uporednih korpusa, nezavisno od višejezičnih korpusa [Tognini-Bonelli & Sinclair, 2005: 209–210]: „dva ili više korpusa su uporedni ako su napravljeni na osnovu istih planskih kriterijuma i slične su veličine". Kao podvrste takvih korpusa navode se **geografski** (eng. **geographical corpora**), **istorijski** (eng. **historical corpora**), **tematski** (eng. **topic corpora**) i **kontrastni korpusi** (eng. **contrastive corpora**).

paralelizovanog korpusa.

Razlikuju se sledeće vrste paralelizovanih korpusa ([Xiao, 2010: 160]):

1. jednosmerni (tj. svi izvorni tekstovi korpusa su na jednom jeziku, a svi prevodi na drugom),

2. dvosmerni (u korpusu postoje i izvorni tekstovi i prevodi na dva jezika) i

3. višesmerni paralelizovani korpusi (ista verzija teksta na više od dva jezika).

Poseban slučaj predstavljaju paralelizovani korpusi čiji su svi tekstovi na istom jeziku jer su u pitanju različiti prevodi istog teksta čiji izvorni jezik nije prisutan u korpusu. Po svom sadržaju (tj. broju jezika) reklo bi se da nisu višejezični, odnosno paralelizovani korpusi, ali postojanje eksplicitne veze na strukturnom nivou između prevoda istog teksta ukazuje na skoro sve osobine biteksta (sem prisustva originalnog teksta na izvornom jeziku), pa se taj terminološki problem toleriše. Ovakvi „jednojezični paralelizovani korpusi" su posebno pogodni za izučavanje teorije prevođenja.

Paralelizovani korpusi mogu imati primenu u nastavi stranog jezika, kontrastnoj lingvistici, pri kreiranju terminoloških baza podataka[10], *prevodilačkih memorija* (eng. *translation memories*)[11], višejezičnih rečnika, kao i u podoblastima računarske lingvistike kojima je blizak problem mašinskog (automatskog) prevođenja.

1.4 Veb i korpus

U periodu od marta 1989. godine do decembra 1990. godine je razvijen **veb** (eng. **World Wide Web**, skr. **WWW, W3, web**), servis Interneta zamišljen kao mreža hipertekstuelnih dokumenata, međusobno povezanih naročitim referencama — **hipervezama** (eng. **hyperlink**)[12]. Mogućnost da sami korisnici na jednostavan način

[10]Kreiranje terminoloških baza podataka je neophodna aktivnost svih međunarodnih organizacija poput Ujedinjenih Nacija ili Evropske Unije.

[11]„Prevodilačka memorija predstavlja bazu podataka u kojoj se čuvaju delovi prevedenih tekstova (rečenice, sintagme ili fraze) kako bi mogli ponovo da se iskoriste kada je potreban sličan prevod" ([Sofer, 2006: 86]).

[12]Iako naziv *hipertekstuelni* sugeriše da se radi isključivo o tekstu sa hipervezama, hipertekstuelni dokument može sadržati, pored teksta, i slike, zvučne i video datoteke, kao i njihove kombinacije — multimedije.

proširuju mrežu dodavanjem sopstvenih hipertekstuelnih dokumenata dovela je do stvaranja ogromne kolekcije sadržaja procenjene na trilion (10^{12}) stranica ([Farhan & D'Agostino, 2012: 9]).

Kao najveća postojeća kolekcija elektronskog teksta, veb je još od kraja devedesetih godina XX veka, pa sve do danas zanimljiv za korpusne lingviste na nekoliko načina ([Bernardini et al., 2006: 10–15]):

- *veb kao surogat korpus* (eng. *Web as corpus surrogate*),
- *veb kao prodavnica korpusa* (eng. *Web as corpus shop*),
- *veb kao korpus za sebe* (eng. *Web as corpus proper*),
- *veb kao mega-korpus/mini-veb* (eng. *Web as mega-corpus/mini-Web*).

Kao što sam naziv *veb kao surogat korpus* sugeriše, pojedini korpusni lingvisti koriste veb kao zamenu za korpus ([Bernardini et al., 2006: 10-11]), pri čemu se pretraga i analiza rezultata pretrage svodi na mogućnosti koje im pružaju pretraživači (Google[13], Yahoo[14], Bing[15], itd.) i lingvistički-orijentisani metapretraživači (WebCorp[16], WebAsCorpus.org[17], WebCONC[18], GlossaNet[19]).

Korisnici *veba kao prodavnice korpusa*, u okviru procesa koji se može donekle automatizovati, na osnovu rezultata upita prosleđenih pretraživačima biraju i preuzimaju tekstove sa veba, a onda na osnovu preuzetih tekstova kreiraju svoje elektronske korpuse po uzoru na tipične predstavnike korpusa II generacije ([Bernardini et al., 2006: 11-12]). S obzirom na način na koji se izgrađuje, korpus ovog tipa se obično naziva **korpus izveden iz veba** (eng. **web-derived corpus**).

Pristup *veb kao korpus za sebe*, za razliku od prethodna dva pristupa, ima za cilj istraživanje samog veba. Pretraga veba se tretira kao uzorkovanje celokupnog veba, tj. koristi se metodologija slična onoj koju korpusna lingvistika primenjuje

[13] `http://www.google.com`
[14] `http://www.yahoo.com`
[15] `http://www.bing.com`
[16] `http://www.webcorp.org.uk/live`
[17] `http://webascorpus.org`
[18] `http://gandalf.uib.no/lingkurs/templates_c/%25%256E%5E6ED%5E6EDF58DD%25%25labor.html.php`
[19] `http://glossa.fltr.ucl.ac.be`

kada donosi zaključke o celokupnom jeziku na osnovu analize korpusa kao uzorka jezika ([Bernardini et al., 2006: 13]).

Koncept *veb kao mega-korpus/mini-veb* u sebi ujedinjuje prethodne pristupe sa ciljem da stvori mega-korpuse ili **veb-korpuse** (eng. **web corpora**). Naziv mega-korpusi se odnosi na njihovu veličinu, s obzirom da su (najmanje) za jedan red veličine veći od postojećih elektronskih korpusa II generacije. S druge strane, mega-korpusi su dostupni preko sličnog sučelja koje koriste pretraživačke mašine za pretragu veba, te se zbog svoje veličine i porekla označavaju kao mini(jaturni)-veb, odnosno kao veb-korpusi. Postojeći veb korpusi su uglavnom i lingvistički anotirani i mogu se istovremeno koristiti i za istraživanje jezika uopšte, kao i za istraživanje samog veba ([Bernardini et al., 2006: 13–14]).

S obzirom da se u praksi navedeni koncepti preklapaju, često se svi nazivaju istim imenom — *veb kao korpus*. U skladu sa ([Bergh & Zanchetta, 2008: 315]), u ovom radu će se ipak razlikovati:

- **veb za korpus** (eng. **Web for Corpus**, skr. **WfC**), odnosno veb kao najbogatiji izvor tekstova za kreiranje elektronskih korpusa II generacije i
- **veb kao korpus** (eng. **Web as Corpus**, skr. **WaC**), čime je obuhvaćeno i tretiranje celog veba kao „najvećeg postojećeg elektronskog korpusa" i korišćenje veba za generisanje veb-korpusa.

O korišćenju veba kao izvora tekstova za kreiranje elektronskih korpusa II generacije biće posebno reči u poglavljima 2.1 i 4.2. Alati za generisanje i pretragu veba kao korpusa opisani su u odeljku 4.2.

U nastavku odeljka razmatra se uticaj veba na preispitivanje definicije korpusa, kao i pojava veb-korpusa kao nove klase jezičkih resursa, odnosno kao potencijalne III generacije elektronskih korpusa.

Veb kao korpus — preispitivanje definicije korpusa

Izraz „veb kao korpus" je među prvima upotrebio Adam Kilgarif u naslovu svog rada u kome naziva veb „korpusom novog milenijuma" ([Kilgarriff, 2001]343). Koncept „veb kao korpus" je detaljno razrađen u [Kilgarriff & Grefenstette, 2003] i

pritom se ponovo otvora pitanje definicije korpusa uz oštro suprotstavljanje dotada dominantnoj definiciji korpusa ([McEnery et al., 2006]) opisanoj u odeljku 1.2:

> „Mekeneri i Vilson (kao i njihovi prethodnici) mešaju pitanja ‚Šta je korpus?' i ‚Šta je *dobar* korpus (za određene vrste istraživanja jezika)?', i zamućuju jednostavno pitanje ‚Da li je korpus x dobar za zadatak y?' semantičkim pitanjem ‚Da li je x uopšte korpus?'. Semantičko pitanje skreće pažnju (...) Da bi se semantičko pitanje ostavilo po strani, definicija korpusa bi trebalo da bude široka. Mi definišemo korpus prosto kao ‚kolekciju tekstova'. Ako se to čini preširoko, dozvoljavamo jednu kvalifikaciju koja se odnosi na domen i kontekste u okviru kojih se pojam koristi (...): Korpus je kolekcija tekstova koji su predmet istraživanja jezika ili književnosti." ([Kilgarriff & Grefenstette, 2003: 334])
>
> „Veb je reprezentativan samo u odnosu na sebe. Ali isto važi i za ostale korpuse, u ma kom jasnom značenju. Izdvajanje tog pitanja samo otkriva koliko je primitivno naše razumevanje te teme i neumoljivo vodi ka većim i samim tim zanimljivijim pitanjima o prirodi jezika i o tome kako ga modelirati." ([Kilgarriff & Grefenstette, 2003: 343])

Zastupnici koncepta „veb kao korpus" (WaC), poredeći veb sa uobičajenim elektronskim korpusima II generacije, navode kao njegove prednosti, pre svega, veličinu veba i dostupnost raznovrsnih tekstova (uključujući i nove tipove tekstova poput blogova, poruka e-pošte, SMS-poruka, itd.), mogućnost da se slobodno i besplatno pristupi ogromnoj količini tekstova, a za razliku od skupe i zahtevne izrade uobičajenih korpusa koji brzo zastarevaju, veb se neprekidno ažurira i uvećava po daleko manjoj ceni ([Hundt et al., 2007: 1–5]).

Kritičari koncepta WaC ističu ([McEnery & Hardie, 2012: 7-8], [Leech, 2007: 144]) sledeće nedostatke:

- veb je ogromna, „žanrovski nediferencirana masa" koja zahteva značajno ulaganje u obradu i klasifikaciju njegovih tekstova;

- statističke analize veba, poput informacija o učestanosti oblika leksema i njihovih sekvenci (v. odeljak 3.3, str. 196) moraju se uzeti sa rezervom pošto se nezanemarljiv broj tekstova na vebu delimično ili u celosti ponavlja;

- tekstovi na vebu sadrže raznovrsne greške (tipografske, pravopisne, sintaksne, faktografske, itd.) koje otežavaju analizu korpusa i dodatno obesmišljavaju rezultate statističkih metoda;

- pojedini tipovi tekstova nisu dovoljno zastupljeni, ukoliko su uopšte i zastupljeni (privatni diskurs, svakodnevna komunikacija, telefonski dijalozi, itd.), a pretraživači ne omogućavaju pristup rukopisima ili govornim tekstovima;

- teško je ili čak nemoguće ustanoviti poreklo tekstova na vebu, vremenski trenutak kad su nastali, jezik na kojem je tekst napisan (videti na primer [Zečević & Vujičić-Stanković, 2013]), da li je autor teksta koristio svoj maternji jezik, itd.;

- veb se neprekidno menja što onemogućava reprodukciju rezultata istraživanja posle dužeg vremenskog perioda (na primer, par godina).

Iako kritičari koncepta WaC priznaju da će veb verovatno „postati nezamenljiv u leksikografiji i leksičko-gramatičkim istraživanjima", kao i da „pobeđuje druge korpuse" kada su u pitanju upotrebe novih reči i kolokacije, ipak smatraju da veb neće dovesti do toga da se sistematično planiranje i uzorkovanje prilikom izrade korpusa tretiraju kao prevaziđena metodologija kreiranja jezičkih resursa ([Leech, 2007: 145]).

Zajednica korpusnih lingvista koji zastupaju koncept WaC[20] počev od 2005. godine održava godišnje radionice *Veb kao korpus* (Web as Corpus Workshop), dok su računarski lingvisti osnovali **Posebnu interesnu grupu za veb kao korpus Udruženja za računarsku lingvistiku** (eng. **Special Interest Group of the Association for Computational Linguistics on Web as Corpus**, skr. **ACL SIGWAC**)[21].

[20] `http://webascorpus.sourceforge.net`
[21] `http://sigwac.org.uk`

Veb-korpusi i projekat WaCky

Najistaknutiji primer veb-korpusa su rezultati projekta WaCky[22], „neformalnog konzorcijuma istraživača zainteresovanih za proučavanje veba kao izvora lingvističkih podataka" ([Baroni et al., 2009]). U periodu od 2005. do 2013. godine u okviru ovog projekta je nastalo više veb-korpusa (Tabela 1.4), veličine oko milijardu reči ili više, sastavljenih od dokumenata automatski preuzetih sa veba i delimično lingvistički anotiranih na morfološkom nivou[23]. Projekat WaCky je uticao na izgradnju sličnih korpusa za većinu jezika, posebno za jezike za koje uopšte nisu postojali korpusi[24].

Izgradnja veb-korpusa je najčešće poluautomatski proces koji počinje automatizovanim krstarenjem vebom (eng. crawling). Automatizovano krstarenje vebom koristi određenu heuristiku za izbor adresa stranica koje će se potom preuzeti sa veba i obraditi za potrebe korpusa. Na primer, adrese stranica mogu biti rezultat pretrage na osnovu upita u formi ručno kreirane liste reči koje karakterišu određeni tekstuelni domen, registar, žanr, ili pak liste reči automatski kreirane na osnovu njihove relativne učestanosti u izabranom referentnom korpusu. Obrada preuzetih stranica sa veba obuhvata filtriranje, brisanje duplikata i približnih duplikata dokumenata. Približni duplikati su stranice koje imaju gotovo identičan sadržaj, ali i prateće elemente po kojima se razlikuju poput reklama, informacija o posećenosti stranice i datumu poslednjeg ažuriranja, itd.

Zanimljivo je upoređivanje elektronskih korpusa II generacije koji nisu veb-korpusi i samih veb-korpusa. Istraživanje zasnovano na poređenju listi učestanosti korpusa ukWaC i BNC, uz primenu mere zasnovane na statističkom testu logaritamske verodostojnosti, je pokazalo da se ta dva korpusa razlikuju po proporcionalnoj zastupljenosti određenih tipova tekstova, odnosno žanrova, tj. da BNC sadrži „više

[22]Naziv projekta WaCky (Web-As-Corpus Kool Yinitiative) je akronim čiji prvi deo (WaC) znači *veb kao korpus*, dok je ostatak igra reči (*Kool Yinitiative* umesto *Cool Initiative* (srp. *opuštena inicijativa*)) sa ciljem da akronim izgleda isto kao engleska reč *wacky* (srp. *ćaknut, uvrnut, otkačen*).

[23]Korpusnim rečima su uglavnom pridružene informacije o vrsti reči i lemi.

[24]Na adresi `https://www.sketchengine.co.uk/documentation/wiki/Website/LanguageResourcesAndTools` se može naći opširna lista veb-korpusa kreirana za različite evropske jezike, između ostalih i za srpski jezik (veb-korpus srWaC). Polovinom oktobra 2013. godine izrada srWaC-a još uvek nije okončana, distribucija konačne verzije korpusa je planirana za kraj 2013. godine, a radna verzija veličine 500 miliona tokena je dostupna u dogovoru sa autorima (za detalje v. `http://nlp.ffzg.hr/resources/corpora/srwac/`). Izgradnja ovog korpusa izaziva polemike.

Tabela 1.4: Pregled veb-korpusa u okviru projekta WaCky (Gw označava milijardu reči, a Gt milijardu tokena). Više detalja se može naći na adresi `http://wacky.sslmit.unibo.it/doku.php?id=corpora` zvanične dokumentacije projekta.

jezik	**naziv veb-korpusa**	**veličina**
engleski	ukWaC	2 Gw
	PukWaC	2 Gw
	WaCkypedia_EN	0,8 Gt
francuski	frWaC	1,6 Gw
italijanski	itWaC	2 Gw
nemački	deWaC	1,7 Gw
	sdeWaC	0,88 Gw

fikcije i govornih tekstova", dok „ukWaC sadrži veći procenat tekstova vezanih za veb, obrazovanje i ‚javnu sferu'" ([Ferraresi et al., 2008]).

1.5 Nacionalni korpusi slovenskih jezika

Nacionalni korpusi su „opšti, referentni korpusi kojima bi trebalo da se reprezentuje nacionalni jezik jedne zemlje" ([Xiao, 2008: 383]). S obzirom da se u ovom radu opisuje izgradnja referentnog korpusa savremenog srpskog jezika, neophodno je najpre izložiti iskustva sličnih poduhvata, posebno kada su u pitanju jezici slovenske grupe kojoj srpski pripada.

Najistaknutiji primer nacionalnog korpusa u okviru II generacije elektronskih korpusa je svakako Britanski nacionalni korpus (BNC), već opisan u odeljku 1.1, po čijem uzoru su nastali svi nacionalni korpusi slovenskih jezika.

Početkom oktobra 2013. godine postoje nacionalni korpusi za sledeće slovenske jezike (uključujući srpski, koji će biti predstavljen u delu III ovog rada):[25]

- beloruski,
- bugarski,
- češki,
- hrvatski,

[25]Korpus makedonskog ([Ivanovska-Naskova, 2006]) je još uvek u izradi.

- poljski,
- ruski,
- slovački,
- slovenački i
- ukrajinski.

U nastavku će najpre biti izložene zajedničke karakteristike navedenih nacionalnih korpusa slovenskih jezika, a potom će za svaki korpus pojedinačno biti razmotrene njegove specifičnosti u odnosu na parametre korpusa opisane u odeljku 1.3.

Nosač Svi navedeni korpusi su elektronski.

Domen i namena Svi navedeni korpusi pretenduju da budu opšti, referentni korpusi, tj. da mogu da se koriste za raznovrsna lingvistička istraživanja jezika koji reprezentuju.

Obim (veličina) Većina predstavljenih nacionalnih korpusa slovenskih jezika, izuzimajući beloruski i ukrajinski, ima bar 100 miliona korpusnih reči, a u pojedinim slučajevima i nekoliko stotina miliona korpusnih reči.

Period Iako pretenduju da budu sinhroni korpusi, tekstovi koji su zastupljeni u nacionalnim korpusima slovenskih jezika pokrivaju intervale od nekoliko godina do nekoliko decenija.

Izvor/medijum S obzirom na složenost prikupljanja govornih tekstova, kao i u slučaju BNC-a kao uzora, u nacionalnim korpusima slovenskih jezika dominiraju pisani tekstovi.

Anotacija Većina nacionalnih korpusa slovenskih jezika poseduje morfosintaksičku anotaciju u vidu informacija o lemi i vrsti reči. Takođe, u većini slučajeva su dostupne i bibliografske informacije o korpusnim tekstovima.

Višejezičnost Svi nacionalni korpusi slovenskih jezika su jednojezični.

Beloruski N-korpus (Беларускі N-корпус)

Zvaničan sajt `http://bnkorpus.info`

Obim (veličina) 30 miliona tokena.

Struktura Korpus sačinjava 50 hiljada novinskih i književnih tekstova (poetski i prozni).

Anotacija i mogućnosti pretrage Korpus je delimično morfološki anotiran (prisutna je informacija o vrsti reči i lemi), a tekstovima su pridružene i bibliografske informacije (autor, stil/žanr, godine kada je tekst napisan, odnosno objavljen).

Bugarski nacionalni korpus (BulNC)

Bugarski nacionalni korpus (eng. **Bulgarian National Corpus**, skr. **BulNC**)[26] je nastao u periodu od 2001. do 2009. godine na Institutu za bugarski jezik *Prof. Ljubomir Andrejčin*[27] u Sofiji ([Koeva & Genov, 2011; Koeva et al., 2012; Kolkovska, Georgieva, Blagoeva & Kostova, 2012; Kolkovska, Koeva & Blagoeva, 2012]).

Zvaničan sajt `http://ibl.bas.bg/BGNC_BulCorpus_bg.htm`

Obim (veličina) BulNC se sastoji od preko 240 hiljada uzoraka tekstova na bugarskom jeziku čija ukupna veličina dostiže 1,2 milijarde tokena.

Period Izvorne verzije tekstova BulNC-a su nastale u periodu od 1945. godine do danas.

[26] U originalu *Български национален корпус.*
[27] U originalu Институт за български език *Проф. Любомир Андрейчин.*

Struktura Tekstovi koji su u originalu na bugarskom jeziku predstavljaju 37,1% korpusa BulNC, prevodi — 40,5%, dok se za preostalih 22,4% ne zna da li je u pitanju tekst u originalu ili prevod.

Tekstovi su klasifikovani po stilu[28], domenu[29] i žanru[30]. Detaljniji podaci o raspodeli tekstova po pojedinim stilovima se mogu naći na zvaničnom sajtu BulNC-a[31].

Izvor/medijum Pisani tekstovi čine 97,35% korpusa, dok su govorni tekstovi zastupljeni sa svega 2.65% (uglavnom su u pitanju predavanja, govori sa skupštinskih zasedanja i filmski podnapisi).

Najveći deo tekstova (97,5%) je prikupljen sa veba, automatskim ili ručnim preuzimanjem, dok je ostatak dobijen od autora tekstova ili njihovih izdavača.

Anotacija i mogućnosti pretrage BulNC koristi 27 kategorija za opis metapodataka koji su pridruženi njegovim tekstovima (naziv datoteke, putanja, datum dodavanja teksta u korpus, autor, dodatne informacije o autoru, prevodilac, dodatne informacije o prevodiocu, naslov teksta, godina nastanka teksta, godina publikovanja teksta, stil, žanr, domen[32], ključne reči, napomene, kvalitet, dostupnost, tip izvora, izvor, itd.).

BulNC je morfološki i semantički anotiran, tako da se tokom pretrage u upitu mogu koristiti vrednosti gramatičkih obeležja i oznake semantičkih relacija.

Morfološka anotacija je usklađena sa standardom Multext EAST (videti odeljak 2.4, str. 146), ali sintaksa upita koristi drugačije označavanje morfoloških kategorija (vrsta reči, rod, broj, lice, glagolski rod u širem smislu, određenost[33], itd.) i njihovih vrednosti. Na primer, upit `<хубав/F/{D=df} *{POS=N}>` pronalazi sve

[28]Stil je definisan kao opšta složena kategorija teksta koja kombinuje pojam registra, mode i diskursa. BulNC razlikuje sledeće stilove: administrativni, naučni, masovne medije, fikciju, neformalni, neformalni/fikcija (filmski podnapisi), popularna nauka, popularno.

[29]Svaki stil je podeljen u tematske domene, ali ponekad isti domen može da bude deo više stilova. Na primer, i naučni stil i novinski stil obuhvataju domene ekonomija i politika.

[30]Žanr u BulNC-u je povezan sa internim formalnim karakteristikama teksta.

[31]`http://ibl.bas.bg/en/BGNC_BulCorpus_en.htm`

[32]Postoje dve kategorije za opis domena, domen1 i domen2, ukoliko se tekst može svrstati u dva različita domena.

[33]Svojstvo koje opisuje da li oblik imenske reči u bugarskom jeziku sadrži određeni član ili ne.

određene (D=df) oblike (F) prideva *хубав* (srp. *lep*) za kojim sledi proizvoljna (*) imenica ({POS=N}).

Semantička anotacija omogućava pretraživanje sinonima, hiperonima i pojmova koji su u relaciji sličnosti. Na primer, хубав/S/ pronalazi sve sinonime prideva *хубав*.

U upitima se mogu koristiti i Bulovske operacije: konjunkcija (&), disjunkcija (|), implikacija (=>), ekvivalencija (<=>) i negacija (!).

Licenca BulNC je dostupan za akademsku i nekomercijalnu upotrebu. Ukoliko se u naučnim i stručnim radovima koriste primeri iz BulNC-a, obavezno je navesti odgovarajuću referencu[34].

Češki nacionalni korpus (ČNK)

Češki nacionalni korpus (češ. **Český národní korpus**, skr. **ČNK**) je akademski projekat koji je proizašao iz dogovora osmoro predstavnika sledećih institucija: Filozofskog fakulteta Karlovog univerziteta u Pragu, Matematičkog i fizičkog fakulteta Karlovog univerziteta u Pragu, Masarikovog univerziteta, Univerziteta Palackog u Olomoucu i Instituta za češki jezik Akademije nauka Češke Republike ([Klimova, 1996]). U svrhu izvođenja projekta, 1994. godine je osnovan poseban institut na Filozofskom fakultetu Karlovog univerziteta u Pragu — Institut za Češki nacionalni korpus (IČNK). IČNK je razvio više korpusa češkog jezika koji se mogu podeliti na grupu sinhronih korpusa i grupu dijahronih korpusa. Unija svih korpusa iz grupe sinhronih korpusa (označena skraćenicom SYN) obuhvata 16 korpusa (SYN2010, SYN2009PUB, SYN2006PUB, SYN2005, SYN2000, itd.)[35] koji se mogu pretraživati i zajedno (istovremeno) i odvojeno, dok grupu dijahronih korpusa čini samo jedan korpus (DIAKORP).

Zvaničan sajt http://ucnk.ff.cuni.cz

[34] „Българския национален корпус © Институт за български език", odnosno „Bulgarian National Corpus © Institute for Bulgarian Language".

[35] Broj u nazivu sinhronog korpusa označava godinu kada je završen rad na njegovoj izradi.

Obim (veličina) Ukupna veličina unije sinhronih korpusa je 1,3 milijarde reči. Najveći među njima je SYN2009PUB (700 miliona reči), slede SYN2006PUB (300 miliona reči) i SYN2010, SYN2005 i SYN2000 (svaki sa po 100 miliona reči), dok su preostali sinhroni korpusi daleko manji[36].

Period Dijahroni korpus DIAKORP obuhvata tekstove od XIII do XX veka, i to zaključno sa 1989. godinom (novinski i posebni tekstovi), odnosno zaključno sa 1944. godinom (fikcija)[37]. Sinhroni korpusi se nadovezuju na dijahroni korpus, tj. SYN2000 obuhvata tekstove u periodu od 1990. do 1999. godine, SYN2005 — tekstove od 2000. do 2004. godine, a SYN2010 — tekstove od 2005. do 2009. godine.

Struktura SYN2000, SYN2005 i SYN2010 su balansirani korpusi, dok su SYN-2006PUB i SYN2009PUB korpusi sačinjeni od tekstova iz novina i časopisa[38].

Izvor/medijum Od šesnaest sinhronih korpusa grupe SYN, pet su govorni korpusi (pojedinačne veličine u intervalu od 490 hiljada do milion reči), dok preostalih jedanaest korpusa sadrže pisane tekstove (ukupno oko 1,3 milijarde reči).

Anotacija i mogućnosti pretrage ČNK koristi pozicionu morfološku anotaciju, tj. svakoj korpusnoj reči je pridružen morfosintaksički opis takav da svaka pozicija u opisu predstavlja jednu od sledećih četrnaest[39] kategorija: vrstu reči, podvrstu reči, rod, broj, padež, rod prisvojnog prideva, broj prisvojnog prideva, lice, glagolsko vreme, stepen poređenja, negaciju, glagolsko stanje, varijantu (pravopisa, stila ili nečeg drugog), glagolski vid. Za pretragu korpusa koristi se NoSketchEngine (Manatee i Bonito), detaljno opisan u odeljku 4.3 (str. 222).

Licenca ČNK je dostupan za akademsku i nekomercijalnu upotrebu. Ukoliko se u naučnim i stručnim radovima koriste primeri iz ČNK-a, obavezno je navesti od-

[36]Njihova pojedinačna veličina se kreće u intervalu od 80 hiljada do 2 miliona reči.

[37]Za detalje videti `http://ucnk.ff.cuni.cz/english/diakorp.php`.

[38]Detaljniji opis strukture ovih korpusa može se naći na zvaničnoj prezentaciji ČNK-a: `http://ucnk.ff.cuni.cz/english/struktura.php`.

[39]U stvari, morfosintaksički opis ima šesnaest pozicija, ali su dve rezervisane i trenutno se ne koriste.

govarajuću referencu[40].

Hrvatski nacionalni korpus (HNK)

Projekat razvoja Hrvatskog nacionalnog korpusa (HNK) je započeo 1998. godine ([Tadić, 2002, 2006, 2009]). Nosilac projekta je Zavod za lingvistiku Filozofskog fakulteta Sveučilišta u Zagrebu.

Zvaničan sajt `http://www.hnk.ffzg.hr`

Obim (veličina) Trenutna verzija korpusa, HNK 3.0, napravljena 2013. godine, sadrži ukupno 101, 3 miliona tokena.

Period Tekstovi HNK-a pokrivaju period od 1990. godine do danas.

Struktura Cilj autora korpusa je da HNK bude balansiran korpus od 100 miliona reči, što po sopstvenom priznanju „još nije postignuto, ali predstavlja zacrtani cilj"[41]. Dok se ne postigne planirana struktura HNK-a (Tabela 1.5), svi tekstovi prikupljeni kao kandidati za HNK biće dostupni za pretragu.

Tekstovi su klasifikovani po mediju i vrsti publikacije (govorni, pisani, e-tekstovi, novine, itd.), žanru, domenu i temi (informativni, naučni, putopisi, politika, ekologija, sport, itd.), književnim vrstama fikcije (proza, drama, poezija), a posebnu klasu predstavljaju mešani tekstovi, tj. tekstovi koji se istovremeno mogu svrstati u više prethodno navedenih klasa[42].

Anotacija i mogućnosti pretrage HNK je morfološki anotiran u skladu sa standardom MULTEXT East (v. odeljak 2, str. 146).

Za pretragu korpusa se počev od 2005. godine koristi NoSketchEngine (Manatee i Bonito), detaljno opisan u odeljku 4.3 (str. 222).

[40]Na primer, ukoliko se citira celokupna grupa sinhronih korpusa SYN, treba koristiti referencu „Czech National Corpus - SYN. Institute of the Czech National Corpus, Praha. 21.10.2013. Accessible at WWW: <`http://www.korpus.cz`>."

[41]http://www.hnk.ffzg.hr/struktura.html

[42]Detalji se mogu naći na adresi `http://www.hnk.ffzg.hr/struktura.html`.

Tabela 1.5: Planirana struktura HNK

Vrsta teksta			%
	Informativni tekstovi		74,0
	Novine		37,0
		dnevne	22,0
		nedeljne	9,0
		polumesečne	6,0
	Časopisi, revije		16,0
		nedeljnici	9,0
		mesečnici	4,0
		dvo-, tromesečnici	3,0
	Knjige, brošure, pisma...		21,0
		publicistika	4,0
		popularni tekstovi	3,5
		korespondencija, efemera	0,5
		nauke i umetnost	13,0
	Imaginativni tekstovi (fikcija)		23,0
		romani	13,0
		pripovetke, novele, crtice	5,0
		ogledi	4,0
		dnevnici, (auto)biografije...	1,0
	Mešani tekstovi		3,0

Nacionalni korpus poljskog jezika (NKJP)

Nacionalni korpus poljskog jezika (polj. **Narodowy Korpus Języka Polskiego**, skr. **NKJP**) [Przepiórkowski et al., 2010] je nastao u periodu od decembra 2007. godine do decembra 2010. godine zajedničkim naporom četiri institucije koje su pre 2006. godine, svaka za sebe, razvijale svoje korpuse poljskog jezika[43]: **Instituta za računarstvo Poljske akademije nauka** (polj. **Instytut Podstaw Informatyki Polskiej Akademii Nauk**, skr. **IPI PAN**) u Varšavi, **Instituta za poljski jezik Poljske akademije nauka** (polj. **Instytut Języka Polskiego Polskiej Akademii Nauk**, skr. **IJP PAN**) u Krakovu, **Naučnog izdavaštva PWN** (polj. **Wydawnictwo Naukowe PWN**[44], skr. **PWN**) i **Laboratorije za korpusnu i računarsku lingvistiku Univerziteta u Lođu** (polj. **Pracownia Językoznawstwa Korpusowego i Komputerowego Uniwersytetu Łódzkiego**). Projekat izrade NKJP je podržalo Ministarstvo nauke i visokog obra-

[43]Korpus IPI PAN (`http://korpus.pl`), korpus PWN (`http://korpus.pwn.pl`) i korpus PELCRA (`http://korpus.ia.uni.lodz.pl`).

[44]PWN je skraćenica imena koju je institucija nosila do 1991. godine: *Państwowe Wydawnictwo Naukowe* (srp. *Nacionalno naučno izdavaštvo*).

zovanja Poljske.

Zvaničan sajt `http://nkjp.pl`

Obim (veličina) Planirana veličina korpusa je milijardu reči, pri čemu će u okviru NKJP-a postojati balansirani potkorpus od najmanje 300 miliona reči.

Izvor/medijum Govorni tekstovi veličine 30 miliona reči predstavljaće deo balansiranog potkorpusa od 300 miliona reči, tj. odnos pisanih i govornih tekstova će biti 9 : 1, isto kao u BNC-u.

Anotacija i mogućnosti pretrage Anotacija je sprovodena u dve faze. U prvoj fazi je izabrani potkorpus balansiranog dela NKJP-a ukupne veličine milion reči ručno anotiran lingvistički i strukturno, a tekstovima su pridruženi bibliografski metapodaci ([Przepiórkowski & Murzynowski, 2009]). Lingvistička anotacija je višeslojna ([Bański & Przepiórkowski, 2009]): morfosintaksička ([Woliński, 2006]), sintaksička ([Głowińska & Przepiórkowski, 2010; Savary et al., 2010])[45], semantička ([Młodzki & Przepiórkowski, 2011])[46]. Ručno anotirani potkorpus je potom iskorišćen za treniranje alata koji se koriste za automatsku morfosintaksičku anotaciju ostatka NKJP-a.

Trenutno postoje dva sučelja za pretragu NKJP-a, koji su inspirisani sučeljima za pretragu korpusa PELCRA i korpusa IPI PAN. Prvo sučelje (PELCRA) je zasnovano na kombinaciji programa Apache Lucene[47] i tehnologije relacionih baza podataka, dok potonje sučelje (IPI PAN) koristi alat Poliqarp ([Janus & Przepiórkowski, 2007a,b]).

Nacionalni korpus ruskog jezika (NKRJ)

Još tokom osamdesetih godina XX veka u bivšem Sovjetskom Savezu je započeta digitalizacija tekstova ruske književnosti, posebno autora iz XIX veka, ali je tek ok-

[45] Označene su sintaksičke reči, sintaksičke grupe i imenovani entitati.
[46] Svakoj korpusnoj reči je pridruženo odgovarajuće značenje.
[47] `http://lucene.apache.org`

tobra 2000. godine na seminaru Centra za lingvističku dokumentaciju pod rukovodstvom Vladimira Aleksandroviča Plungjana (Владимир Александрович Плунгян) i Mihaila Aleksandroviča Danielja (Михаил Александрович Даниэль) Sergej Aleksandrovič Šarov (Сергей Александрович Шаров) među prvima izneo ideju da se napravi reprezentativan i balansiran korpus ruskog jezika od najmanje 100 miliona reči po uzoru na Britanski nacionalni korpus ([Сичинава, 2005]). Sam projekat izgradnje **Nacionalnog korpusa ruskog jezika** (rus. **Национальный корпус русского языка**, skr. **NKRJ**) je započela Ruska akademija nauka 2003. godine u okviru programa Filologija i informatika, ([NKRJ, 2005; Плунгян et al., 2009],[Sharoff, 2006], [Рахилина, 2009]).

Zvaničan sajt `http://www.ruscorpora.ru`

Obim (veličina) Iako ime NKRJ sugeriše da se radi o jednom korpusu, u pitanju je više različitih korpusa koji se tretiraju kao potkorpusi NKRJ-a:

- osnovni ili glavni korpus ruskog jezika, ukupne veličine 230 miliona reči, se kreira po ugledu na Britanski nacionalni korpus; međutim, tipologija tekstova, uključujući i proporcionalnu zastupljenost pojedinih tipova teksta, su razvijeni nezavisno od BNC-a;
- novinski korpus ruskog jezika, ukupne veličine preko 170 miliona reči, je korpus članaka iz sredstava javnog informisanja nastalih posle 2000. godine;
- korpusi paralel(izova)nih tekstova (ukupno oko 38 miliona reči): englesko-ruski, nemačko-ruski, francusko-ruski, špansko-ruski, italijansko-ruski, poljsko-ruski, ukrajinsko-ruski, belorusko-ruski ([Добровольский et al., 2005]);
- govorni korpus ruskog jezika od 10,3 miliona reči ([Гришина, 2005; Гришина & Савчук, 2009]);
- dijalektski korpus ruskog jezika od 195 hiljada reči ([Летучий, 2005, 2009]);
- akcenatski korpus ruskog jezika od 12,7 miliona reči ([Гришина, 2009]);

- multimedijalni korpus ruskog jezika MURKO od 3,3 miliona reči ([Гришина, 2005, 2009; Grishina, 2009]);
- korpus poetskih tekstova ruskog jezika od 9,2 miliona reči ([Гришина et al., 2009]);
- školski (obrazovni) korpus ruskog jezika od preko 664 hiljade reči ([Савчук & Сичинава, 2009]);
- korpus istorijskih tekstova od 4,7 miliona reči (crkvenoslovenski, staroruski (rus.*древнерусские*) i srednjeruski tekstovi).

Celokupan NKRJ sadrži 384 miliona reči. Ovde će, pre svega biti reči o osnovnom ili glavnom korpusu ruskog jezika.

Period Osnovni (pot)korpus NKRJ-a sačinjavaju:

- savremeni pisani tekstovi (od 1950. godine do danas),
- savremeni govor (od 1950. godine do danas) i
- tekstovi perioda od sredine XVIII veka do 1950. godine.

Izvor/medijum Dominiraju pisani prozni tekstovi originalno nastali na ruskom jeziku.

Anotacija i mogućnosti pretrage Tekstovi osnovnog korpusa su automatski morfološki anotirani i pridruženi su im odgovarajući bibliografski podaci. Razrešavanje morfološke višeznačnosti prouzrokovane homografijom, semantička anotacija i akcentovanje se obavljaju ručno na potkorpusu osnovnog korpusa koji je dostigao veličinu od šest miliona reči.

Osim osnovnog korpusa, još jedan od potkorpusa NKRJ-a, **Duboko anotirani korpus** (rus. **Глубоко аннотированный (синтаксический) корпус, Синтаксически размеченный корпус русского языка**, skr. **SinTagRus**), je detaljno anotiran, pri čemu je, osim morfološke i semantičke anotacije, primenjena i sintaksička anotacija ([Apresjan et al., 2006]). Sintaksička anotacija SinTagRus-a je

zasnovana na gramatici zavisnosti Igora Aleksandroviča Meljčuka (Игорь Александрович Мельчук), detaljnije opisanoj u odeljku 2.3, str. 105. U okviru SinTagRus-a su razrešene i morfološke i sintaksičke višeznačnosti. SinTagRus se trenutno sastoji od 539 tekstova, skoro 50 hiljada rečenica i preko 750 hiljada reči.

Slovački nacionalni korpus (SNK)

Slovački nacionalni korpus (slov. **Slovenský národný korpus**, skr. **SNK**) je projekat čiji je cilj izgradnja elektronskih lingvističkih resursa za slovački jezik ([Horák et al., 2004; Šimková, 2005]). Projekat sprovodi Odeljenje za Slovački nacionalni korpus pri Lingvističkom institutu *Ljudovit Štur* Slovačke akademije nauka u Bratislavi.

Zvaničan sajt `http://korpus.juls.savba.sk`

Obim (veličina) Trenutna verzija korpusa, SNK prim- 6.0, kreirana 2013. godine, sadrži 1,155 milijardi tokena, odnosno oko 880 miliona korpusnih reči. Pored nje, dostupna je i prethodna verzija korpusa, SNK prim-5.0 čija je veličina 719 miliona tokena.

Period SNK obuhvata pisane tekstove objavljene od 1955. godine do danas, s obzirom da je 1953. godine izvršena (do danas) poslednja pravopisna reforma slovačkog jezika.

Struktura Najveći deo korpusa (77,8%) čine novinski tekstovi, 9,8% fikcija, 11% stručni tekstovi i 1,4% ostali tekstovi.

Anotacija i mogućnosti pretrage SNK je automatski lematizovan i morfosintaksički anotiran ([Gianitsová, 2005])[48].

Tekstovima korpusa su pridružene i informacije o žanru, stilu, kao i odgovarajuće bibliografske informacije[49].

[48] Oznake vrednosti morfoloških kategorija koje se pridružuju korpusnim rečima su detaljno objašnjene na adresi `http://korpus.juls.savba.sk/morpho.html`.

[49] `http://korpus.juls.savba.sk/bibstyle.html`

Za pretragu korpusa se koristi NoSketchEngine (Manatee i Bonito), detaljno opisan u odeljku 4.3 (str. 222).

Licenca SNK je dostupan za akademsku i nekomercijalnu upotrebu. Ukoliko se u naučnim i stručnim radovima koriste primeri iz SNK-a, obavezno je navesti odgovarajuću referencu.[50]

Korpus slovenačkog jezika FidaPLUS

FidaPLUS ([Arhar et al., 2007]) je rezultat projekta nadogradnje i proširenja korpusa slovenačkog jezika FIDA ([Erjavec, Gorjanc & Stabej, 1998]), izgrađenog u periodu od 1997. do 2000. godine. U projektu su učestvovali Filozofski fakultet i Fakultet društvenih nauka Univerziteta u Ljubljani, kao i Institut Jožef Stefan[51].

Zvaničan sajt `http://www.fidaplus.net`

Obim (veličina) FidaPLUS trenutno sadrži preko 620 miliona reči.

Period Tekstovi korpusa FidaPLUS su nastali u periodu od 1979. do 2006. godine[52].

Izvor/medijum Najveći deo korpusa FidaPLUS čine pisani tekstovi (Tabela 1.6). Raspodela tekstova po tipu teksta (žanru) je dostupna na adresi `http://www.fidaplus.net/Info/Info_main_statistike_3_eng.html`.

Anotacija i mogućnosti pretrage FidaPLUS je lematiziran i morfološki anotiran korpus u skladu sa standardom MULTEXT East (v. odeljak 2, str. 146). Prilikom pretrage se koriste drugačije oznake u odnosu na MULTEXT East, ali je notacija takođe poziciona. Na primer ([Arhar, 2007]):

[50] „Slovenský národný korpus – prim-6.0-public-all. Bratislava: Jazykovedný ústav Ľ. Štúra SAV 2013. Dostupný z WWW: `http://korpus.juls.savba.sk`".

[51] `http://www.ijs.si/`

[52] Raspodela tekstova korpusa FidaPLUS po vremenskom periodu je dostupna na adresi `http://www.fidaplus.net/Info/Info_main_statistike_1_eng.html`.

Tabela 1.6: Raspodela izvora tekstova u korpusu FidaPLUS.

izvor	broj reči	%
Internet	7.682.895	1,24%
knjige	54.306.387	8,74%
novine	405.347.516	65,26%
časopisi	144.494.504	23,26%
ostalo	9.318.698	1,50%
ukupno	621.150.000	100,00%

- upit `#1pisati` predstavlja pretragu po lemi (`#1`) i u ovom slučaju pronalazi sve oblike glagola `pisati`;
- upit `#2slmet??n` predstavlja pretragu po vrednostima morfoloških kategorija (`#2`) i u ovom primeru pronalazi sve oblike vlastitih (`l`) imenica (`s`) muškog roda (`m`) u akuzativu (`t`) jednine (`e`) koje predstavljaju nešto neživo (`n`)[53].

Korpusi ukrajinskog jezika

Tri nezavisne institucije u Ukrajini razvijaju svoje nacionalne korpuse ukrajinskog jezika:

- Ukrajinski jezičko-informacioni fond pri Narodnoj akademiji nauka Ukrajine u Kijevu, pod rukovodstvom Vladimira Širokova (Володимир Анатолійович Широков), razvija **Ukrajinski nacionalni lingvistički korpus** (ukr. **Український національний лінгвістичний корпус**, skr. **UNLK**) [Широков et al., 2005];
- Laboratorija za računarsku lingvistiku Nacionalnog univerziteta *Taras Ševčenko* u Kijevu, pod rukovodstvom Natalije Darčuk (Наталія Петрівна Дарчук), razvija **Korpus ukrajinskog jezika** (ukr. **Корпус української мови**, skr. **KUM**) [Дарчук, 2012a];
- Institut ukrajinskog jezika pri Narodnoj akademiji nauka Ukrajine u Kijevu, pod rukovodstvom Orisje Demske-Kuljčicke (Орися Демська-Кульчицька), razvija **Nacionalni korpus ukrajinskog jezika** (ukr. **Національний корпус української мови**, skr. **NKUM**) [Демська-Кульчицька, 2005].

[53]Pozicije čije vrednosti nisu bitne su označene upitnicima.

Zvaničan sajt

UNLK `http://lcorp.ulif.org.ua/virt_unlc/`

KUM `http://www.mova.info/corpus.aspx`

NKUM `http://nkum.nm.ru/`

Obim (veličina)

UNLK se sastoji iz opšteg korpusa (76 miliona reči) i korpusa zakonodavnih tekstova (18 miliona reči)[Kotsyba, 2013])[54];

KUM KUM se sastoji od četiri potkorpusa koji se mogu pretraživati isključivo nezavisno: potkorpusa narodne književnosti, potkorpusa proznih književnih tekstova, potkorpusa poetskih književnih tekstova i potkorpusa neknjiževnih tekstova. Ukupna veličina korpusa je 17 miliona reči ([Дарчук, 2012a]);

NKUM planirana veličina korpusa je 20 miliona reči[55], korpus je još uvek u izgradnji.

Anotacija i mogućnosti pretrage

UNLK Tekstovima UNLK-a su pridružene ne samo odgovarajuće bibliografske informacije, već i liste ključnih reči koje karakterišu tekstove, tako da se korisniku prilikom pretrage nudi mogućnost da suzi pretragu na tekstove koji sadrže izabrane ključne reči. UNLK je lematiziran i podržava pretragu po lemi, njenim sinonimima, nizu lema (koje se u korpusu pojavljuju u proizvoljnom redosledu, zadatom redosledu ili na određenom rastojanju). Morfosintaksički opisi su opisani u [Широков et al., 2005: 420–438], ali pretraga po njima nije javno dostupna.

[54]Na zvaničnom sajtu UNLK je moguće samo preuzimanje klijentske aplikacije koja koristi veb servise za pristup i pretragu korpusa. S obzirom da nije jasno gde se korisnik može registrovati, a bez korisničkog naloga i lozinke se ne može pristupiti korpusu, većina podataka izloženih o UNLK je prikupljena iz sekundarnog izvora ([Kotsyba, 2013]) koji referiše na [Широков et al., 2005] i [Широков et al., 2011].

[55]`http://nkum.nm.ru/NKUM.htm`

KUM KUM je anotiran kako bibliografskim podacima, tako i morfološki ([Дарчук, 2012b]). Uz pretragu po lemi i vrsti reči, dostupna je i pretraga po nekim morfološkim kategorijama (rod, broj, padež).

NKUM NKUM nije javno dostupan jer je još u izgradnji.

1.6 Srpska korpusna lingvistika

I pre osamdesetih godina XX veka, kada je pojam *korpusna lingvistika* postao deo naučne terminologije, započeli su radovi na stvaranju jezičkih resursa srpskog jezika namenjeni tadašnjim računarima u svrhu automatske obrade srpskog. Kako primećuje Vitas [2010: str. 257] u svom osvrtu na istorijat resursa i metoda za obradu srpskog, „Džordžtaunski projekat izgradnje automatskog prevodioca s ruskog na engleski obeležio je rane početke istraživanja ove oblasti [mašinsko prevođenje, prim. aut.] u svetu, pa i u ondašnjoj Jugoslaviji". Na Institutu za eksperimentalnu fonetiku i patologiju govora u Beogradu je u periodu od 1957. do 1962. godine, kao deo šireg projekta kojim je rukovodio Đorđe Kostić, razvijan preelektronski dijahroni korpus srpsko-hrvatskog jezika koji je trebalo da obuhvati period od XII veka do savremenog jezika ([Kostić, 2001]). Projekat je imao oko 400 saradnika, među njima 80 lingvista koji su ručno anotirali korpus od 11 miliona reči, pridružujući svakoj korpusnoj reči lemu i informacije o morfološkim kategorijama (rod, broj, padež, lice, glagolsko vreme, itd.). Kao posledica negativnog izveštaja **Savetodavnog komiteta za automatsku obradu jezika** (eng. **Automatic Language Processing Advisory Committee**, skr. **ALPAC**), poznatijeg i kao *Pirsov izveštaj*[56] ([Pierce et al., 1966]), finansiranje mnogih projekata širom sveta posvećenih automatskom prevođenju je obustavljeno, pa je tako prestao sa radom i Kostićev projekat ([Hutchins, 1986]), a njegovi resursi su ostali nedigitalizovani sve do sredine devedesetih godina XX veka.

Do osamdesetih godina XX veka „u Srbiji su istraživanja bila usredsređena na metode prepoznavanja govora bez upotrebe ozbiljnijih lingvističkih resursa" ([Vitas, 2010: 258]). Krajem sedamdesetih godina XX veka u Srbiji na Matematičkom

[56] Alternativni naziv potiče od prezimena rukovodioca ALPAC-a, Džona R. Pirsa.

institutu Srpske akademije nauka i umetnosti započinje sa radom Stalni seminar za matematičku i računarsku lingvistiku. Ovaj seminar je predstavljao početak računarske lingvističke škole u Srbiji čiji je osnivač i rukovodilac Duško Vitas. U okviru te škole je nastala i **Grupa za jezičke tehnologije** (eng. **Human Language Technology Group**, skr. **HLT Group**) Univerziteta u Beogradu, koja sve vreme svog postojanja radi na razvijanju jezičkih resursa i alata za automatsku obradu srpskog jezika. Tako je već 1981. godine započeo rad na izgradnji Korpusa savremenog srpskog jezika u okviru projekta *Matematička i računarska lingvistika* ([Vitas, 1990]). Tokom rada na izgradnji jezičkih resursa i alata HLT je uspostavila brojne kontakte sa inostranim stučnjacima, od kojih je Volfgang Tojbert (Wolfgang Teubert) sa Instituta za nemački jezik u Manhajmu posebno pomogao na početku njenog rada, održavši seminar o korpusima.

Krajem osamdesetih godina XX veka Grupa za jezičke tehnologije je uspostavila vezu sa Morisom Grosom (Maurice Gross), rukovodiocem *Laboratorije za automatsku dokumentaciju i lingvistiku* (fr. *Laboratoire d'Automatique Documentaire et Linguistique*, skr. *LADL*), što je omogućilo da tokom devedesetih godina XX veka otpočne razvoj morfološkog elektronskog rečnika srpskog jezika (v. odeljak 2, str. 145 i [Krstev, 2008]). U tom periodu je razvoj računarske lingvistike usporen s obzirom na građanske ratove u bivšoj Jugoslaviji i međunarodnu izolaciju Srbije.

U periodu od 1996. do 2003. godine, kao rezultat saradnje Instituta za eksperimentalnu fonetiku i patologiju govora i Laboratorije za eksperimentalnu psihologiju pri Filozofskom fakultetu Univerziteta u Beogradu, Aleksandar Kostić, sin Đorđa Kostića, počinje da rukovodi radom na obnovi projekta svog oca ([Kostić, 2012]). Saradnici na obnovljenom projektu su digitalizovali postojeće resurse projekta Đorđa Kostića, sa ciljem da ih uključe u odgovarajući elektronski dijahroni korpus, Korpus srpskog jezika (KSJ), kao i da prošire KSJ tekstovima savremenog jezika[57]. Dobijeni dijahroni korpus KSJ obuhvata tekstove podeljene u pet vremenskih delova:

1. period od XII do XVIII veka;

2. jezik XVIII veka (Dositej Obradović, Milovan Vidaković, Joakim Vujić, itd.);

[57] Originalni resursi Đorđa Kostića obuhvataju tekstove do 1957. godine.

3. sabrana dela Vuka Stefanovića Karadžića;

4. tekstove iz druge polovine XIX veka (Branko Radičević, Njegoš, Jovan Jovanović Zmaj, Đura Jakšić, Marko Miljanov);

5. savremeni jezik (dnevna štampa, naučna i politička literatura, prozni i poetski tekstovi).

Iako postoji zvaničan sajt KSJ-a[58], KSJ nije javno dostupan preko veba. Aleksandar Kostić je objavio papirna izdanja listi učestanosti nekoliko srednjevekovnih tekstova ([Kostić, 2012]), uz koje su priložene i elektronske verzije tih tekstova (sa mogućnošću pretrage teksta po morfološkim kategorijama) i listi učestanosti (uz mogućnost pretrage po azbučnom redu, frekvenciji, itd.).

Posle 2001. godine se intenzivira rad Grupe za jezičke tehnologije. Ona učestvuje u nizu domaćih i međunarodnih projekata i, između ostalog[59], konstruiše prvu javno dostupnu verziju Korpusa savremenog srpskog jezika (SrpKor) Krstev & Vitas [2005] o čijoj će izgradnji biti više reči u delu III ovog rada.

Nagli razvoj i popularnost korpusne lingvistike počev od kraja osamdesetih godina XX veka, kao i prepoznavanje njenog značaja u oblastima računarske lingvistike, su doprineli da se tokom poslednjih decenija razvoju korpusa posvete i lingvisti i računarski lingvisti u Srbiji. Iz okvira ovog rada izlaze sve aktivnosti na tom polju, ali neke od njih treba i spomenuti.

AlfaNum je projekat koji je okupio Grupu nastavnika i saradnika Katedre za telekomunikacije i obradu signala, Fakulteta tehničkih nauka u Novom Sadu, pod rukovodstvom Vlade Delića[60]. S obzirom da svi saradnici AlfaNuma nisu mogli biti zaposleni na Univerzitetu u Novom Sadu, projekat je prerastao u preduzeće *AlfaNum d.o.o.* koje se bavi **automatskim prepoznavanjem govora** (eng. **Automatic Speech Recognition**, skr. **ASR**), identifikacijom i verifikacijom govornika, kao i **automatskom sintezom govora na osnovu teksta** (eng. **Text-to-Speech**, skr. **TTS**) na srpskom jeziku ([Sečujski et al., 2002], [Sečujski et al., 2011],

[58] `http://www.serbian-corpus.edu.rs/indexns.htm`

[59] Videti na primer *Jezičke tehnologije - resursi i alati* `http://poincare.matf.bg.ac.rs/~cvetana/LT-pregled.html`.

[60] `http://alfanum.ftn.uns.ac.rs/frameset_results_srpski.htm`

[Popović et al., 2013], [Delić et al., 2013]). S obzirom da savremeni ASR i TTS-sistemi koriste statističko modeliranje na osnovu korpusa, AlfaNum razvija svoje govorne i pisane korpuse, morfološki rečnik i alate za automatsku morfološku anotaciju teksta. U osnovi svojih istraživanja AlfaNum koristi korpus od 11 hiljada rečenica i 200 hiljada reči, ručno anotiranih tako da sadrže informacije o morfološkim kategorijama i akcentuaciji ([Sečujski, 2009; Sečujski & Delić, 2008]). Svoj ručno anotirani korpus AlfaNum koristi kao skup za treniranje alata za automatsku morfološku anotaciju teksta, pomoću kojih se anotiraju veći korpusi za potrebe razvijanja ASR i TTS-sistema.

Balkanološki institut Srpske akademije nauka i umetnosti u saradnji sa Grupom za jezičke tehnologije Univerziteta u Beogradu razvija multimodalni korpus posvećen kulturnoj baštini Balkana ([Tanasijević et al., 2012]). Korpus sadrži snimljenu terensku građu (audio, video materijali i fotografije) koja je digitalizovana i anotirana tako da se može pretraživati po prostornim kriterijumima (lokaciji gde je snimak načinjen — opštini, naselju) i neprostornim kriterijumima: tipu građe (tekst, audio, video, fotografija), autoru snimka, jeziku i nacionalnosti snimljenih učesnika, kao i grupi kojoj pripadaju. Razgovori učesnika su dostupni za pregled i kao audio i video materijali i kao transkripti. Pojedini razgovori predstavljaju dijalektske govorne tekstove, za koje postoji i odgovarajuća verzija na standardnom jeziku. Takođe, za pojedine tekstove je dostupan i prevod na neke od ostalih balkanskih jezika, koji se mogu istovremeno pregledati, tako da se korpus može koristiti i kao paralel(izova)ni.

Katedra za opštu lingvistiku Filološkog fakulteta u Beogradu u okviru kurseva na osnovnim studijama nudi Korpusnu lingvistiku kao stručno-aplikativni predmet. U okviru kursa studenti se osposobljavaju, pre svega, za praktičan rad sa postojećim korpusima (pretraga i analiza), a upoznaju se i sa najvažnijim teorijskim i metodološkim pitanjima koja se odnose na konstrukciju korpusa[61]. Takođe, u okviru pojedinih predmeta na Katedri za opštu lingvistiku studenti pomažu u prikupljanju tekstova za govorni korpus ([Polovina & Panić Cerovski, 2012, 2013]), tako što tokom izrade seminarskog rada preslušavaju audio i pregledaju video snimke, a potom transkribuju govor i prateće fenomene poput pauza, uzdaha, itd.

[61] `http://www.fil.bg.ac.rs/katedre/opstaling/nastava1.html`

1.7 Cilj rada

U ovom radu je predstavljeno istraživanje mogućnosti za izgradnju elektronskog, dinamičkog, opšteg, sinhronog, balansiranog, anotiranog korpusa savremenog srpskog jezika, pretežno sastavljenog od pisanih ekavskih tekstova, kao i pratećih višejezičnih paralelnih korpusa (englesko-srpski, francusko-srpski, itd.).

U istraživanju se polazi od sledećih hipoteza:

- moguće je automatski anotirati korpus odgovarajućim morfološkim i strukturnim obeležjima korišćenjem postojećeg morfološkog elektronskog rečnika srpskog i raspoloživih programa zasnovanih na statističkim metodama;
- moguće je poluautomatsko uparivanje verzija istog teksta na različitim jezicima korišćenjem raspoloživih programa zasnovanih na statističkim metodama.

Očekuju se sledeći naučni rezultati:

- klasifikacija tekstova za korpus prema osnovnim funkcionalnim stilovima;
- pregled mogućnosti morfološke i strukturne anotacije korpusa.

Deo II

Kreiranje i analiza korpusa

2

Prethodna obrada korpusa

U ovom poglavlju se razmatra niz radnji neophodan za kreiranje elektronskih korpusa[1]:

- prikupljanje, digitalizacija i klasifikacija tekstova za korpus (odeljak 2.1);
- konverzija korpusnih tekstova u odgovarajući format elektronskog teksta (odeljci 2.1, 2.2)[2];
- lingvistička obrada i anotacija elektronskih tekstova za korpus (odeljci 2.3 i 2.4);
- indeksiranje i kompresija tekstova korpusa (odeljak 2.5).

S obzirom da se tek poslednjom grupom navedenih radnji (indeksiranje i kompresija) efektivno kreira korpus spreman za pretragu i analizu, pomenute radnje se ponekad označavaju zajedničkim imenom **prethodna obrada korpusa** (eng. **corpus preprocessing**).

[1] Poglavlje se sa posebnim naglaskom bavi kreiranjem korpusa pisanih tekstova.

[2] Konverzija konkretnih formata elektronskog teksta u format čistog teksta je obrađena u odeljku 6.2.

2.1 Prikupljanje, digitalizacija i klasifikacija tekstova

Prikupljanje tekstova je radnja kojom započinje kreiranje (elektronskog) korpusa. U slučaju da su kreatori korpusa prethodno usaglasili okvir uzorkovanja/okvir uzorka (v. odeljak 1.2), tj. popis tekstova koji će sačinjavati korpus, njihov sledeći zadatak je pribavljanje samih tekstova, dok se u slučaju oportunističkog korpusa (v. odeljak 1.3, str. 25), tekstovi prikupljaju *ad hoc*, tj. u korpus se uvršćuje bilo koji relevantni elektronski tekst do kog se lako dolazi.

Nezavisno od prirode korpusa, kreatori korpusa moraju imati zakonsko pravo da koriste prikupljene tekstove.

Takođe, bez obzira na format u kome su se tekstovi nalazili tokom prikupljanja (rukopis, štampani tekst, digitalna slika teksta, elektronski tekst, itd.), oni mogu postati deo elektronskog korpusa isključivo kao elektronski tekst (v. odeljak 2.2). Stoga prilikom prikupljanja tekstova, prednost treba dati izvorima dostupnih elektronskih tekstova, kako bi se izbegle nepotrebne konverzije iz drugih (nedigitalnih i digitalnih) formata u format elektronskog teksta.

Autorska prava

Većina tekstova koji se uvrštavaju u korpus — pre svih novinski, naučni i književnoumetnički — je zaštićena zakonom o autorskim i srodnim pravima, što se svodi na zabranu umnožavanja i reprodukovanja autorskog dela na bilo koji način, ni u celini, niti u delovima, bez dozvole nosilaca autorskog prava. Kreatori korpusa su u prilici da naruše zakon o autorskim i srodnim pravima prilikom pripreme korpusnih tekstova (digitalizacija, konverzije digitalnih formata, anotacija, itd.), kao i tokom distribucije i eksploatacije korpusa, odnosno stavljanjem korpusa na uvid drugim istraživačima. Kako ne bi narušili zakon, kreatori korpusa su dužni da pre prikupljanja i obrade tekstova provere da li je tekst zaštićen autorskim pravom ili ne, a u slučaju da jeste, jedino što im preostaje je da provere da li postoji izdanje teksta čije je autorsko pravo isteklo. Ukoliko su kreatori korpusa posebno zainteresovani za određeno izdanje teksta zaštićeno autorskim pravom, neophodno je postići spo-

razum sa nosiocima autorskih prava, tj. dobiti dozvolu za upotrebu teksta kao dela elektronskog korpusa. Tom prilikom kreatori korpusa treba jasno da specifikuju ([Barnbrook, 1996: 33]):

- za koju vrstu istraživanja im je potreban tekst;
- opis medijuma i formata koji će koristiti za skladištenje teksta;
- listu svih vrsta publikacija koje će se proizvesti na osnovu korišćenja teksta;
- formulaciju zahvalnosti nosiocu autorskog prava, uključujući i reference na njegovo autorsko delo kao sastavni deo proizvedenih publikacija.

S obzirom da je i korpus autorsko delo, njegovo umnožavanje, distribucija i eksploatacija se takođe uređuje nekom od licenci kojom se štite autorska prava kreatora korpusa (v. odeljak 4.1, str. 199). Kakav će sporazum sklopiti kreatori korpusa sa nosiocima autorskih prava na tekstovima koje žele da uvrste u korpus zavisi i od toga da li će korpus biti korišćen u komercijalne svrhe ili ne, odnosno od licence samog korpusa ([Kennedy, 1998: 77], [McEnery et al., 2006: 125]). U slučaju da se korpus koristi u komercijalne svrhe, nosioci autorskih prava na korpusnim tekstovima će sasvim sigurno zahtevati finansijsku naknadu. U pojedinim slučajevima nosioci autorskih prava daju dozvolu za korišćenje teksta dok se istraživanje ne završi, a onda zahtevaju uništavanje elektronske verzije teksta ([Kennedy, 1998: 77]).

Za razliku od komercijalnih korpusa, kod kojih se naknada nosiocima autorskih prava finansira na osnovu naknade za korišćenje korpusa, kreatori nekomercijalnih korpusa najčešće ne raspolažu finansijskim sredstvima koja bi bila dovoljna za izmirenje obaveza koje propisuje zakon o autorskim i srodnim pravima. Kreatori prvih nekomercijalnih elektronskih korpusa, poput Braunovog korpusa ili LOB-a, su bili prinuđeni da se upuste u „obimnu prepisku sa nosiocima autorskih prava", objašnjavajući da je korpus „naučni poduhvat" i da ne raspolažu sredstvima da plate naknadu za korišćenje autorskih dela kao tekstova za korpus ([Kučera, 2002: 307–308]). Kao rezultat, „nosioci autorskih prava su velikodušno dozvolili njihovo korišćenje [uzoraka teksta, prim. prev.] bez plaćanja naknade uz sporazum da će se [Braunov, prim. prev.] Korpus prvenstveno koristiti za akademska istraživanja

u lingvistici, stilistici i drugim relevantnim disciplinama" ([Francis & Kučera, 1964]). S obzirom da Braunov korpus obuhvata 500 uzoraka različitih tekstova, jasno je da posao prikupljanja potrebnih dozvola od nosilaca autorskih prava nije bio nimalo jednostavan, osim u slučajevima kada je od istog nosioca autorskog prava (na primer izdavača) dobijena blanko dozvola za korišćenje više tekstova.

Sa pojavom II generacije korpusa, sa Britanskim nacionalnim korpusom kao tipičnim predstavnikom (v. odeljak 1.1), u poređenju sa Braunovim korpusom veličina korpusa je porasla 100 puta, a broj uzoraka tekstova preko osam puta, tako da je i problem sa autorskim pravima na korpusnim tekstovima postao za red veličine složeniji.

Međutim, tek je nagli razvoj Interneta, posebno veba kao jednog od njegovih najvažnijih servisa, ukazao koliko je pitanje autorskih prava složeno. Zakonodavstvo je pokušalo da razvije mehanizme koji će moći da se primene i na nove tehnologije, pa se tako i stranice i drugi resursi na vebu mogu naći pod zaštitom zakona o autorskim pravima, što često nailazi na otpor korisnika veba.

U slučaju da se radi o tekstovima koje kreatori korpusa preuzimaju sa veba da bi ih uključili u korpus, problem sa autorskim pravima se svodi na sledeće:

> „Zakoni o autorskim pravima se primenjuju na dokumente na vebu na isti način kao i na štampane dokumente, tj. nelegalno je preuzeti tekst sa veba i distribuirati ga kao deo korpusa bez dozvole autora odgovarajuće veb stranice. Iako ovo deluje nerazumno, s obzirom da je veb stranica javno dostupna i svako može da je pročita, mnogi autori veb stranica na svojim stranicima izdaju prostor za oglase, a oglašivači plaćaju uslugu po broju posetilaca. Stoga ako samo jedna osoba poseti kopiju *umesto* [istakao autor, prim. prev.] originalne veb stranice, time se nanosi finansijska šteta autoru originalne veb stranice. Ako neko preuzme jednu kopiju originalne veb stranice na svoj računar za sopstvene potrebe (kopiranje stranice sa veba na lokalni računar se zaista i dešava svaki put kad korisnik poseti neku stranicu na vebu, bez obzira da li želi kopiju na svom računaru ili ne), tome se ne može prigovoriti. Ali

> ukoliko distribuira lokalnu kopiju, time krši zakon o autorskim pravima." ([McEnery & Hardie, 2012: 58])

I pre pojave veba je bilo teško utvrditi ko je sve nosilac prava na nekom autorskom delu (autor, autorova porodica, izdavač, prevodilac, itd.). Na vebu, kao i u novinskim člancima (štampanim ili elektronskim) potpisanim inicijalima, je još teže utvrditi ko je autor nekog članka. Kako ističu pojedini autori ([McEnery et al., 2006: 126]), u literaturi se zastupaju potpuno suprotna gledišta kada su u pitanju veb i autorska prava, od onih koji tvrde da najveći broj stranica na vebu nije ni zaštićen autorskim pravom i da se njihovi autori trude da dosegnu što više ljudi ([Spoor, 1996]67), do onih koji smatraju da je sav materijal na vebu zaštićen autorskim pravom na isti način kao i štampani dokumenti ([Cornish, 1999: 141]).

Pod ovakvim uslovima postaje nepraktično, ako ne i nemoguće obezbediti dozvolu od svih nosilaca autorskih prava na korpusnim tekstovima. Ono što dodatno komplikuje problem autorskih prava na vebu je mogućnost da tekst originalno potiče iz jedne zemlje, korpus u koji je taj tekst uključen se distribuira iz druge zemlje, a korisnik koji pristupa tom korpusu živi u trećoj zemlji, i svaka od tih zemalja ima svoj zakon o autorskim pravima. Tokom rasprave o autorskim pravima na elektronskoj dopisnoj listi Corpora-List, M. Dejvis (M. Davies) je izneo da advokati i profesori prava tvrde da je:

> ... bitan zakon o autorskim pravima zemlje iz koje se korpusni materijali distribuiraju, a NE zemlje u kojoj su kreirani originalni tekstovi, NITI zemlje iz koje krajnji korisnici koriste materijal. ([Davies, 2002])

S obzirom na eventualne teškoće prouzrokovane autorskim pravima na nekim korpusnim tekstovima, koje bi dovele do prinudnog uklanjanja tekstova iz okvira uzorkovanja/okvira uzorka, pojedini autori ([Kennedy, 1998: 78]) savetuju kreatorima korpusa da teškoće preduprede time što će za svaku od kategorija korpusnih tekstova koje njihov korpus obuhvata pribaviti više tekstova nego što su prvobitno planirali.

Moguće alternative procesu obezbeđivanja svih potrebnih dozvola od nosilaca autorskih prava na korpusnim tekstovima su ([McEnery & Hardie, 2012]59):

©1) uključiti u korpus samo one tekstove koji su javno dobro ili koriste licencu koja omogućava slobodno umnožavanje, konverziju i distribuiranje;

©2) kreirati korpus na osnovu tekstova sa veba, ne distribuirati sam korpus već listu adresa korpusnih tekstova na vebu;

©3) kreirati korpus, ne tražeći dozvolu od nosilaca autorskih prava na korpusnim tekstovima, a potom ne distribuirati korpus već omogućiti korisnicima ograničen pristup koji ne narušava zakon o autorskim pravima.

Alternativa ©1) značajno sužava skup potencijalnih izvora i umanjuje raznovrsnost tekstova za korpus što neminovno dovodi u pitanje balansiranost i reprezentativnost, posebno opštih korpusa[3].

Alternativa ©2) naizgled omogućava da svaki istraživač, na osnovu liste adresa korpusnih tekstova na vebu, preuzme te tekstove na svoj računar, napravi svoj korpus i reprodukcijom rezultata proveri tvrdnje istraživača koji je distribuirao korpus kao listu adresa na vebu. Problem je u tome što se veb neprekidno menja, pojavljuju se novi tekstovi, ali isto tako i „nestaju" stari tekstovi, bilo promenom njihove adrese na vebu, bilo gašenjem domena kome je pripadala njihova adresa ili povlačenjem samog teksta sa veba.

Alternativa ©3) je trenutno najzastupljenija, implementirana pomoću konkordancera IV generacije (v. odeljak 3.2, str. 187), a zasniva se na tzv. **„poštenoj upotrebi"** (eng. **fair use, fair dealing**[4]). „Poštena upotreba" predstavlja deo zakona o autorskim i srodnim pravima koji propisuje ograničenja i suspenzije isključivih autorskih prava, tj. određuje u kojim slučajevima se autorsko delo može umnožavati, reprodukovati i distribuirati bez dozvole nosioca autorskog prava. Zakonodavstva različitih država različito propisuju šta se podrazumeva pod „poštenom upotrebom". Sudovi u Sjedinjenim Američkim Državama pri odlučivanju razmatraju četiri fak-

[3]Ukoliko kreatori opšteg korpusa imaju za cilj da se korpus koristi za lingvističko istraživanje savremenog jezika, oni se ne mogu zadovoljiti prikupljanjem isključivo onih tekstova sa licencom koja omogućava slobodno umnožavanje, konverziju i distribuiranje teksta. Takvom odlukom bi, već na početku svog rada, morali da se odreknu većine savremenih književno-umetničkih tekstova, a rezultujući korpus ne bi mogao da se označi kao opšti.

[4]**Fair use** se koristi kao termin u zakonodavstvu Sjedinjenih Američkih Država, dok je **fair dealing** karakterističan za zakonodavstva zemalja Komonvelta. Kao što je navedeno u nastavku, postoje suptilne razlike u značenju ovih termina.

tora (svrhu i karakter upotrebe kopirane sadržine autorskog dela, prirodu autorskog dela, veličinu kopirane sadržine u odnosu na celo autorsko delo, efekat upotrebe na potencijalno tržište). Zakoni zemalja Komonvelta (Velika Britanija, Kanada, Australija, Novi Zeland, itd.) nabrajaju kategorije[5] na koje isključivo mogu da se primene ograničenja i suspenzije autorskih prava.

Kreatori korpusa kojima se pristupa preko konkordancera IV generacije, se pozivaju na „poštenu upotrebu" iz sledećih razloga:

- korpus se ne distribuira;
- korisnici preko čitača veba i veb-sučelja korpusa prosleđuju upit korpusu i kao rezultat dobijaju generisane konkordance ograničenog konteksta (v. odeljak 3.2).
- u praksi se konkordanca prikazana u čitaču veba svodi na deo rečenice, rečenicu ili par rečenica, što po kreatorima korpusa predstavlja „poštenu upotrebu" uzimajući u obzir da korisnici nemaju pristup celom tekstu, niti mogu na jednostavan način da rekonstruišu celokupan tekst na osnovu konkordanci.

Zaštita privatnosti

Pored zakona o autorskim i srodnim pravima, kreatori korpusa moraju da vode računa i o zakonu o zaštiti privatnosti iz sledećih razloga:

(ZP1) prilikom pretrage korpusa u tekstovima se mogu naći informacije o osobama i organizacijama koje zadiru u njihovu privatnost;

(ZP2) informacije pridružene tekstovima korpusa (metapodaci) poput identiteta osoba koje su izvori govornih tekstova, zvučni zapisi njihovog govora i transkripti tih zapisa takođe mogu narušiti privatnost, kako samih govornika, tako i osoba i organizacija o kojima govore;

[5]Kategorije variraju od jedne zemlje do druge, ali se uglavnom svode na istraživanje, prikaz, kritiku, satiru, parodiju, informisanje, sudski postupak, itd.

(ZP3) ukoliko se od korisnika korpusa očekuje da se registruje kako bi mogao da dobije korisničko ime i lozinku, tj. da pristupi korpusu preko sistema autorizacije, postoji potencijalna opasnost da se lični identifikacioni podaci koji se zahtevaju od korisnika prilikom registracije zloupotrebe.

Primer rigorozne primene zakona o zaštiti privatnosti u slučaju (ZP1) iskusili su slovenački korpusni lingvisti u julu 2012. godine ([Krek, 2012]). Tada je slovenački poverenik za informacije izdao obavezujuće rešenje kojim se sva lična imena u korpusu slovenačkog jezika „Nova beseda" moraju ili izuzeti iz rezultata pretrage ili zameniti bilo anonimnim referencama[6] bilo drugim imenima. Posle „pregovora" sa poverenikom za informacije, kreatori „Nove besede" su dobili dozvolu da korisnici koriste lična imena u pretrazi, ali da kombinacije imena i prezimena nisu dozvoljene, tj. rezultati pretrage u tom slučaju moraju biti prazni sa odgovarajućom napomenom korisnicima korpusa.

Slučaj (ZP2) se razrešava tako što se od ljudi čiji se govor koristi u korpusu traži pismeni pristanak za snimanje i transkripciju njihovog govora, pri čemu se transkripti i zvučni zapisi uključuju u korpus pod uslovom da govornici ostanu anonimni ili da kasnije odslušaju i eventualno obrišu snimke za koje ne žele da budu sastavni deo korpusa.

U slučaju (ZP3) se od korisnika takođe traži pristanak prilikom registracije za korišćenje ličnih identifikacionih podataka pri čemu se kreatori korpusa obavezuju da će te podatke koristiti isključivo za potrebe provere identiteta korisnika (na primer, u slučaju da korisnik zaboravi lozinku i zatraži od administratora korpusa da mu pošalje novu lozinku) i neće ih prosleđivati trećim licima (na primer, odeljenju za marketing neke firme, itd.).

Izvori elektronskog teksta

Izvori elektronskog teksta su višestruki:

[6] Jedna moguća realizacija anonimnih referenci je definisana standardom TEI (v. odeljak 2.4, str. 129) kao XML-element `<gap desc='name' reason='anonymization'/>`, gde naziv elementa `gap` ukazuje na prazninu u tekstu, atribut `desc` opisuje šta se originalno nalazilo u tekstu, u ovom slučaju ime (`name`), dok atribut `reason` ukazuje na uzrok praznine u tekstu, u konkretnom primeru je reč o zameni imena anonimnom referencom, tj. anonimizaciji (`anonymization`).

- Internet;
- elektronska izdanja monografskih publikacija i periodike na spoljašnjim memorijama (CD, DVD, BD);
- **dokumenti nastali kao digitalni** (eng. **digital-born documents**).

Internet Jedan od glavnih izvora elektronskog teksta danas je svakako Internet, posebno veb kao jedan od njegovih najpopularnijih servisa. Državne institucije, mediji, preduzeća, organizacije i pojedinci svakodnevno proizvode nove elektronske sadržaje na svojim prezentacijama bilo da su u pitanju zakoni, vesti, reklame, elektronske knjige, elektronski časopisi, oglasi, lične prezentacije, lični dnevnici (eng. blog), forumi, arhivi teksta, digitalne biblioteke, itd.

Ne samo što je veb neiscrpan izvor elektronskog teksta, već se u većini slučajeva lako preuzima tekstuelni sadržaj stranica na vebu, bilo snimanjem odgovarajuće datoteke na lokalni disk (u formatu čistog teksta ili u formatu HTML), bilo kopiranjem sadržaja stranice u lokalnu tekstuelnu datoteku pomoću nekog uređivača teksta (eng. editor). U preuzetom tekstu je ponekad neophodna intervencija kako bi se uklonili sadržaji poput reklama, oglasa i navigacionih menija.

Ručno preuzimanje i korekcija elektronskog teksta sa veba imaju smisla samo u slučaju kada broj preuzimanja i veličina preuzetog teksta nisu obimni, inače je neophodno automatizovati ceo proces. U tu svrhu već postoji dovoljno programa koji mogu preuzimati čitave prezentacije, odnosno skupove stranica (v. poglavlje 4, posebno odeljak 4.2).

Elektronska izdanja na spoljašnjim memorijskim medijumima Pojavom jeftinih spoljašnjih memorijskih medijuma velikog kapaciteta (CD, DVD, BD), omogućeno je pohranjivanje ne samo audio i video zapisa (na primer, muzičkih albuma i filmova), već i ogromnih količina teksta. Danas je uobičajeno da izdavači periodičnih publikacija u pravilnim vremenskim intervalima izdaju DVD sa elektronskim verzijama tekstova tih publikacija. Na taj način kompletni tekstovi novina i časopisa postaju dostupni, besplatno ili uz određenu nadoknadu. Na sličan se može doći i do kolekcija

književnih tekstova, rečnika, enciklopedija, itd. Pojedini korpusi, ili bar njihovi delovi, takođe su dostupni u ovom formatu (na primer, korpusi engleskog jezika poput COBUILD, BNC, itd.).

Digitalizacija teksta

Digitalizacija pisanog teksta I pored svakodnevne produkcije novih elektronskih tekstuelnih dokumenata, ogroman broj tekstova, nastao pre pojave Interneta, postoji samo u nedigitalnom obliku (rukopis, štampani tekst). Ukoliko postoji potreba da se neki od takvih tekstova nađe u korpusu, neophodno je izvršiti njegovu konverziju u format elektronskog teksta[7].

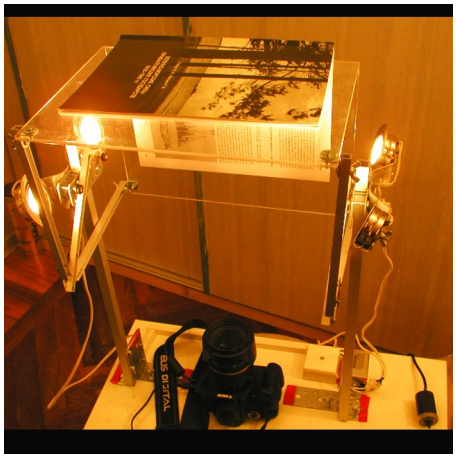
Postupak konverzije analognih (nedigitalnih) objekata u digitalni oblik naziva se **digitalizacija** (eng. **digitization**). Elektronski tekst je samo jedan od mnogobrojnih digitalnih formata, pa time i jedan od mogućih rezultata digitalizacije.

Transformacija nedigitalnog teksta u elektronski tekst može biti direktna i indirektna.

Direktna transformacija nedigitalnog teksta u elektronski tekst se realizuje prekucavanjem teksta uz pomoć tastature. Iako se na taj način neposredno proizvodi elektronski tekst, postupak zahteva ljudske, a često i finansijske[8] resurse, vremenski je zahtevan i podložan greškama. Pošto je prekucavanje skupa transformacija nedigitalnog teksta u elektronski tekst, primenjuje se u retkim slučajevima, obično kao krajnja alternativa (na primer, za potrebe digitalizacije drevnih rukopisa).

Indirektnom transformacijom nedigitalnog teksta se najpre proizvodi međuproizvod u digitalnom formatu koji nije elektronski tekst, a potom se taj međuproizvod softverski konvertuje u elektronski tekst. U praksi se indirektna transformacija nedigitalnog teksta uglavnom svodi na kombinovanje skaniranja i optičkog prepoznavanja karaktera. Skaniranjem se nedigitalni tekst najpre transformiše u neki od mnogobrojnih formata digitalnih slika (GIF, JPEG, PNG, TIFF, BMP, itd.), a potom se optičkim prepoznavanjem karaktera digitalna slika konvertuje u elektronski tekst. Skaniranje se vrši pomoću posebnih uređaja koji se nazivaju skaneri, mada se u istu

[7]Elektronski tekst je definisan u odeljku 2.2.
[8]Finansijski resursi se koriste kao nadoknada za rad ljudskih resursa.



Slika 2.1: Skaniranje pomoću digitalnog fotoaparata (preuzeto sa adrese `http://www.diybookscanner.org/`)

svrhu mogu koristiti i digitalni fotoaparati (uz prateću opremu u vidu reflektora, stativa, postolja, itd., Slika 2.1).

Konverziju slike dobijene skaniranjem u elektronski tekst obavlja posebni softver za **optičko prepoznavanje karaktera** (eng. **optical character recognition**, skr. **OCR**), koji se najčešće isporučuje zajedno sa skanerom.

Iako je indirektna transformacija u elektronski tekst značajno jeftinija od direktne transformacije, njen glavni nedostatak je daleko veći broj grešaka u odnosu na ručno prekucavanje teksta. Greške proizvodi softver za optičko prepoznavanje karaktera jer nije u stanju da dovoljno precizno razlikuje simbole ili grupe simbola koji imaju sličan oblik (npr. `1` i `l`, `m` i `ni`, odnosno `rn`, itd.). **Softver za proveru pisanja** (eng. **spelling checker**) i programi koji koriste (morfološke) elektronske rečnike (v. odeljak 4.3, str. 216) mogu da detektuju očigledne greške, ali u tekstu ne mogu lako otkriti greške koje se pojavljuju kao pravilno zapisani oblici leksema iz rečnika koji koristi softver (npr. `mace` umesto `mače`, `čini` umesto `čim`). Sem toga, proces indirektne transformacije iz nedigitalnog u elektronski tekst je relativno spor.

Ponekad kreatori korpusa ne mogu da biraju između dva navedena načina za transformaciju nedigitalnog u elektronski tekst. Naime, ako je neki tekst neophodan u korpusu, a skaniranje daje loše rezultate, na kraju je prekucavanje, iako daleko skuplje, jedino rešenje za dobijanje odgovarajućeg elektronskog teksta.

Bez obzira na sve predostrožnosti, pogotovo tokom prekucavanja, elektronski

tekst će na kraju sadržati greške (tipografske greške iz originalnog nedigitalnog teksta, nekorektno prekucan tekst sa eventualnim dvostruko unetim ili pak izostavljenim delovima teksta, greške pri prepoznavanju karaktera i sl.). Stoga se postavlja pitanje korekcije grešaka po završetku transformacije u elektronski tekst. Pre nego što se pristupi toj fazi, neophodno je da se utvrdi prag tolerancije, kako bi se ta faza izvela što efikasnije, i, ako je moguće, automatski.

Digitalizacija govornog teksta U slučaju materijala za korpus koji čine audio zapisi govornog jezika, potrebno je izvršiti njihovu konverziju u format pisanog teksta, tj. transkripciju govornog jezika. Ovaj proces se uglavnom izvodi ručno jer je razvoj softvera koji bi se time bavio još uvek u povoju[9]. Pre same transkripcije neophodno je ustanoviti pravila za zapis govornog jezika. Naime, sem verbalnog dela, transkripcija opisuje i dužinu pauza, tip intonacije, kao i paralingvističke fenomene poput smeha, uzdaha, itd.

Kad se obavi transkripcija i kao rezultat dobije pisani tekst, njegova konverzija u elektronski tekst se obavlja na već opisani način (v. str. 68).

U ovom radu će, pre svega, biti reči o kreiranju korpusa pisanih tekstova. Više o govornim korpusima se može naći, na primer, u [Wichmann, 2008].

Evidencija i klasifikacija tekstova

Tokom prikupljanja tekstova neophodno je voditi odgovarajuću evidenciju u kojoj se čuvaju sledeće informacije:

- kad su tekstovi prikupljeni (datum);
- koji je originalni izvor teksta (rukopis, štampana publikacija, dokument sa veba, elektronska knjiga izdata na spoljašnjem medijumu, itd.);
- podaci o originalnoj verziji teksta (na primer, u slučaju da se radi o štampanoj publikaciji, neophodni podaci su naslov, autor(i), izdavač, godina izdanja, prevodilac, itd.);

[9]Time se bavi disciplina u okviru računarske lingvistike/obrade prirodnog jezika — **prepoznavanje zvuka** (en. **speech recognition**).

- podaci o elektronskoj verziji teksta (da li je tekst nastao kao digitalan ili je digitalizovan; ako je digitalizovan, da li je prekucan ili je izvršeno skaniranje i optičko prepoznavanje karaktera; ko je obavio digitalizaciju; ko je kontrolisao ispravnost teksta, odnosno ispravljao eventualne greške; da li je tekst anotiran, na koji način, ko je obavio anotaciju, itd.);

Evidencija može da obuhvati i statističke podatke o tekstu, tj. broj tokena, korpusnih reči, tipova i korpusnih tipova (v. odeljak 2.3, str. 92).

U okviru evidencije je poželjno klasifikovati tekstove kako bi se tokom anotacije lakše pridružile odgovarajuće informacije sa ciljem da se korisniku omogući filtriranje rezultata pretrage. Prethodno je neophodno odrediti kriterijume klasifikacije, kao i kategorije tekstova definisane odgovarajućim kriterijumom. Kriterijumi klasifikacije mogu biti:

- funkcionalni stil teksta (književno-umetnički, razgovorni, novinski, administrativni, naučni);
- vrsta teksta (roman, pripovetka, intervju, članak, feljton, blog, itd.);
- da li je tekst napisan na jeziku korpusa ili je preveden na jezik korpusa, itd.

Kad god je moguće, treba iskoristiti postojeće standardne klasifikacije, poput Univerzalne decimalne klasifikacije (UDK, v. na primer [Bulajić, 2009]).

2.2 Predstavljanje elektronskog teksta u računaru

Karakterski skupovi, kodne sheme, glifovi. ASCII

Vizuelna reprezentacija teksta, papirnog ili elektronskog, ostavlja utisak da je i interna reprezentacija elektronskog teksta u računaru takođe dvodimenzionalna, sastavljena od linija (redova) i kolona u čijem preseku se nalaze pojedinačni simboli teksta.

U stvarnosti, elektronski tekst, kao i svi drugi podaci koji se čuvaju na računaru, na najnižem fizičkom nivou su predstavljeni (jednodimenzionalnim) nizom binarnih

nula i jedinica (bitova). Jednodimenzionalnost interne reprezentacije elektronskog teksta postiže se uvođenjem specijalnih oznaka za kraj linije (jedne ili više njih, u zavisnosti od operativnog sistema[10]). Na taj način se omogućava da se svi simboli teksta, uključujući razmake u okviru pojedinačne linije, kao i oznake za kraj linije interno tretiraju kao nizovi bitova.

Da bi korisnici mogli da razmenjuju elektronski tekst i da pred sobom imaju njegovu vizuelnu reprezentaciju istovetnu originalnoj verziji autora teksta, neophodno je da postoji standard za interno predstavljanje simbola elektronskog teksta. Nažalost, čak i kada su započeli prvi pokušaji standardizacije šezdesetih godina XX veka, pa sve do današnjih dana, uvek bi se nezavisno pojavilo bar dva „standarda", jedan iniciran organizacijama zaduženim za standardizaciju, dok su drugi predlog razvijale i forsirale moćne korporacije. Tako se **američki standardni kôd za razmenu informacija** (eng. **American Standard Code for Information Interchange**, skr. **ASCII**), predlog **Američkog nacionalnog instituta za standarde** (eng. **American National Standards Institute**, skr. **ANSI**), iako usvojen 1963. godine, punih 18 godina „borio" da bude prihvaćen umesto standarda EBCDIC koji je razvila korporacija IBM; tek kada je IBM 1981. godine počeo da proizvodi prve personalne računare sa operativnim sistemom PC-DOS, odnosno MS-DOS, ASCII je postao pravi standard za zapis elektronskog teksta. Komercijalni uspeh operativnog sistema MS-Windows, najpre kao grafičke nadgradnje sistema MS-DOS, a potom i kao novog operativnog sistema koji je preuzeo ASCII, doprineo je da svaki sledeći predlog standarda za predstavljanje elektronskog teksta u računaru bude samo proširenje u odnosu na ASCII.

Na primeru standarda ASCII može se ilustrovati zapis elektronskog teksta u računaru. ASCII podržava zapis ukupno 128 **apstraktnih karaktera**. Apstraktni karakter je jedinica informacije koja se koristi za organizaciju, kontrolu ili prezentaciju tekstuelnih podataka ([Stanojević, 2001]). Skup svih apstraktnih karaktera koje podržava određeni standard naziva se **repertoar**.

Vizuelna reprezentacija apstraktnog karaktera naziva se **slovni oblik** ili **glif**

[10]Operativni sistemi MS DOS i MS Windows koriste dve takve oznake, dok sve distribucije sistema Linux primenjuju jednu oznaku.

(eng. **glyph**). Treba naglasiti da apstraktni karakter i njegova vizuelna reprezentacija nisu isto. Jednom apstraktnom karakteru može odgovarati više različitih glifova (Tabela 2.1), ali i jednom glifu može odgovarati više karaktera[11].

Tabela 2.1: Primer apstraktnog karaktera i njegovih glifova

apstraktni karakter	**glifovi (slovni oblici)**
Veliko latinično slovo A	**A** *A* *A* **A** A A 𝔸 𝓐 A

Apstraktni karakteri se preslikavaju u određeni, uglavnom nenegativni podskup skupa celih brojeva. Preslikavanje repertoara u podskup skupa celih brojeva naziva se **karakterski skup** (eng. **character set**), a slika apstraktnog karaktera pri tom preslikavanju naziva se **kodna tačka** (eng. **code point**) ili **kodna pozicija** (eng. **code position**) Uređen par koga čine apstraktni karakter i njemu pridružena kodna pozicija naziva se **karakter** (eng. **character**). Karakterski skup i kodne pozicije se obično predstavljaju (kodnom) tabelom, tako da se dopisivanjem oznaka linije i kolone u kojoj se nalazi određeni simbol dobija njegova kodna pozicija kao broj u heksadekadnom sistemu. Na primer, uvidom u kodnu tabelu za ASCII (Slika 2.2) zaključuje se da je kodna pozicija velikog latiničnog slova A $41_{16} = 65$.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	NUL	STX	SOT	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Slika 2.2: Karakterski skup ASCII (preuzeto iz [Janičić & Marić, 2011])

Fizičku reprezentaciju kodne tačke (pozicije) na računaru predstavlja niz bitova koji se još naziva **kodirani karakter** (eng. **coded character**). Preslikavanje kodnih pozicija u odgovarajuće nizove bitova definiše jedno **kodiranje karaktera** (eng. **character encoding**), odnosno jednu **kodnu shemu** (eng. **character encoding scheme**), odnosno jedan **kôd**.

[11] Na primer, za latinično i ćirilično slovo A (isti glif) postoje posebni, različiti karakteri u okviru karakterskog skupa Unicode (v. str 80).

Ukoliko se svakoj kodnoj poziciji pridružuje niz bitova iste dužine, u pitanju je **kôd fiksne dužine**, u protivnom se radi o **kôdu promenljive dužine**.

Za karakterski skup ASCII se primenjuje istoimeni kôd fiksne dužine koji svakoj kodnoj poziciji dodeljuje niz od 7 bitova, zbog čega se još kaže da je ASCII sedmobitni (7-bitni) kôd (Tabela 2.2).

Tabela 2.2: ASCII kôd: razmak, cifre i slova

ASCII			
Karakter	Kodna pozicija		Kodirani karakter
	dekadna	heksadekadna	
␣	32	20_{16}	$010\vert0000_2$
0	48	30_{16}	$011\vert0000_2$
...	...	...	...
9	57	39_{16}	$011\vert1001_2$
A	65	41_{16}	$100\vert0001_2$
...	...	...	...
Z	90	$5A_{16}$	$101\vert1010_2$
a	97	61_{16}	$110\vert0001_2$
...	...	...	...
z	122	$7A_{16}$	$111\vert1010_2$

Prema tome, elektronski tekst je predstavljen u računaru kao niz karaktera, a svaki karakter — kao niz bitova. Svakom karakterskom skupu odgovaraju glifovi ili slovni oblici koji se grupišu u fontove. Na osnovu informacija o primenjenoj kodnoj shemi i izabranom fontu, program za uređivanje teksta internu reprezentaciju elektronskog teksta mapira u odgovarajuće slovne oblike koje prikazuje na ekranu.

Računarska reprezentacija teksta na srpskom jeziku

YUSCII Karakterski skup ASCII obuhvata 33 kontrolna karaktera (kodne pozicije 0–31 i 127) i 95 grafičkih karaktera (kodne pozicije 32–126). Grafičke karaktere čine razmak, velika i mala slova engleske abecede i znaci interpunkcije. Samim tim je ASCII prevashodno bio namenjen engleskom govornom području i nije mogao da se koristi za kodiranje teksta na većini ostalih jezika, uključujući i srpski.

Skoro deset godina pre no što će ASCII zvanično zavladati u računarskoj industriji, 1972. godine se pojavilo prvo izdanje standarda ISO/IEC 646 IRV[12]. Osnovna

[12]IRV je skraćenica od *International Reference Version* (srp. *Međunarodna referentna verzija*).

ideja iza ISO/IEC 646 IRV je kreiranje 7-bitnog međunarodnog koda za zapis teksta koji bi na isti način kodirao zajedničke karaktere različitih nacionalnih pisama, pri čemu bi pojedine retko korišćene kodne pozicije u različitim zemljama mogle da se koriste za kodiranje različitih karaktera, tj. karaktera specifičnih za odgovarajuća nacionalna pisma. Na taj način bi se omogućilo formiranje različitih nacionalnih varijanti standarda ISO/IEC 646 IRV, pri čemu bi se „engleska varijanta" svela na već postojeći kôd ASCII.

Tako je tokom osamdesetih godina XX veka Jugoslovenski zavod za standarde definisao dve nacionalne varijante standarda ISO/IEC 646 IRV: latiničnu JUS I.B1.002 (za srpsko-hrvatski i slovenački jezik) i ćiriličnu JUS I.B1.003 (za srpski i makedonski jezik). Ova dva standarda su poznatija pod neformalnim imenima YU-ASCII, odnosno YUSCII. Obe verzija standarda YUSCII iskoristile su mogućnost da kodne pozicije 64, 91–94, 96 i 123-126 dodele karakterima specifičnim za srpski jezik, dok je ćirilični YUSCII iskoristio i kodne pozicije karaktera 81, 87, 88, 113, 119 i 120 (Tabela 2.3).

Korišćenjem standarda YUSCII izgubljena je jednoznačnost kodiranja teksta. Različiti karakteri kodirani su na isti način, a jedino je informacija o fontu omogućavala različitu vizuelnu reprezentaciju. Međutim, kako YUSCII-fontovi nisu bili sastavni deo operativnog sistema, to su korisnici bili prinuđeni da ih sami izrađuju. Tako se pojavilo više rešenja, uglavnom nekvalitetnih, a u slučaju pripreme za štampu i neprikladnih, koji su dodatno ugrozili prenosivost teksta. Naime, preuzimanjem YUSCII-teksta na računar na kome nije bilo odgovarajućeg YUSCII-fonta bila bi izgubljena jedina preostala informacija o kodiranju teksta, a slova specifična za srpski jezik bi se u vizuelnoj reprezentaciji pretvorila u ASCII-karaktere sa odgovarajućih kodnih pozicija (@, [, \,], ^, ‘, {, |, }, ~), popularno nazvane „kuke i kvake". S obzirom da su pomenuti karakteri, iako „žrtvovani" kao „retko korišćeni", nezamenljivi za mnoge korisnike, posebno programere, to je u bivšoj Jugoslaviji, s jedne strane, izazvalo pojavu novih kodova, pri čemu su i „uticajniji računarski časopisi... kao što su ‚Računari' ili ‚Moj Mikro' razvili sopstvene osmobitne, međusobno nekompatibilne kodove" ([Stanojević, 2001: 8]). S druge strane, kao reakcija na proizvedenu konfuziju, mnogi nisu koristili nijedan od prilagođenih

Tabela 2.3: Kôd YUSCII

YUSCII				
ASCII karakter	YUSCII karakter (latinica/ćirilica)	Kodna pozicija		Kodirani karakter
		dekadna	heksadekadna	
␣	␣	32	20_{16}	$010\|0000_2$
0	0	48	30_{16}	$011\|0000_2$
...	...	...	...	...
9	9	57	39_{16}	$011\|1001_2$
...	...	...	...	...
@	Ž/Ж	64	40_{16}	$100\|0000_2$
A	A/А	65	41_{16}	$100\|0001_2$
...	...	...	...	...
Q	Q/Љ	81	51_{16}	$101\|0001_2$
...	...	...	...	...
W	W/Њ	87	57_{16}	$101\|0111_2$
X	X/Џ	88	58_{16}	$101\|1000_2$
...	...	...	...	...
Z	Z/З	90	$5A_{16}$	$101\|1010_2$
[	Š/Ш	91	$5B_{16}$	$101\|1011_2$
\	Đ/Ђ	92	$5C_{16}$	$101\|1100_2$
]	Ć/Ћ	93	$5D_{16}$	$101\|1101_2$
^	Č/Ч	94	$5E_{16}$	$101\|1110_2$
...	...	...	...	...
‘	ž/ж	96	60_{16}	$110\|0000_2$
a	a/а	97	61_{16}	$110\|0001_2$
...	...	...	...	...
q	q/љ	113	71_{16}	$111\|0001_2$
...	...	...	...	...
w	w/њ	119	77_{16}	$111\|0111_2$
x	x/џ	120	78_{16}	$111\|1000_2$
...	...	...	...	...
z	z/з	122	$7A_{16}$	$111\|1010_2$
{	š/ш	123	$7B_{16}$	$111\|1011_2$
|	đ/ђ	124	$7C_{16}$	$111\|1100_2$
}	ć/ћ	125	$7D_{16}$	$111\|1101_2$
~	č/ч	126	$7E_{16}$	$111\|1110_2$

kodova, već su se opredelili za ASCII.

Osmobitni kodovi (ISO-8859 i Microsoft) Tokom implementacije koda ASCII, odnosno nacionalnih varijanti standarda ISO/IEC 646 IRV, na personalnim računarima, uočena je mogućnost proširivanja odgovarajućeg karakterskog skupa. Naime, količina podataka koja je mogla istovremeno da se prenese ili obradi u okviru procesora prvih personalnih računara (procesorska reč) je predstavljala niz od osam bitova. Samim tim, i operativna memorija personalnih računara je organizovana kao niz osmorki bitova, odnosno kao niz **okteta** (eng. **octet**) ili **bajtova** (eng. **byte**)[13]. Ako se u jednom bajtu čuva jedan ASCII-karakter, kodiran odgovarajućom 7-bitnom kodnom shemom, jedan bit ostaje neiskorišćen. Međutim, ukoliko se kodna pozicija karaktera predstavlja nizom od osam bitova, tj. ako se iskoriste svi bitovi jednog bajta, broj karaktera koji može da se kodira na takav način je $2^8 = 256$, odnosno dvostruko veći od broja karaktera koje kodira kodna shema ASCII.

Međutim, kao u slučaju karakterskih skupova i kodnih shema ASCII i EBCDIC, ponovo je došlo do dva odvojena pokušaja standardizacije zapisa teksta u računaru zasnovanih na osmobitnim kodovima, s tom razlikom što se nasuprot Međunarodnoj organizaciji za standardizaciji (ISO) pojavio konzorcijum Microsoft. Suštinski gledano, oba pokušaja su zasnovana na istim idejama. Da bi se sačuvala kompatibilnost

[13]Ovde je potrebno naglasiti da pojmovi *oktet* i *bajt* nisu u potpunosti sinonimi. Naime, *oktet* je tehnički termin koji uvek predstavlja niz od osam bitova. S druge strane, *bajt* je najmanji adresibilni niz bitova u određenoj arhitekuri računara i tokom istorije računarstva njegova dužina se menjala. Tako su do sredine šezdesetih godina XX veka korišćeni računari kod kojih je dužina bajta bila šest bitova. Dužina bajta je usko vezana za dužinu niza bitova kojim se kodirao jedan karakter, kao što je i dužina procesorske reči uglavnom umnožak dužine jednog bajta, s obzirom da je prirodno da se u okviru procesora istovremeno mogu preneti ili obraditi svi bitovi kojima je kodiran jedan karakter. Tako je u 6-bitnoj arhitekturi uobičajena dužina procesorske reči bila 36 (kao umnožak broja šest), a karakter je kodiran pomoću šest bitova, te je ukupno moglo da se predstavi $2^6 = 64$ različita karaktera, i to neki znaci interpunkcije (najviše 28), dekadne cifre (10 karaktera), slova engleskog alfabeta bez mogućnosti pravljenja razlike između velikih i malih slova (26 karaktera), ponekad i neki kontrolni karakteri umesto znakova interpunkcije. Počev od centrale (eng. mainframe) IBM 360 koja je imala dužinu bajta od osam bitova, a pogotovo pojavom personalnih računara IBM-PC čiji su mikroprocesori koristili procesorsku reč iste dužine (osam bitova), broj osam *de facto* postaje podrazumevana dužina bajta, iako ni dan danas ne postoji nikakav standard koji to propisuje. U tehničkoj dokumentaciji se i danas strogo vodi računa o razlikovanju pojmova *oktet* i *bajt*, iz razloga što možda još postoje zastareli, prevaziđeni sistemi koji koriste dužinu bajta koja nije jednaka osam. Međutim, s obzirom da se poslednjih trideset godina dužina bajta faktički poistovetila sa brojem osam, a značenje pojma *bajt* sa značenjem pojma *oktet*, u nastavku će oba pojma biti korišćena u istom značenju, pogotovo što je uobičajeno da se pojam *bajt* češće koristi u tom značenju nego pojam *oktet*.

sa ranijim zapisom teksta u računaru, odlučeno je da se kodne pozicije 0–127 bez ikakvih izmena i dalje koriste za ASCII-karaktere (*donja kodna strana* karakterskog skupa), a da se preostale pozicije 128-255 iskoriste za predstavljanje novih karaktera (*gornja kodna strana* karakterskog skupa). S obzirom da je odmah na početku bilo jasno da ni 256 kodnih pozicija neće biti dovoljno za predstavljanje svih simbola koji se koriste u raznim jezicima i pismima, i ISO i Microsoft su odlučili da umesto jednog koda kreiraju, svako za sebe, porodicu kodova.

ISO je stvorio porodicu kodova poznatu pod nazivom ISO 8859 i ona je opisana istoimenim standardima ISO/IEC 8859 čija su prva izdanja objavljena 1987. godine, a potom održavana i dopunjavana sve do 2004. godine. U planu je bila izrada 16 kodova (ISO/IEC 8859-1, ISO/IEC 8859-2, ..., ISO/IEC 8859-16), ali se od ISO-/IEC 8859-12 zvanično odustalo 1997. godine. Od svih kodova porodice ISO 8859 najviše je korišćen ISO 8859-1 ili Latin-1 koji obuhvata simbole većine zapadnoevropskih jezika i pisama, dok su za zapis tekstova na srpskom jeziku najbitniji ISO 8859-2 ili Latin-2 (zbog velikih i malih latiničnih slova Š, Ž, Ć, Č i Đ, Slika 2.3) i ISO 8859-5 ili Cyrillic (zbog slova ćiriličnog pisma, Slika 2.4). Standard ISO 8859 rezerviše prvih 32 karaktera gornje kodne strane za kontrolne karaktere, tako da je moguće predstaviti svega još 96 novih karaktera, odnosno ukupno 191 karakter.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
8																
9																
A	NBSP	Ą	˘	Ł	¤	Ľ	Ś	§	¨	Š	Ş	Ť	Ź	SHY	Ž	Ż
B	°	ą	˛	ł	´	ľ	ś	ˇ	¸	š	ş	ť	ź	˝	ž	ż
C	Ŕ	Á	Â	Ă	Ä	Ĺ	Ć	Ç	Č	É	Ę	Ë	Ě	Í	Î	Ď
D	Đ	Ń	Ň	Ó	Ô	Ő	Ö	×	Ř	Ů	Ú	Ű	Ü	Ý	Ţ	ß
E	ŕ	á	â	ă	ä	ĺ	ć	ç	č	é	ę	ë	ě	í	î	ď
F	đ	ń	ň	ó	ô	ő	ö	÷	ř	ů	ú	ű	ü	ý	ţ	

Slika 2.3: Kodna shema ISO 8859-2 (preuzeto iz [Janičić & Marić, 2011])

Microsoft je najpre razvio jednu porodicu kodova namenjenu operativnom sistemu MS-DOS (tzv. OEM kodne strane), a potom i drugu (ANSI ili Windows-kodne strane) za operativni sistem Windows. Windows-kodne strane se obično označavaju prefiksom *CP* (skr. od engleskog *code page*) ili prefiksom *windows* i brojem od 1250 do 1258 (CP-1250, ..., CP-1258, odnosno windows-1250, ..., windows-1258). Za

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8																
9																
A	NBSP	Ё	Ђ	Ґ	Є	S	I	Ї	J	Љ	Њ	Ћ	Ќ	SHY	Ў	Џ
B	A	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
C	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
D	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F	№	ё	ђ	ґ	є	s	i	ї	j	љ	њ	ћ	ќ	§	ў	

Slika 2.4: Kodna shema ISO 8859-5 (preuzeto iz [Janičić & Marić, 2011])

razliku od standarda ISO 8859, Microsoft koristi i neke rezervisane kodne pozicije gornjih kodnih strana. Pri tom su neke od tih gornjih kodnih strana nadskupovi odgovarajućih gornjih kodnih strana ISO-karakterskog skupa (npr. CP-1252 u odnosu prema ISO 8859-1), dok su ostale delimično ili potpuno izmenjene u odnosu na odgovarajuće ISO/IEC standarde (poput CP-1250 u odnosu prema ISO 8859-2, odnosno CP-1251 u odnosu prema ISO 8859-5). Za kodiranje teksta na srpskom jeziku bitne su kodne sheme CP-1250 (latinično pismo, Slika 2.5) i CP-1251 (ćirilično pismo, Slika 2.6), dok je CP-1252 nadskup od ISO 8859-1, a takođe i podrazumevani kôd za englesku verziju operativnog sistema Windows.[14]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	€		‚		„	…	†	‡		‰	Š	‹	Ś	Ť	Ž	Ź
9		‘	’	“	”	•	–	—		™	š	›	ś	ť	ž	ź
A		ˇ	˘	Ł	¤	Ą	¦	§	¨	©	Ş	«	¬		®	Ż
B	°	±	˛	ł	´	µ	¶	·	¸	ą	ş	»	Ľ	˝	ľ	ż
C	Ŕ	Á	Â	Ă	Ä	Ĺ	Ć	Ç	Č	É	Ę	Ë	Ě	Í	Î	Ď
D	Đ	Ń	Ň	Ó	Ô	Ő	Ö	×	Ř	Ů	Ú	Ű	Ü	Ý	Ţ	ß
E	ŕ	á	â	ă	ä	ĺ	ć	ç	č	é	ę	ë	ě	í	î	ď
F	đ	ń	ň	ó	ô	ő	ö	÷	ř	ů	ú	ű	ü	ý	ţ	

Slika 2.5: Kodna shema Microsoft Windows CP 1250 (preuzeto iz [Janičić & Marić, 2011])

Ni porodice kodova ISO 8859 i Windows nisu rešile problem predstavljanja karaktera različitih karakterskih skupova u istoj datoteci, jer svaka datoteka može da

[14] U opcijama programa za Microsoft Windows kodna strana CP-1250 se često označava sa `Central European (Windows)`, kodna strana CP-1251 sa `Cyrillic (Windows)`, a kodna strana CP-1252 sa `Western European (Windows)`.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ђ		‚		„	…	†	‡		‰	Љ	‹	Њ	Ќ	Ћ	Џ
9		‘	’	“	”	•	–	—		™	љ	›	њ	ќ	ћ	џ
A		Ў	ў	Ј	¤	Ґ	¦	§	Ё	©	Є	«	¬		®	Ї
B	°	±	І	і	ґ	µ	¶	·	ё	№	є	»	ј	Ѕ	ѕ	ї
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	

Slika 2.6: Kodna shema Microsoft Windows CP 1251 (preuzeto iz [Janičić & Marić, 2011])

koristi tačno jednu kodnu shemu. Time je ponovo ograničen broj karaktera koji se mogu koristiti u jednoj datoteci, ali sada na 256 karaktera.

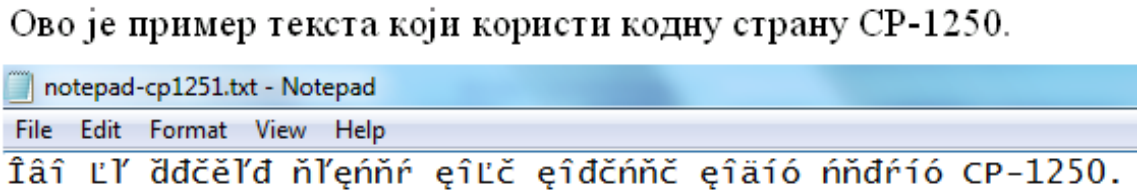
Drugi problem sa porodicama osmobitnih kodova je što za većinu tekstuelnih formata ne postoji mogućnost da se u datoteci čuva i informacija o primenjenoj kodnoj shemi. Izuzetak od ovog pravila su formati HTML i XML koji to rešavaju uvođenjem posebnog atributa čija je vrednost naziv kodne sheme (Tabela 2.4).

Tabela 2.4: Primeri čuvanja informacije o primenjenoj kodnoj shemi u pojedinim formatima teksta

Format	**Informacija o primenjenoj kodnoj shemi**
HTML	`<meta http-equiv="content-type"` `content="text/html; charset=windows-1250">`
XML	`<?xml version="1.0" encoding="iso-8859-2"?>`

Ukoliko tekstuelna datoteka kodirana nekim osmobitnim kodom nema informaciju o primenjenom kodu, programi koji prikazuju sadržaj tih datoteka mogu pogrešno interpretirati njihov sadržaj. Tako program Notepad za svaku datoteku kodiranu osmobitnim kodom pretpostavlja da je u pitanju kodna strana CP-1252 (Slika 2.7), a dijalog za čuvanje (snimanje) sadržine datoteke među ponuđenim kodnim shemama nudi samo jednu osmobitnu (opcija ANSI) koja je zapravo kodna strana CP-1252.

Unicode Da bi se prevazišla ograničenja osmobitnih kodova koja su onemogućavala razmenu teksta na globalnom nivou — pre svega ograničenost karakterskih skupova i nemogućnost da se zbog preklapanja kodnih pozicija istovremeno koriste karakteri iz



Slika 2.7: Pravilno prikazan tekst kodiran kodnom shemom CP-1251 i prikaz istog teksta u programu Notepad

različitih karakterskih skupova — krajem osamdesetih godina XX veka ponovo je paralelno započela izrada dva različita standarda za predstavljanje teksta u računaru. Izradu prvog, industrijskog standarda su započele firme XerocParc i Apple, da bi im se postepeno pridružile korporacije IBM, Sun Microsystems, Adobe, Microsoft i druge, čime je 1991. godine stvoren moćan neprofitni konzorcijum Unicode. S druge strane, Međunarodna organizacija za standardizaciju (ISO) je započela rad na izradi standarda ISO/IEC 10646, odnosno **Univerzalnog skupa karaktera kodiranih grupom okteta** (eng. **Universal Multiple-Octet Coded Character Set**, skr. **UCS**). Tako je izgledalo da će se ISO i vodeći predstavnici računarske industrije ponovo razići u pokušajima da ostvare isti cilj, međutim, na kraju je ipak postignut sporazum, te su verzije standarda Unicode (srp. Unikod) i ISO/IEC 10646 počev od druge verzije pa do danas konstantno usklađene.

Standard UCS je prvobitno trebalo da ostvari sledeće ciljeve ([Stanojević, 2001: 10–18]):

(U1) Kodira se čist tekst (bez formatiranja);

(U2) Kodiraju se karakteri, a ne glifovi (slovni oblici). Karakteri imaju unapred definisano značenje;

(U3) Repertoar koji se kodira treba da obuhvati sve znakove koji se koriste u razmeni teksta, uključujući one iz postojećih međunarodnih, nacionalnih i industrijskih standarda;

(U4) Kodiraju se pisma, a ne jezici;

(U5) Kodirani karakter je niz od 16 bitova, čime se omogućava predstavljanje $2^{16} = 65.536$ različitih karaktera;

(U6) Kod je potpun, tj. za kodiranje se koriste sve kodne pozicije;

(U7) Zagarantovana je konvertibilnost, tj. tačna konverzija između novog standarda i postojećih standarda kodiranja;

Obično se naglašava da je Unicode dobio ime po tri „uni", tj. kao *univerzalan* (eng. *universal*), *jednoznačan* (eng. *unique*) i *uniforman* (eng. *uniform*) kôd što izražavaju svojstva (U3)–(U5). Kodna shema koja je pokušala da implementira sva navedena svojstva nazvana je UCS-2, gde broj 2 predstavlja broj bajtova, preciznije okteta, koji se koriste za kodiranje jednog Unicode-karaktera. Međutim, da bi karakterski skup Unicode zaista bio univerzalan, tj. obuhvatio sve neophodne karaktere, 65.536 kodnih pozicija nije dovoljno. Zato je UCS-2 morao biti napušten.

Unicode-kodni prostor, tj. skup svih kodnih pozicija koje se mogu dodeliti Unicode-karakterima, podeljen je na 17 skupova iste veličine (65.536 kodnih pozicija) koji se nazivaju **ravni** (eng. **plane**). Ravni su numerisane celim brojevima od 0 do 16. Trenutno je na raspolaganju ukupno $17 \cdot 65.536 = 1.114.112$ kodnih pozicija za Unicode-karaktere. Kodne pozicije 0–65.535 formiraju tzv. nultu ili **osnovnu višejezičnu ravan** (eng. **Basic Multingual Plane**, skr. **BMP**) i one su dodeljene najčešće korišćenim karakterima. Prvih 256 karaktera u okviru BMP se poklapa sa elementima karakterskog skupa ISO 8859-1, pa su prvih 128 karaktera u BMP istovetni kao u karakterskom skupu ASCII. Kodne pozicije ravni 3–13 su nedodeljene, ravni 15 i 16 su rezervisane za privatnu upotrebu od strane subjekata izvan ISO i konzorcijuma Unicode, ravan 14 se koristi za negrafičke karaktere, dok se ravni 1 i 2 koriste za dodatne grafičke karaktere kojih nema u BMP.

Da bi se povećao broj Unicode-karaktera koji se mogu kodirati, bilo je neophodno ili odustati od uniformnosti koda (tj. fiksnog broja bitova kojim se predstavlja kodirani karakter), ili insistirati na uniformnosti koda, pri čemu je neminovno povećati broj bitova kodiranog karaktera. U prvom slučaju nastao je kôd promenljive dužine, UTF-16[15], a u drugom slučaju kôd fiksne dužine, UCS-4.

[15]UTF je skraćenica za *format za transformaciju Unikoda* (eng. *Unicode Transformation For-*

UTF-16 kodira karaktere kojima odgovaraju pozicije iz osnovne višejezične ravni (BMP) pomoću dva okteta, dok se karakteri ostalih ravni kodiraju pomoću četiri okteta, tj. pomoću dve šesnaestobitne vrednosti koje se nazivaju surogat-parovi.

UCS-4 kodira sve karaktere pomoću četiri okteta, odnosno nizom fiksne dužine od 32 bita, zbog čega se još naziva i UTF-32. UCS-4 je efikasniji od UTF-16 kada je u pitanje vreme pristupa kodnoj poziciji jer UCS-4 omogućava direktan pristup. Međutim, kada je iskorišćeni prostor u pitanju, četiri bajta po karakteru je izuzetno neefikasno rešenje, pogotovo što se karakteri van BMP retko koriste, a za karaktere iz BMP se dvostruko više troši prostor u odnosu na UTF-16.

Udvostručenje veličine datoteka i nemogućnost ASCII-orijentisanih ili bajt-orijentisanih programa da obrađuju tekstove kodirane shemama UCS-2, UTF-16 i UCS-4, dovela je do toga da su korisnici nekoliko godina ignorisali Unicode, pogotovo što su hteli da izbegnu i konverziju nezanemarljive količine postojećih dokumenata kodiranih osmobitnim kodnim shemama. To je dovelo do kreiranja još jedne kodne sheme, promenljive dužine, koja je nazvana UTF-8. Osnovna ideja iza UTF-8 je da se ASCII-karakteri kodiraju pomoću jednog bajta na isti način kako to rade osmobitni kodovi, a da se za ostale karaktere upotrebe dva, tri ili četiri bajta. Kodna shema UTF-8 se pokazala izuzetno ekonomičnom sa mogućnošću efikasne konverzije u UTF-16 i obrnuto, tako da se danas najviše koristi u odnosu na ostale Unicode-kodne sheme.

Nažalost, sve navedene sheme (UCS-2, UTF-16, UTF-32, UTF-8) se pojavljuju u dve varijante koje se međusobno razlikuju u redosledu bajtova kojima se kodira jedan karakter. Naime, procesor prilikom obrade 16-bitnog niza može, u zavisnosti od arhitekture, da obrađuje najpre viši bajt (eng. big-endian, skr. BE) pa niži, ili pak najpre najpre niži bajt (eng. little-endian, skr. LE) pa viši. Da Unicode ne bi favorizovao nijednu od pomenutih arhitektura, pogotovo što su obe više nego zastupljene, za svaku od pomenutih kodnih shema postoje obe varijante: UTF-16 LE, UTF-16 BE, itd. Da bi se ove varijante međusobno razlikovale, na početku čistog teksta se ubacuje specijalna sekvenca dva bajta koja se zove **oznaka redosleda bajtova** (eng. **Byte Order Mark**, skr. **BOM**). Heksadekadne vrednosti bajtova

mat).

BOM-sekvence su FF, odnosno FE, pri čemu se u slučaju BE-kodiranja koristi BOM-sekvenca FFFE, a u slučaju LE-kodiranja — BOM-sekvenca FEFF.

Karakteri specifični za srpsko latinično i ćirilično pismo se nalaze u osnovnoj višejezičnoj ravni (Tabela 2.5 i Slika 2.8).

Tabela 2.5: Dekadne i heksadekadne Unicode-kodne pozicije specifičnih karaktera srpske latinice

Karakter	**Kodna pozicija**	**Karakter**	**Kodna pozicija**
Ć	$262 = 0106_{16}$	**ć**	$263 = 0107_{16}$
Č	$268 = 010C_{16}$	**č**	$269 = 010D_{16}$
Đ	$272 = 0110_{16}$	**đ**	$273 = 0111_{16}$
Š	$352 = 0160_{16}$	**š**	$353 = 0161_{16}$
Ž	$381 = 017D_{16}$	**ž**	$382 = 017E_{16}$

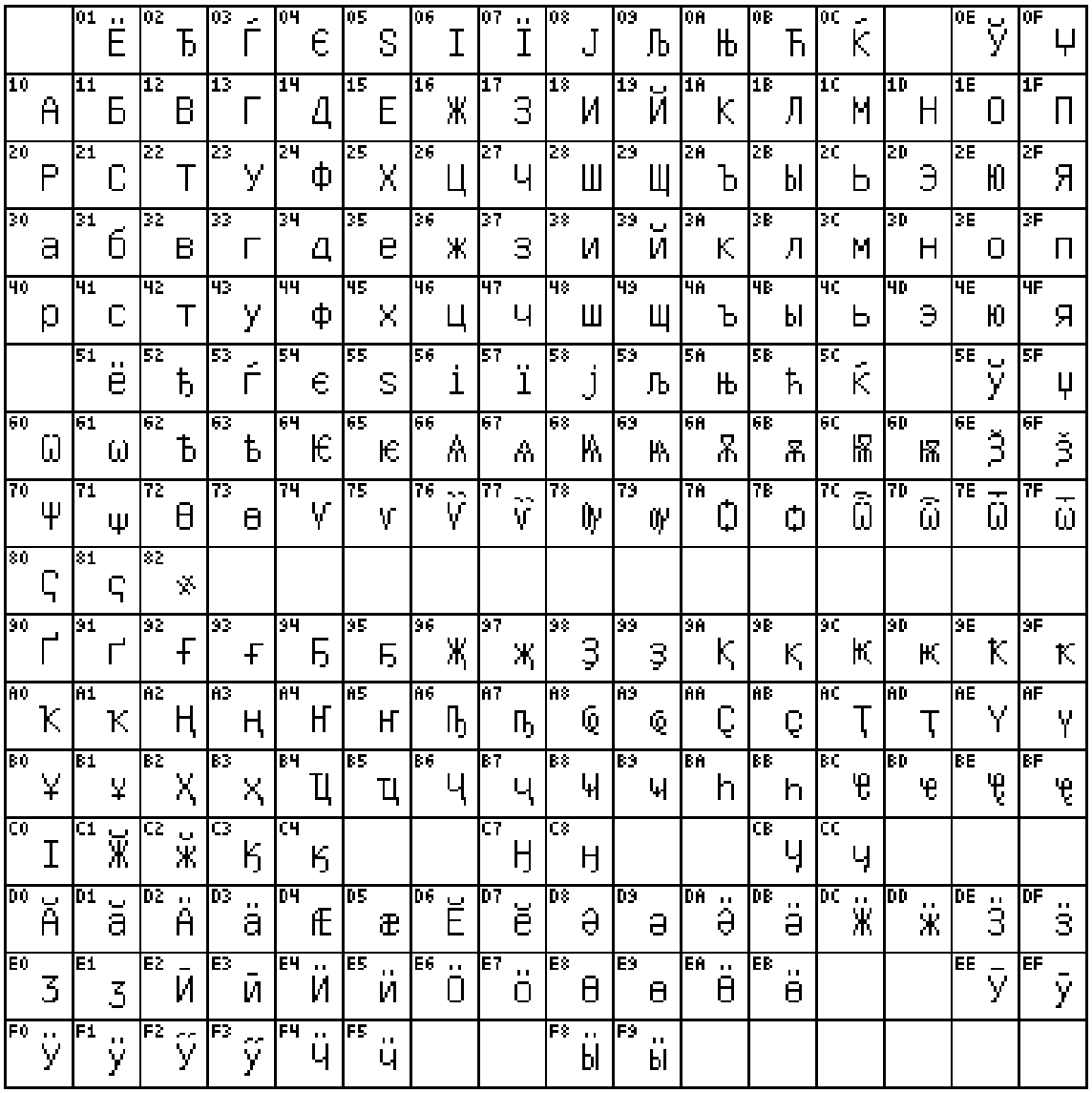
Počev od verzije Windows 2000, Microsoft tretira Unicode kao sastavni deo svog operativnog sistema. To je značajno uticalo da u narednim godinama i ostali proizvođači softvera, pogotovo oni koji su i sami članovi konzorcijuma Unicode, omoguće u svojim programskim paketima i podršku za Unicode.

Zapis teksta na srpskom jeziku u savremenim računarima Na osnovu izloženog u ovom odeljku, zaključuje se da se na savremenim računarima tekstovi na srpskom jeziku mogu kodirati (Slika 2.9):

- prevaziđenom latiničnom ili ćiriličnom verzijom 7-bitne kodne sheme YUSCII;
- 8-bitnim kodnim shemama ISO 8859-2 ili CP-1250 (latinica), odnosno ISO 8859-5 ili CP-1251 (ćirilica);
- proizvoljnom kodnom shemom koja kodira repertoar Unikoda (UTF-16 LE, UTF-16 BE, UTF-8, ...).

Od navedenih rešenja, YUSCII se definitivno ne preporučuje, ali se nažalost i dalje koristi, tako da se na internetu još uvek može naći popriličan broj zvaničnih dokumenata, prezentacija, pa čak i knjiga izdatih posle 2000. godine koje koriste neki od zaostalih YUSCII-fontova.

Osmobitni kodovi dopuštaju predstavljanje ili samo ćiriličnog ili samo latiničnog teksta, dok Unicode omogućava kodiranje teksta u kome se pojavljuju oba pisma.



Slika 2.8: Unicode-karakteri ćiriličnog pisma (kodna pozicija je zapisana heksadekadno, i to samo niži bajt, viši bajt je uvek 04.)

Zbog toga sve više dokumenata koristi Unicode-kodne sheme, posebno UTF-8 kao najekonomičnije rešenje.

2.3 Obrada elektronskog teksta

Elektronski tekst, sa stanovišta računara, je zapravo niz karaktera. Mogućnost bilo kakve automatske obrade elektronskog teksta pomoću računara, koja vodi računa o jeziku na kome je tekst napisan, zahteva da se u tekstu prethodno identifikuju lingvističke jedinice kao što su:

(l1) reči, brojevi, interpunkcija;

(l2) sintagme;

(l3) klauze, rečenice;

(l4) pasusi;

(l5) delovi diskursa, itd.

Proces identifikacije svih navedenih lingvističkih jedinica u tekstu se naziva **segmentacija teksta**. U zavisnosti od vrste lingvističkih jedinica (l1)–(l5) koje su predmet identifikacije, razlikuju se sledeći tipovi segmentacije teksta:

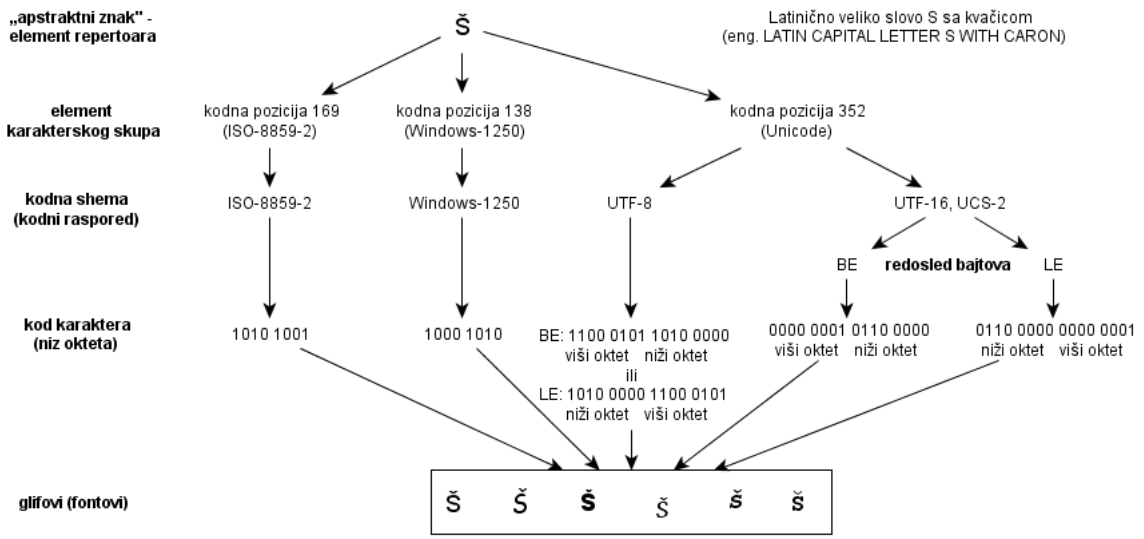
(s1) **tokenizacija** (eng. **tokenization**) ili **segmentacija na reči** (eng. **word segmentation**);

(s2) **segmentacija na rečenice** (eng. **sentence breaking**) ili **identifikacija kraja rečenice** (eng. **sentence boundary disambiguation**, skr. **SBD**);

(s3) **plitko parsiranje** (eng. **shallow parsing, chunking**);

(s4) **sintaksička analiza** ili **parsiranje** (eng. **parsing**).

Da bi uopšte moglo da se raspravlja o ovim procesima, neophodno je preciznije definisati lingvističke jedinice kao što su `reč`, odnosno `rečenica`. U računarstvu se, u tu svrhu, obično koriste osnovni pojmovi **teorije formalnih jezika**.



Slika 2.9: Predstavljanje karaktera „latinično veliko slovo S sa kvačicom" u različitim kodnim shemama

Osnovni pojmovi teorije formalnih jezika

Teorija formalnih jezika (skr. TFJ) je disciplina primenjene matematike sa primenama u računarstvu, lingvistici, formalnoj semantici, matematičkoj logici i drugim oblastima.

Ovde navodimo definicije pojmova *azbuka*, *niska*, *(formalni) jezik*, *regularni izraz*, preuzete iz računarstva, preciznije, iz teorije kompilacije ([Vitas, 2006]).

Definicija 2.1. **(Formalna) azbuka (alfabet)** je proizvoljan konačan skup simbola.

Definicija 2.2. Označimo sa Σ proizvoljnu azbuku. **Niska nad azbukom** Σ (eng. **string**) je konačan niz simbola azbuke Σ. Preciznije, ako je $n \geq 0$, a $x_1, x_2, \ldots, x_n$ simboli azbuke Σ, tada je $x = x_1 x_2 \cdots x_n$ niska nad azbukom Σ. Broj simbola niske x predstavlja **dužinu** niske x i označava se sa $|x|$, tj. $|x| = n$.

Definicija 2.3. Niska dužine nula se naziva **prazna niska** i označava se sa ε, tj. $|\varepsilon| = 0$.

Umesto višeznačnog termina `reč`, radije ćemo koristiti `niska`. Naime, niska je uopštenje reči (u lingvističkom smislu) jer može da sadrži sve simbole (formalne) azbuke i ne mora predstavljati reč prirodnog jezika sa određenim značenjem.

Primer 2.1. Neka je A azbuka koja sadrži sva slova srpske latinice (velika i mala) i razmak ($\sqcup$), tada su `Fruška␣Gora` i `šđćčž` niske nad azbukom A. Čak i u prvom slučaju, u odnosu na uobičajeno poimanje reči u svakodnevnom govoru ne bi zvučalo korektno nazvati niz karaktera `Fruška␣Gora`, među kojima je i razmak, jednom `rečju`.

Skup svih niski nad azbukom Σ, uključujući i praznu nisku, označavaćemo sa Σ^*, dok ćemo skup nepraznih niski nad azbukom Σ označavati sa Σ^+. Koristeći matematičku notaciju, pretpostavku da je x niska nad azbukom Σ označavaćemo skraćeno zapisom $x \in \Sigma^*$.

Na osnovu navedenih definicija, elektronski tekst se može smatrati jednom niskom nad azbukom $\mathcal{K}$, gde je $\mathcal{K}$ neki od karakterskih skupova navedenih u odeljku 2.2.

Definicija 2.4. Neka su $x, u, v, w \in \Sigma^*$ i neka se niska x može predstaviti kao rezultat dopisivanja niski u, v, w, tj. $x = uvw$. Tada svaka od niski u, v, w predstavlja **faktor** ili **podnisku** (eng. **substring**) niske x. Pri tome:

- Niska u se još naziva i **levim faktorom** ili **prefiksom** niske x. U slučaju da je $|u| < |x|$, kaže se da je u **pravi levi faktor**, odnosno **pravi prefiks** niske x.
- Niska w se još naziva i **desnim faktorom** ili **sufiksom** niske x. U slučaju da je $|w| < |x|$, kaže se da je w **pravi desni faktor**, odnosno **pravi sufiks** niske x.
- U slučaju da su u i w neprazne niske, v predstavlja **pravi faktor**, odnosno **pravu podnisku**, odnosno **infiks** niske x.

Definicija 2.5. Proizvoljni skup niski nad zadatom azbukom Σ naziva se **(formalnim) jezikom** nad tom azbukom.

TFJ definiše različite formalizme za specifikaciju formalnih jezika, a jedan od tih formalizama su i regularni izrazi. Regularnim izrazima nije moguće opisati sve moguće formalne jezike nad zadatom azbukom, već samo jednu potklasu jezika koji se u hijerarhiji Čomskog nazivaju **jezici tipa 3** ili **regularni skupovi** ([Vitas, 2006: 62]).

Kao preteča regularnih izraza, u literaturi se obično navodi model neurona opisan u radu dvojice neurofiziologa 1943. godine ([McCulloch & Pitts, 1943]). Prvu formalnu definiciju regularnih izraza je dao Klini ([Kleene, 1951, 1956]). Tom definicijom se regularni izrazi uvode kao algebarska notacija za opis regularnih skupova (jezika).

Definicija 2.6. Neka je Σ azbuka. **Regularni izrazi nad azbukom** Σ se definišu rekurzivno na sledeći način:

(i) $\emptyset$ je regularan izraz kojim se označava prazan skup;

(ii) ε je regularan izraz kojim se označava skup $\{\varepsilon\}$ koji sadrži samo praznu nisku;

(iii) Ako je a simbol azbuke Σ, tada je a regularni izraz kojim se označava skup $\{a\}$;

(iv) Ako su r_1 i r_2 regularni izrazi kojim su označeni jezici L_1 i L_2 tim redom, $(r_1|r_2)$ je regularni izraz kojim se označava[16] **unija** jezika L_1 i L_2, tj. skup

$$L_1 \cup L_2 = \{x \mid x \in L_1 \vee x \in L_2\};$$

(v) Ako su r_1 i r_2 regularni izrazi kojim su označeni jezici L_1 i L_2 tim redom, $(r_1 r_2)$ je regularni izraz kojim se označava **proizvod** jezika L_1 i L_2, tj. skup

$$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\};$$

(vi) Ako je r regularni izraz kojim je označen jezik L, $r*$ je regularni izraz kojim se označava[17] **Klinijevo zatvorenje** jezika L, tj. skup[18]

$$L^* = \{x \mid (\exists n \in \mathbb{N}_0)(\exists x_1, x_2, \ldots, x_n \in L)(x = x_1 x_2 \cdots x_n)\};$$

(vii) Ako je r regularni izraz kojim je označen jezik L, tada se regularnim izrazom (r) takođe označava jezik L;

(viii) Ništa osim (i)–(vii) nije regularan izraz nad azbukom Σ.

Definicija 2.7. Formalni jezici nad azbukom Σ, koji se mogu označiti regularnim izrazom nad tom azbukom, predstavljaju klasu **regularnih skupova** (jezika) nad azbukom Σ u oznaci $\mathcal{R}(\Sigma)$.

Definicija 2.8. Operacije sa regularnim izrazima i skupovima (iv)–(vi) nazivaju se **regularne operacije**.

[16] U teoriji formalnih jezika se unija regularnih skupova zapravo označava sa $r_1 + r_2$. Međutim, standard POSIX, čiju specifikaciju regularnih izraza implementira većina programskih jezika i programa za uređivanje teksta, označava uniju specijalnim karakterom |, dok + ima sasvim drugo specijalno značenje. Pošto će se u nastavku izlaganja koristiti POSIX-notacija za regularne izraze, radi jednostavnosti se ista notacija koristi i u definiciji regularnih izraza umesto uobičajene notacije teorije formalnih jezika.

[17] Kao u slučaju unije jezika, u definiciji Klinijevog zatvorenja i u daljem izlaganju umesto uobičajene notacije teorije formalnih jezika za regularne izraze (r^*) koristi se POSIX-notacija ($r*$).

[18] U definiciji Klinijevog zatvorenja se, radi jednostavnosti izlaganja, prećutno koristi da su operacije dopisivanja niski i proizvoda jezika asocijativne. Strožije zasnivanje regularnih izraza može se naći, na primer, u ([Vitas, 2006]).

Neformalno, proizvod dva jezika L_1 i L_2 je skup svih niski dobijenih dopisivanjem proizvoljne niske jezika L_2 na proizvoljnu nisku jezika L_1. Takođe, Klinijevo zatvorenje jezika L može da se posmatra kao skup svih niski nad azbukom L ako niske jezika L tretiramo kao simbole jedne azbuke.

Stavka (vii) definicije 2.6 omogućava da se u zapisu regularnog izraza izostave zagrade. Međutim, u tom slučaju se pojedini regularni izrazi mogu tumačiti na različite načine, kao što pokazuje Primer 2.2.

Primer 2.2. Neka se azbuka Σ sastoji od simbola karakterskog skupa ASCII. Jedan od regularnih izraza nad tom azbukom je `pismo|a` čija su moguća tumačenja `pism(o|a)` i `(pismo)|a`. Prvo tumačenje odgovara regularnom skupu

$$\{pism\}(\{o\} \cup \{a\}) = \{pism\}\{o, a\} = \{pismo, pisma\},$$

dok potonje tumačenje označava regularni skup

$$\{pismo\} \cup \{a\} = \{pismo, a\}.$$

Kako bi svaki regularni izraz nad zadatom azbukom imao jednoznačno tumačenje, regularnim operacijama se pridružuje različiti prioritet: najveći prioritet ima Klinijevo zatvorenje, potom proizvod jezika, a najmanji — unija jezika.

Primer 2.3. U skladu sa prioritetom regularnih operacija, regularni izraz `pismo|a` iz Primera 2.2 može da se tumači na jedan jedini način, kao oznaka regularnog skupa

$$\{pismo\} \cup \{a\} = \{pismo, a\},$$

tj. regularni izrazi `pismo|a` i `(pismo)|a` imaju isto značenje.

Primer 2.4. Neka je azbuka ista kao u Primeru 2.2. Regularni izrazi `pism(o|a)` i `pismo|pisma` označavaju isti regularni skup:

$$\{pism\}(\{o\} \cup \{a\}) = \{pism\}\{o, a\} = \{pismo, pisma\} = \{pismo\} \cup \{pisma\}.$$

Primer 2.5. Neka je azbuka ista kao u Primeru 2.2. Neka su $A = \{ha\}$ i $B = \{u\}$ jednočlani regularni skupovi. Tada se Klinijevo zatvorenje skupa A, odnosno skupa

B, svodi na skup niski koje se sastoje iz nula ili više pojavljivanja niske `ha`, odnosno niske `u`:

$$\begin{aligned} A^* &= \{\varepsilon, ha, haha, hahaha, hahahaha, hahahahaha, \ldots\}, \\ B^* &= \{\varepsilon, u, uu, uuu, uuuu, uuuuu, \ldots\}. \end{aligned}$$

Primer 2.4 ilustruje da različiti regularni izrazi mogu označavati isti regularni skup. Štaviše, postoji beskonačno mnogo regularnih izraza kojima se može označiti isti regularni skup i oni se smatraju međusobno jednakim ili ekvivalentnim. Svojstva regularnih operacija, koja navodimo bez dokaza u Tvrđenju 2.1, omogućavaju da se regularni izraz transformiše u ekvivalentan regularni izraz sa jednostavnijim zapisom.

Tvrđenje 2.1. Neka su p, q i r regularni izrazi nad azbukom Σ. Tada važi:

(a) $p|q = q|p$ (komutativnost unije);

(b) $(p|q)|r = p|(q|r)$ (asocijativnost unije);

(c) $(pq)r = p(qr)$ (asocijativnost proizvoda);

(d) $p\varepsilon = \varepsilon p = p$ (prazna niska je neutralni element operacije dopisivanja);

(e) $p(q|r) = (pq)|(pr)$ i $(p|q)r = (pr)|(qr)$ (distributivnost proizvoda u odnosu na uniju);

(f) $p|p = p$ (idempotentnost unije);

(g) $(p*)* = p*$ (idempotentnost Klinijevog zatvorenja);

(h) $p* |p = p*$. □

Tokenizacija

Definicija 2.9. Neka je $\mathcal{K}$ karakterski skup i T elektronski tekst, odnosno niska nad azbukom $\mathcal{K}$. Pretpostavimo da se karakterski skup $\mathcal{K}$ može predstaviti kao unija dva neprazna disjunktna skupa, tj. $\mathcal{K} = G \cup A$, $G \neq \emptyset$, $A \neq \emptyset$, $G \cap A = \emptyset$. Elemente skupa G nazivamo **graničnicima** ili **separatorima** (eng. **delimiters**).

Definicija 2.10. Ako se niska T može predstaviti u obliku

$$T = pu_1v_1u_2v_2 \cdots u_{m-1}v_{m-1}u_mq \tag{2.1}$$

gde su $u_1, u_2, \ldots, u_m$ niske nad azbukom A, $v_1, v_2, \ldots, v_{m-1} \in G^+$, a $p, q \in G^*$, tada se reprezentacija 2.1 naziva **tokenizacija** teksta T u odnosu na skup separatora G, a niske $u_1, u_2, \ldots, u_m$ se nazivaju **tokenima**.

Primer 2.6. Ako je $\mathcal{K}$ karakterski skup ISO 8859-2, a skup separatora G_1 sastavljen od karaktera koji predstavljaju beline (razmak, tabulatori, oznake kraja linije), i $A = \mathcal{K} \setminus G$, tada se tokenizacijom niske T

```
Kakav␣14.␣januar,␣zar␣ne?
```

u odnosu na skup separatora G_1 mogu uočiti tokeni:

```
Kakav
14.
januar,
zar
ne?
```

U računarstvu se skup belina najčešće koristi kao skup separatora u procesu tokenizacije.

Primer 2.7. Ako u primeru 2.6 umesto G_1 uočimo skup separatora G_2 koji, osim belina, sadrži i sve nealfanumeričke karaktere (tj. karaktere koji ne predstavljaju ni slova ni cifre), tada se kao tokeni izdvajaju:

```
Kakav
14
januar
zar
ne
```

Tokenizacija teksta na prirodnom jeziku U računarstvu se kao elementi separatorskog skupa obično uzimaju beline (karakteri razmak, tabulator, znak za novi red), ali to nije dovoljno dobro kada su prirodni jezici u pitanju (v. Primer 2.6). U slučaju prirodnih nesegmentiranih jezika (poput kineskog i većine ostalih orijentalnih jezika), nema granice između tokena, već se oni pišu neposredno jedan pored drugog. S druge strane, kod segmentiranih jezika (kakav je i srpski) pored belina treba uzeti u obzir i interpunkciju, ali ni tada se separatorski skup ne može precizno definisati. Naime, znaci interpunkcije se odlikuju višeznačnošću koja je, pre svega, uslovljena kontekstom u kome se pojavljuju. Stoga je teško doneti odluku da li ih tretirati kao pojedinačne tokene ili kao delove drugih tokena:

- crtica se pojavljuje u polusloženicama poput `general-major`, koristi se za rastavljanje reči na kraju reda, ali i u ulozi separatora klauza (umetnutih rečenica). Crtica kao separator klauza je token za sebe, ali u slučaju polusloženice prirodnije je da crtica bude tretirana kao deo (jednog) tokena (`general-major`) nego kao pojedinačni token;

- apostrof se javlja u funkciji navodnika, ali i kao zamena za izostavljeno slovo (na primer `rek'o`) i kao deo prezimena (`O'Brajen`). U prvom slučaju apostrof je token za sebe, dok je u ostalim slučajevima prirodnije da bude tretiran kao deo (jednog) tokena (`rek'o`, odnosno `O'Brajen`)[19];

- zapeta se koristi i kao separator delova rečenice, ali i u decimalnom zapisu brojeva. Ponovo, u prvom slučaju je prirodnije tretirati zapetu kao pojedinačni token, a u drugom — kao deo tokena;

- tačka može biti separator hiljada u zapisu brojeva (npr. $9.812,35$), deo skraćenice, a koristi se i kao terminator rečenice (bilo kao jedna ili tri tačke[20]). Jedna tačka kao kraj rečenice se može tretirati kao pojedinačni token, dok u svim drugim slučajevima može da se tretira kao deo tokena. Posebno je zanimljiv

[19]Pojedini tokenizatori koriste specijalne liste kako bi pre tokenizacije zamenili apostrof odgovarajućim izostavljenim slovom, tj. od skraćenog oblika proizveli puni oblik. Tipičan primer je zamena engleskih skraćenih oblika `can't`, `didn't`, `haven't`, `hasn't`, `isn't`, `I'm`, `you're`, itd. punim oblicima `can not`, `did not`, `have not`, `has not`, `is not`, `I am`, `you are`, itd.

[20]Pravopisni znak '...' ('tri tačke') se u većini kodnih rasporeda predstavlja jednim ne-ASCII karakterom, ali se u pisanim tekstovima često pojavljuje kodiran i kao tri ASCII karaktera '.'.

slučaj kada se skraćenica nalazi na kraju rečenice, te tačka igra dvostruku ulogu.

Naposletku, treba primetiti da se beline, sem kao separatori reči teksta, pojavljuju i u oblicima **višečlanih leksema** (eng. **Multi Word Unit / Expression**, skr. **MWU / MWE**) poput `Novi Beograd`, kao i u okviru **imenovanih entiteta** (eng. **Named Entity**) kao na primer `20 procenata`, `tri do pet miliona dinara`, `8. mart`, itd. Oblike višečlanih leksema i imenovane entitete je takođe prirodnije tretirati kao pojedinačne tokene umesto kao nizove tokena (v. pododeljak Plitko parsiranje, str. 110).

U slučaju segmentiranih jezika korpusni lingvisti za skup separatora najčešće biraju nealfanumeričke karaktere zadatog karakterskog skupa. Međutim, pošto je upotreba interpunkcije u tekstovima korpusa bitna za razne primene korpusa (lingvistička istraživanja i aplikacije za obradu prirodnih jezika), skup separatora G se razbija na dva disjunktna podskupa, B i I, koji imaju različiti tretman. Podskup separatora B uglavnom čine beline i one se ignorišu u daljoj obradi, tj. predstavljaju prave separatore u smislu definicije 2.9. S druge strane, svaki element skupa I (pretežno znaci interpunkcije) tretira se istovremeno i kao separator i kao token.

Primer 2.8. Ako karakterski skup $\mathcal{K}$, niska T i skup separatora G_2 imaju iste vrednosti kao u primeru 2.6, i ako elemente skupa G_2 koji nisu beline tretiramo i kao separatore i kao tokene, onda se u okviru niske T izdvajaju sledeći tokeni:

```
Kakav
14
.
januar
,
zar
ne
?
```

Korpusne reči i tipovi U nastavku teksta podrazumevaćemo da se kao skup separatora uvek koristi skup S nealfanumeričkih karaktera, pri čemu se svaki separator koji nije belina tretira i kao pojedinačni token.

Definicija 2.11. Tokene iz definicije 2.10, određene separatorskim skupom S i sastavljene isključivo od slovnih karaktera nazivaćemo **formalnim (tekstuelnim) rečima** (eng. **word form**). Za individualna pojavljivanja formalnih reči u tekstu ili korpusu koristićemo termin **korpusna reč** (eng. **word token**).

Definicija 2.12. Elemente skupa različitih korpusnih reči, odnosno tokena, nazivaćemo respektivno **korpusni tipovi** (eng. **word types**), odnosno **tipovi** (eng. **types**).

U sledećem pasusu se nalazi (ako ne razlikujemo velika i mala slova) ukupno 12 tokena i svega 8 tipova, odnosno 8 korpusnih reči i 5 korpusnih tipova (Tabela 2.6).

- Ko je rekao *ja*?
- Ja sam rekao *ja*.

U odeljku 3.3 će biti više reči o ulozi tokena i tipova u statističkoj analizi korpusa, kao i o različitim pristupima njihovom prebrojavanju.

Tabela 2.6: Primer identifikacije tokena, tipova, korpusnih reči i korpusnih tipova u tekstu.

Token	Tip	Korpusna reč	Korpusni tip
-	-		
Ko	ko	Ko	ko
je	je	je	je
rekao	rekao	rekao	rekao
ja	ja	ja	ja
?	?		
-	-		
Ja	ja	Ja	ja
sam	sam	sam	sam
rekao	rekao	rekao	rekao
ja	ja	ja	ja
.	.		

Definicija 2.13. Program koji automatski obavlja tokenizaciju elektronskog teksta naziva se **tokenizator** (eng. **tokenizer**).

Primeri tokenizatora su detaljno obrađeni u poglavlju 4.

Uticaj kodne sheme teksta na tokenizaciju Jedan od preduslova za rad tokenizatora je utvrđivanje kodne sheme kojom je kodiran ulazni tekst. Ponekad tekst poseduje eksplicitnu informaciju o primenjenoj kodnoj shemi (HTML i XML-datoteke), ali, kada to nije slučaj, precizno automatsko utvrđivanje kodne sheme teksta nije uvek moguće. Poseban problem predstavljaju osmobitne kodne sheme i UTF-8 jer na isti način kodiraju prvih 128 ASCII-karaktera, a ako se u tekstu pojavljuju samo ASCII-karakteri, praktično svaki od tih kodnih shema može biti u pitanju. Oznaka redosleda bajtova (BOM) omogućava da se potencijalni kandidati suze na kodne sheme koje kodiraju Unicode-karaktere.

U slučaju ćiriličnog teksta tokenizatori mogu da analiziraju opseg i raspodelu celih brojeva koji se pojavljuju kao sadržaj bajtova kodiranih karaktera, i budući da kodne sheme poput ISO 8859-5 i UTF-16 koriste različite opsege kodnih pozicija za ćirilična slova, prostom heuristikom mogu da identifikuju primenjenu kodnu shemu ([Palmer, 2010: 12–3]).

Dehifenacija Tokom pripreme za štampu uobičajeno je da se reči na kraju linija rastavljaju na slogove kako bi se poravnale margine elektronskog teksta i postigao ujednačen razmak između pojedinačnih reči u tekstu. Proces rastavljanja reči na slogove se još naziva **hifenacija** (eng. **hyphenation**)[21]. Hifenacija otežava izdvajanje tokena u tekstu jer je prethodno potrebno poništiti efekte rastavljanja reči na slogove, odnosno obaviti **dehifenaciju** teksta.

Proces dehifenacije deluje kao jednostavan zadatak: kad god se na kraju linije prepozna crtica koja neposredno sledi za nizom slova, treba eliminisati crticu i znak za novi red i tako rekonstruisati rastavljenu reč. (Da ne bi došlo do spajanja linija u kojima su se nalazili delovi rekonstruisane rastavljene reči, ispred ili iza te reči se može ubaciti znak za novi red, tako da dehifenacija ne promeni broj linija teksta). Međutim, to je samo jedan od mogućih slučajeva ([Schmid, 2008: 531]):

[21] Naziv *hifenacija* potiče od engleske reči *hyphen* za interpunkcijski znak crtica ('-') kojim se razdvajaju slogovi rastavljene reči.

(dh1) Pretpostavimo da je reč `srednjoevropske` rastavljena na `srednjo-` i `evropske`. U tom slučaju se brišu crtica i znak za novi red;

(dh2) Pretpostavimo da je reč `srednjo-evropske` rastavljena na `srednjo-` i `evropske`. U tom slučaju treba obrisati samo znak za novi red;

(dh3) Pretpostavimo da je niz reči `srednjo- i istočno- evropske` rastavljen na `srednjo-` i ostatak. U tom slučaju ne treba brisati ni crticu ni znak za novi red, odnosno znak za novi red treba tretirati kao separator tokena.

Od navedenih slučajeva najverovatniji je (dh1), dok je (dh3) „dovoljno redak da se može ignorisati a da pritom ne dođe do primetnog smanjenja tačnosti" ([Schmid, 2008: 533]).

Kodna shema primenjena u tekstu takođe može da utiče na tačnost i preciznost dehifenacije. Naime, dok ASCII sadrži samo jedan karakter (na kodnoj poziciji 45) koji se istovremeno koristi za sve vrste crta i crticu, uključujući i matematički simbol za oduzimanje (minus), dotle Unicode pravi razliku između tih simbola i uvodi više od dvadeset različitih karaktera, od kojih se najčešće koriste njih pet (Tabela 2.7). Ukoliko se ti Unicode-karakteri koriste u tekstu dosledno svojoj nameni, proces dehifenacije je svakako lakši. Praksa je nažalost drugačija, tako da se u tekstu najčešće koristi samo jedan od tih karaktera (ASCII-crtica), prevashodno zbog toga što je jedini direktno dostupan na tastaturi, dok se ostali karakteri za crte ili umeću u tekst iz liste specijalnih karaktera (npr. u programu Microsoft Word) ili se dobijaju višestrukim kucanjem ASCII-crtice (na primer, u programskom sistemu LaTeX dvostruka ASCII-crtica proizvodi en-crtu, a trostruka — em-crtu).

Tabela 2.7: Unicode-karakteri za predstavljanje crta i crtica

glif	**naziv**	**engleski naziv**	**kodna pozicija**
-	crtica i minus	hyphen and minus sign	45
(ne postoji u TeX-u)	cifra-crta	figure-dash	8210
–	en-crta	n-dash	8211
—	em-crta	m-dash	8212
(ne postoji u TeX-u)	horizontalna crta	horizontal bar, quotation dash	8213

Navedeni karakteri za crte i crticu imaju posebna imena u engleskom jeziku, dok srpski pravopis ([Pešikan et al., 2009]) govori isključivo o crti i crtici, pri čemu

crtici odgovara *hyphen* (u slučajevima kada se crtica koristi za rastavljanje reči na slogove i zapisivanje polusloženica poput `auto-mehaničar`) i ponekad en-crta (kada se crtica koristi u zapisu telefonskih brojeva poput `333-444`), dok crti odgovara em-crta (razdvajanje umetnutih rečenica) i ponekad en-crta (kada se crta koristi za razdvajanje intervala brojeva, na primer godina rođenja i smrti `1787-1864`, ili između naziva mesta da bi se označio pravac kretanja, na primer `Beograd-Niš`).

Kada je teško razrešiti o kom slučaju (dh1–dh3) se radi, mogu se iskoristiti dodatne informacije, pre svega leksičke, i na taj način povećati preciznost dehifenacije. Mikheev (2003), navodi sledeći algoritam za koji tvrdi da smanjuje udeo grešaka dehifenacije sa 4.9% na 0.9%:

1. neka se na kraju jedne linije nalazi niska x i crtica, a sledeća linija počinje niskom y;

2. ako se niska dobijena dopisivanjem niske y na nisku x (u oznaci xy) nalazi u rečniku, tada se ispisuje xy (slučaj (dh1));

3. ako to nije slučaj, a obe niske x i y su prisutne u rečniku, onda se ispisuje x-y (slučaj (dh2), polusloženice);

4. u protivnom se ispisuje xy (slučaj (dh3) se ignoriše i sve što se ne može prethodnim pravilom svesti na slučajeve (dh1) ili (dh2) tretira se kao slučaj (dh1)).

Umesto rečnika mogu se iskoristiti liste tokena i njihove učestanosti (frekvencije) ekstrahovane iz postojećih korpusa, pri čemu se koristi sledeće pravilo ([Schmid, 2008: 533]):

1. ako neka od niski xy i x-y ima veću učestanost u odnosu na drugu, ispisuje se niska sa većom učestanošću;

2. ako im je učestanost ista i obe niske se nalaze u rečniku, ispisuje se x-y;

3. u protivnom se ispisuje xy.

Greške u tekstu i tokenizacija Na preciznost procesa tokenizacije utiču i greške u tekstu, posebno izostavljeni razmaci između dve susedne reči. Jedan pristup rešavanju ovog problema je korišćenje programa za proveru pravopisa ili analiza teksta pomoću morfološkog elektronskog rečnika. Međutim, i na taj način nije moguće razrešiti sve višeznačnosti, što ilustruje primer `pojavi`: bez detaljne analize konteksta ne može se doneti odluka da li su se u tekstu pojavila dva tokena (`po javi`) ili jedan (`pojavi`), pošto su sva tri tokena prisutna u rečniku srpskog jezika.

Drugi pristup koristi informacije o frekvencijama tokena u postojećim korpusima, na primer za razrešavanje slučajeva kada su dve tekstuelne reči razdvojene tačkom bez razmaka ([Schmid, 2008: 531]). Ako je sporna niska oblika $s.r$, i ako sa $f(s)$ označimo učestanost niske s, a sa $f(.r)$ učestanost niske r na početku rečenice, pri čemu je N veličina korpusa iz kog su ekstrahovane učestanosti $f(s)$ i $f(.r)$, tada Schmid (2008) predlaže sledeće pravilo (koje se na sličan način može primeniti i na ostale znake interpunkcije): ako je $f(s) \cdot f(.r) > N \cdot f(s.r)$, tada nisku $s.r$ treba razbiti na tri tokena, tj. token s, tačku i token r.

Identifikacija kraja rečenice

Tokenizacija se često kombinuje sa procesom automatske identifikacije rečenica, tj. obično se najpre elektronski tekst podeli na rečenice, a onda se u okviru svake rečenice izdvajaju njeni tokeni. Osim u slučaju tokenizacije, identifikacija rečenica je bitan preduslov za razne tipove automatske obrade elektronskog teksta poput sintaksičke analize ili parsiranja (v. odeljak 2.3, str. 105), automatske morfološke anotacije (lematizacija i određivanje vrste reči, v. odeljak 2.4, str. 140), paralelizacije tekstova (v. odeljak 6.5, str. 290), itd. Većina složenih programskih sistema za automatsku obradu prirodnog jezika sadrži poseban **modul za detekciju granice između rečenica** (eng. **sentence boundary detection module**, skr. **SBD-modul**) koji se koristi u ranoj fazi obrade i čija preciznost utiče i na uspešnost kasnijih faza obrade.

Umesto identifikovanja rečenica kao lingvističkih jedinica, uobičajeno je da se identifikuje kraj rečenice. U slučaju jezika, poput srpskog, kod kojih se posebnim znacima interpunkcije (tačka, uzvičnik, upitnik, itd.) označava kraj rečenice, termi-

nator rečenice se lakše identifikuje u odnosu na tekstove na jezicima koji ne samo da nemaju terminatore rečenica, već ni separatore tokena (kao što je, na primer, slučaj sa tajlandskim).

Čak i kada u tekstu postoje terminatori rečenice, njihova identifikacija nije nimalo trivijalan problem. Za početak, znaci interpunkcije koji se koriste kao terminatori rečenice variraju od jezika do jezika ([Palmer, 2010: 23]). Evropski jezici, pored najčešćih terminatora rečenice (tačke, uzvičnika, upitnika) ponekad koriste i tačku zapetu, dve tačke, tri tačke, pa čak i crticu i zapetu da označe kraj rečenice. S druge strane, kineski i japanski jezik koriste kombinaciju ideografskih i evropskih simbola interpunkcije, pa se tako umesto evropskog simbola za tačku koristi ◦, dok su upitnik i uzvičnik slični evropskim oznakama, ali drugačije veličine.

Ukoliko je tekst jednojezičan, skup potencijalnih terminatora rečenice se sužava, ali problem njihove identifikacije je i dalje kompleksan s obzirom da pravopis dozvoljava da se znaci interpunkcije kojima se označavaju terminatori rečenica koriste i u drugim značenjima.

Za tačku, kao jedan od terminatora rečenice, je već navedeno (str. 94) da se može koristiti i u zapisu skraćenica (`itd.`, `i sl.`) i brojeva, bilo kao separator hiljada (na primer u srpskom pravopisu), bilo kao decimalna tačka (na primer u engleskom pravopisu), ali takođe i u zapisu datuma (`1.05.2010. godine`), adresa korisnika e-pošte (`laza@fil.bg.ac.rs`) i elektronskih adresa resursa na vebu (URL, na primer `http://www.fil.bg.ac.rs`), prilikom navođenja referenci u tekstu (`Poglavlje 2.1`), itd.

Upitnik i uzvičnik, kao terminatori rečenica, nekada se pojavljuju i unutar rečenice sa drugačijom ulogom. Uzvičnik se može pojaviti kao deo vlastitog imena (na primer, naziv veb servisa `Yahoo!`), ali i za naglašavanja unutar rečenica (primeri (C1)) i (C2)), dok se upitnik može pojaviti u seriji kratkih pitanja (primer C3). Ponekad se uzvičnik i upitnik kombinuju (`?!` ili `!?`) kako na kraju rečenice, tako i na kraju pojedine klauze kako bi se izrazilo čuđenje ili neverica (primeri (C4) i (C5)).

(C1) „Znaš li ti da je car lud - lud! - i da se održava samo krvlju svojih pobeda koje ne vode nikud i ničem?" (Ivo Andrić, *Travnička hronika*)

(C2) „A bez priča, reče otac, bez pravih priča, ceo ovaj svet će - cap! - on puknu prstima." (David Albahari, *Šetnje pored reke*)

(C3) „(ah, koliko je meseci molila bez uspeha Srbu da je povede sa sobom u pozorište, a sada odjednom ... zbog čega? kako? i kako je imao svetlo lice!)" (Branimir Ćosić, *Dva carstva*)

(C4) „Kako ga nije sramota?! ... Lud čovek! Šta ima da učini?!" (Radoje Domanović, *Mrtvo more*)

(C5) „Nisam ti pričao!?" (Danilo Kiš, *Mansarda*)

Da bi se tokom segmentacije teksta na rečenice utvrdilo da li posmatrani znak interpunkcije predstavlja kraj rečenice, osim provere da li pripada skupu potencijalnih terminatora rečenice, neophodno je analizirati levi i desni kontekst tog znaka interpunkcije. U osnovi svih analiza desnog konteksta kandidata za kraj rečenice u tekstovima na nekom od evropskih jezika je algoritam poznat pod nazivom „tačka-razmak-veliko slovo" ([Mikheev, 2003: 213]). Svoj naziv algoritam duguje tome što se njegovom primenom svi kandidati za terminator rečenice u tekstu (tačka, upitnik, uzvičnik), za kojima sledi bar jedan razmak i veliko slovo, identifikuju kao terminatori rečenice. Algoritam se može proširiti ako se, osim razmaka, u obzir uzmu i drugi karakteri koji se pojavljuju između terminatora rečenice i prve reči naredne rečenice, poput zatvorene zagrade ili znaka navoda. Međutim, da bi se poboljšala preciznost ovog algoritma mora se dodatno analizirati i levi i desni kontekst potencijalnog terminatora rečenice, jer se ovakvim algoritmom kao kraj rečenice može pogrešno idenitifikovati tačka kojoj prethodi skraćenica ili datum, a za kojom sledi vlastito ime (na primer `prof. Petar Petrović` ili `29. VI 2012.`). Analiza levog i desnog konteksta potencijalnog terminatora rečenice zahteva konsultovanje odgovarajućih leksičkih resursa (elektronski morfološki rečnici, liste vlastitih imena, liste skraćenica, itd.). Poboljšana verzija algoritma „tačka-razmak-veliko slovo" (ukoliko se razmatra samo tačka kao potencijalni terminator rečenice) bi izgledala ovako ([Mikheev, 2003]212–213):

(i) ako tački prethodi tekstuelna reč koja nije skraćenica, gotovo sigurno[22] tačka

[22]Naime, može se dogoditi da tačka nije kraj rečenice jer iza nje stoji rimski broj, kao u primeru

predstavlja kraj rečenice (na primer `... bavio. Sada je...`);

(ii) ako tački prethodi skraćenica, mora se konsultovati desni kontekst:

(a) ako za tačkom sledi znak interpunkcije, broj ili tekstuelna reč koja počinje malim slovom, onda tačka gotovo sigurno[23] ne predstavlja kraj rečenice (na primer `... prof. dr Petar Petrović...`);

(b) ako za tačkom sledi tekstuelna reč koja počinje velikim slovom, i ta tekstuelna reč nije vlastita imenica, onda je tačka najverovatnije[24] kraj rečenice (na primer `..., itd. Nastade tišina.`);

(c) ako za tačkom sledi tekstuelna reč koja počinje velikim slovom i jeste vlastita imenica, onda nije jasno da li je tačka kraj rečenice ili ne (u primeru `... prof. Petar Petrović...` tačka nije kraj rečenice, ali jeste u primeru `... činjenice, argumenti, itd. Petar Petrović zaćuta.`);

Pristupi unapređivanju algoritma „tačka-razmak-veliko slovo" se mogu svrstati u dve grupe u zavisnosti od toga da li SBD-modul koristi ručno konstruisana pravila ili statističke metode.

Ručno konstruisana pravila U slučaju SBD-modula prve grupe, administrator sistema i korisnici zadaju pravila koristeći formalizme poput regularnih izraza ili kontekstno slobodnih gramatika[25]. Sistem potom transformiše pravilo u automat (regularni izraz u odgovarajući konačni transduktor, kontekstno slobodnu gramatiku u odgovarajući potisni transduktor[26]) koji u tekstu prepoznaje i anotira sve niske opisane pravilom. Jedna moguća implementacija pravila za segmentaciju na rečenice

`29. VI 2012.`

[23]Izuzetak su tipografske greške u slučajevima kada tačka jeste kraj rečenice, ali sledeća rečenica počinje malim slovom.

[24]Pojedini programi za uređivanje i formatiranje teksta, na primer MS Word, imaju opciju da automatski ispravljaju tekst dok korisnik kuca (ako korisnik ne podesi program drugačije). Na taj način se u svakoj reči koja sledi za tačkom, a koja je otkucana sa malim početnim slovom, početno slovo automatski zamenjuje odgovarajućim velikim slovom bez obzira da li je na toj poziciji kraj rečenice ili ne.

[25]Regularni izrazi su detaljno opisani u odeljku 2.3, str. 89, dok se o kontekstno slobodnim gramatikama može više naći, na primer, u [Vitas, 2006], glava 5.

[26]O konačnim automatima i transduktorima, potisnim automatima i transduktorima se može naći više u, na primer, [Vitas, 2006] (glave 3 i 6), [Roche & Shabes, 1997], [Maurel & Guenthner, 2005]

tekstova na srpskom jeziku opisana je u odeljku 3.1, pododeljak Segmentacija na rečenice[27].

Regularni izrazi su pogodni za opis elemenata leksičkih resursa koje koristi SBD-modul (liste vlastitih imena, liste skraćenica, arapski i rimski brojevi, datumi, itd.), što prilikom formulisanja pravila za segmentaciju teksta na rečenice omogućava veću izražajnost i preciznost.

Primeri sistema koji koriste SBD-module zasnovane na pravilima su GATE/ANNIE ([Cunningham et al., 2002]), Intex/NooJ ([Silberztein, 2003]), Unitex ([Paumier, 2011]), itd.

Drugi pristup unapređivanju algoritma „tačka-razmak-veliko slovo" koristi statističke metode, pre svega, **nadgledano i nenadgledano mašinsko učenje** (eng. **supervised and unsupervised machine learning**).

Mašinsko učenje Mašinsko učenje je „podoblast veštačke inteligencije, posvećena razvoju algoritama koji uče ili unapređuju svoje performanse na osnovu iskustva ili prethodne interakcije sa podacima" ([Pustejovsky & Stubbs, 2012]).

Sistem zasnovan na mašinskom učenju se obučava („trenira") na osnovu jedne grupe podataka koja se naziva **skup za treniranje** (eng. **training set**), a primenje se na drugu grupu podataka, tj. na **skup za testiranje** (eng. **test set**). Pojedinačni podaci oba skupa se nazivaju instance.

Određivanje granice između rečenice pomoću nadgledanog mašinskog učenja svodi se na **problem klasifikacije** (eng. **classification problem**). U opštem slučaju se problem klasifikacije sastoji u tome da se svaka instanca skupa za testiranje anotira jednim od obeležja iz zadatog skupa obeležja. U slučaju segmentacije teksta na rečenice, instance skupa za testiranje su znaci interpunkcije kao kandidati za terminator rečenice, dok je skup obeležja binaran, tj. instanca se ili anotira kao terminator rečenice ili kao znak interpunkcije koji ne predstavlja kraj rečenice. Skup za treniranje u slučaju nadgledanog mašinskog učenja je tekst ili skup tekstova u kojima su kandidati za terminator rečenice već obeleženi jednim od dva moguća obeležja.

[27]Razlog za odlaganje navođenja konkretnog primera pravila SBD-modula je nemogućnost da se dovoljno koncizno i pregledno objasni mehanizam pravila pre no što se izloži sintaksa regularnih izraza u skladu sa standardom POSIX (odeljak 3.1, pododeljak POSIX: prošireni regularni izrazi).

Algoritmi nadgledanog mašinskog učenja (stabla odlučivanja, princip maksimalne entropije, neuronske mreže) koriste statistička svojstva anotiranog teksta kako bi realizovao **klasifikator** (eng. **classifier**) — program koji će, koristeći iskustvo na osnovu skupa za treniranje, moći samostalno da anotira tekst sa kojim se prethodno nije susreo. Drugim rečima, obučavajući se na tekstovima u kojima su rečenice jasno razgraničene, klasifikator će odrediti granice između rečenica u proizvoljnom tekstu na osnovu statističkih svojstava levog i desnog konteksta potencijalnih terminatora rečenice: pravopisa, dužine reči, sufiksa, prefiksa, vrste reči (uključujući i podvrste poput vlastitih imena ili različitih klasa skraćenica), itd. ([Palmer, 2010: 25]).

Da bi klasifikator bio što uspešniji, neophodno je da skup za treniranje pokrije sve moguće slučajeve, kao i da raspodela različitih slučajeva omogući klasifikatoru da pravilno izabere odgovarajuće obeležje. Stoga priprema skupa za treniranje klasifikatora nije nimalo jednostavan zadatak, a zahteva i ručno obeležavanje teksta, koncentraciju i konzistentnost ljudi koji na tome rade, pri čemu pre početka izrade treba jasno ustanoviti kriterijume šta će se sve smatrati rečenicom.

Da bi se izbegla izrada prethodno anotiranog skupa za treniranje, pribegava se nenadgledanom mašinskom učenju, što se svodi na problem **klasterizacije ili grupisanja** (eng. **clustering**). U opštem slučaju, klasterizacija tokom treniranja deli podatke u klastere ili grupe koristeći neku zadatu meru sličnosti, a onda se različitim klasterima obeležja mogu i ručno dodeliti.

Detaljnija analiza algoritama mašinskog učenja, uopšte i u slučaju segmentacije teksta na rečenice, izlazi iz okvira ovog rada. Neki važniji pristupi izradi SBD-modula korišćenjem nadgledanog i nenadgledanog mašinskog učenja su: [Reynar & Ratnaparkhi, 1997], [Stamatatos et al., 1999] [Palmer & Hearst, 1997], [Schmid, 2000], [Mikheev, 2000, 2002, 2003], [Kiss & Strunk, 2006].

Na kraju treba spomenuti da pojedini sistemi za obradu prirodnih jezika, na primer GATE/ANNIE ([Cunningham et al., 2002]), koriste obe vrste SBD-modula, tj. i SBD-modul zasnovan na pravilima i SBD-modul zasnovan na mašinskom učenju.

Parsiranje

U računarstvu se pod **parsiranjem** (eng. **parsing**) u opštem slučaju podrazumeva „kombinacija prepoznavanja ulazne niske i pridruživanja neke strukture" ([Jurafsky & Martin, 2008: 45]) dok se u užem značenju parsiranje poistovećuje sa (automatskom) sintaksičkom analizom pomoću računara, „čiji je zadatak prepoznavanje rečenice i pridruživanje (odgovarajuće) sintaksičke strukture" ([Jurafsky & Martin, 2008: 427])[28].

Delovi rečenice koji predstavljaju jednu celinu sa određenom funkcijom (npr. subjekat, predikat, objekat, itd.) u okviru prepoznate sintaksičke strukture nazivaju se **konstituenti**. Rezultat parsiranja u užem smislu je najčešće **drvo sintaksičke analize** (eng. **parsing tree**) kojim se predstavljaju sintaksičke relacije između konsituenata rečenice (Slika 2.11). Program koji ulaznoj rečenici pridružuje njeno drvo sintaksičke analize naziva se **parser** (eng. **parser**).

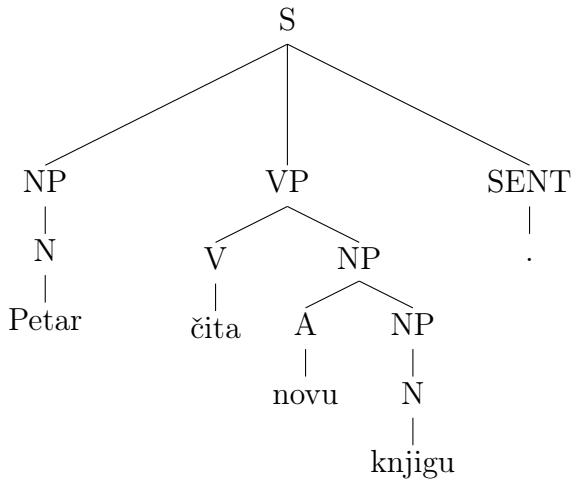
$$
\begin{aligned}
S &\rightarrow NP\ VP\ SENT \\
NP &\rightarrow N \\
NP &\rightarrow A\ NP \\
VP &\rightarrow V\ NP \\
SENT &\rightarrow . \mid ? \mid ! \\
N &\rightarrow \text{Petar} \mid \text{knjigu} \\
V &\rightarrow \text{čita}
\end{aligned}
$$

Slika 2.10: Primer formalne gramatike na osnovu koje se generiše rečenica `Petar čita novu knjigu.`

Parsiranje je bitan zadatak u automatskoj obradi teksta na prirodnom jeziku i često je neophodan preduslov za naknadne obrade i primene poput semantičke analize, analize diskursa, ekstrakcije informacija, mašinskog prevođenja, itd. Takođe, parsiranje je složen zadatak i postoji više različitih pristupa rešavanju tog problema, okupljenih u klase sintaksičkih teorija, među kojima su najznačajnije **gramatike frazne strukture ili konstituentske gramatike** (eng. **phrase structure grammars, constituency grammars**) i **gramatike zavisnosti** (eng. **dependancy grammars**)).

[28]Ukoliko se pojam *parsiranje* koristi u opštem značenju (*analiza*), onda se **morfološko parsiranje** (eng. **morphological parsing**) koristi kao sinonim za morfološku analizu, **sintaksičko parsiranje** (eng. **syntactic parsing**) za sintaksičku analizu, itd.

Tipičan predstavnik konstituentskih gramatika su formalne ili generativne gramatike ([Chomsky, 1957]). Primenom formalne gramatike rečenica se postepeno formira tako što se najpre generišu opštije „fraze" (na primer subjekatska i predikatska „fraza") od kojih se dalje generišu specifičnije „fraze", odnosno konstituenti. Sintaksičku analizu gramatikom frazne strukture (Slika 2.10) ilustruje primer generisanja rečenice `Petar čita novu knjigu.` (Slika 2.11):



Slika 2.11: Primer drveta sintaksičke analize za rečenicu `Petar čita novu knjigu.`

(i) polazeći od simbola rečenice `S`, najpre se generišu dve „fraze", imeničke sintagma (`NP`) i glagolska sintagma (`VP`);

(ii) `NP` se zamenjuje konstituentom `N`, odnosno morfosintaksičkom rečju `Petar`;

(iii) `VP` se zamenjuje konstituentom `V` i imeničkom sintagmom (`NP`). Konstituent `V` se realizuje kao morfosintaksička reč `čita`;

(iv) imenička sintagma `NP`, kao deo glagolske sintagme, zamenjuje se konstituentom `A`, odnosno morfosintaksičkom rečju `novu`, i novom imeničkom sintagmom `NP`. Poslednja imenička sintagma `NP` se zamenjuje konstituentom `N`, odnosno morfosintaksičkom rečju `knjigu`.

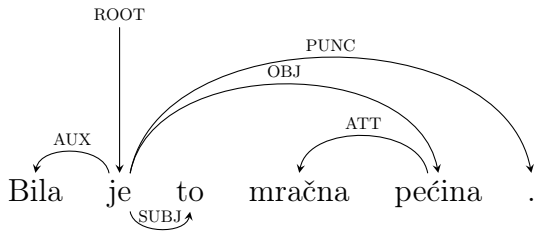
(v) Drvo (Slika 2.11) predstavlja konačan rezultat generisanja i parsiranja polazne rečenice.

Opisane zamene se najčešće mogu obaviti nezavisno od konteksta, te se parseri zasnovani na ovom principu najčešće implementiraju pomoću kontekstno slobodnih gramatika (v. na primer [Vitas, 2006], glava 5, [Jurafsky & Martin, 2008], glave 12–14).

U slučaju gramatika zavisnosti ([Mel'čuk, 1988]) nad skupom morfosintaksičkih reči u okviru rečenice uspostavlja se binarna relacija takva da je morfosintaksička reč x u relaciji sa morfosintaksičkom rečju y ako i samo ako y „zavisi" od x, pri čemu svaka od zavisnosti predstavlja neku gramatičku funkciju (na primer subjekat, objekat, itd.) ili opštu semantičku relaciju (agent, cilj, itd.). Rezultat sintaksičke analize gramatikom zavisnosti se takođe predstavlja drvetom, pri čemu:

- čvorovi drveta, izuzev korena, su morfosintaksičke reči u rečenici;
- (usmerene) grane drveta predstavljaju zavisnosti i usmerene su ka zavisnoj reči,
- svaka grana je obeležena odgovarajućom gramatičkom funkcijom ili semantičkom relacijom;
- koren drveta je specijalan čvor koji je granom povezan sa glagolom kao centralnom („glavnom") rečju u rečenici.

Parsiranjem rečenice `Bila je to mračna pećina.` gramatikom zavisnosti dobija se drvo zavisnosti (Slika 2.12), pri čemu su grane obeležene gramatičkim funkcijama (Tabela 2.8).



Slika 2.12: Drvo zavisnosti za rečenicu `Bila je to mračna pećina.`

Detaljno razmatranje gramatika frazne strukture i gramatika zavisnosti izlazi iz okvira ovog rada.

Tabela 2.8: Neke gramatičke relacije zavisnosti (Slika 2.12)

oznaka zavisnosti	**objašnjenje**
ROOT	„glavna reč" (nosilac značenja rečenice)
AUX	pomoćni glagol
SUBJ	subjekat
OBJ	direktni objekat
ATT	atribut
PUNC	interpunkcija

Banka (sintaksičkih) stabala Kada je u pitanju konstrukcija korpusa, problem parsiranja je neposredno vezan za posebnu vrstu anotiranih korpusa koji se nazivaju **banke (sintaksičkih) stabala** (eng. **treebank**)[29]. Kao što sam naziv sugeriše, banka stabala je skup sintaksički analiziranih rečenica, pri čemu je zapis drveta sintaksičke analize uobičajena forma koju banka stabala koristi za predstavljanje svojih elemenata.

Namena banaka stabala je višestruka. S jedne strane, sintaksički anotiran korpus omogućava da se preciznije zadaje upit tako što se kao parametri pretrage koriste sintaksičke jedinice (konstituenti, odnosno sintagme), kao i sintaksičke i semantičke relacije (subjekat, direktni ili indirektni objekat, agent, itd.). S druge strane, precizno anotirana banka stabala može da se koristi kao skup za treniranje i evaluaciju parsera zasnovanih na mašinskom učenju (v. odeljak Mašinsko učenje, str. 104).

Ljudski resursi su posebno bitni prilikom konstrukcije banke stabala jer je ručna sintaksička anotacija bila jedino rešenje tokom izrade prvih banaka stabala. Međutim, takvo rešenje iziskuje značajne ljudske resurse, stručnost, kao i konzistentnost. Stoga ponekad isti zadatak sintaksičke anotacije nekog teksta istovremeno obavlja nekoliko ljudi i rezultat anotacije se usvaja ukoliko je identičan kod svih angažovanih anotatora, dok konfliktne slučajeve razrešava „anotator-kontrolor višeg autoriteta".

Međutim, ručna anotacija nije isplativo rešenje kada su u pitanju veće količine teksta. Upravo stoga se postojeće banke stabala koriste kao podaci na osnovu kojih se obučavaju statistički parseri. Ideja na kojoj počiva obuka parsera je konstruisanje **probabilističke kontekstno slobodne gramatike** (eng. **Probabilistic Context Free Grammar**, skr. **PCFG**). Naime, na osnovu svakog podstabla visine 1 proi-

[29]Pojam *treebank* je najverovatnije skovao Dž. Lič ([Sampson, 2003])

zvoljnog drveta sintaksičke analize u banci stabala konstruiše se pravilo kontekstno slobodne gramatike koja se naziva još i **gramatika banke stabala** (eng. **treebank grammar**). Pravilu p gramatike banke stabala se potom pridružuje verovatnoća kao količnik ukupne učestanosti pravila p i ukupne učestanosti svih pravila gramatike banke stabala koja imaju istu levu stranu kao i pravilo p ([Carroll, 2003: 243–244]). Ukoliko posmatramo pojednostavljenu banku stabala koja sadrži samo jedno drvo sintaksičke analize (Slika 2.11) rečenice `Petar čita novu knjigu.`, na osnovu tog drveta sintaksičke analize se može konstruisati PCFG (Slika 2.13).

$$
\begin{array}{rcll}
S & \rightarrow & NP\ VP\ SENT & (1) \\
NP & \rightarrow & N & (\frac{1}{2}) \\
NP & \rightarrow & A\ NP & (\frac{1}{2}) \\
VP & \rightarrow & V\ NP & (1) \\
SENT & \rightarrow & . & (\frac{1}{3}) \\
SENT & \rightarrow & ? & (\frac{1}{3}) \\
SENT & \rightarrow & ! & (\frac{1}{3}) \\
N & \rightarrow & \text{Petar} & (\frac{1}{2}) \\
N & \rightarrow & \text{knjigu} & (\frac{1}{2}) \\
V & \rightarrow & \text{čita} & (1)
\end{array}
$$

Slika 2.13: Primer probabilističke kontekstno slobodne gramatike konstruisane na osnovu banke stabala koja sadrži samo jedno drvo sintaksičke analize, i to za rečenicu `Petar čita novu knjigu.` (Slika 2.11). Verovatnoće pravila su date u zagradama.

Naravno, ovakav pristup ima smisla ako su sva pravila sa istom levom stranom podjednako verovatna, što ne mora odgovarati stvarnosti.

Ovako obučeni parseri, prilikom konstruisanja drveta sintaksičke analize, svakoj analizi pridružuju verovatnoću kao proizvod verovatnoća pravila koja su iskorišćena prilikom konstrukcije drveta. Parser na kraju bira kao ispravnu analizu onu kojoj odgovara drvo sintaksičke analize sa najvećom pridruženom verovatnoćom. Na taj način se automatski može obraditi veća količina teksta, ali i količina grešaka u rezultatu nije zanemarljiva i uvećava se u skladu sa složenošću primenjene anotacije ([Nivre, 2008]233). Stoga se posle automatskog parsiranja ponovo pribegava ručnoj korekciji grešaka, ali to je isplativo na manjim uzorcima teksta, tako da se često do-

bije banka stabala u kojoj su izmešani ručno ispravljeni delovi i delovi koji su prošli isključivo automatsku obradu. Upravo ovakav pristup je primenjen u slučaju banke stabala za engleski **Penn Treebank** ([Taylor et al., 2003], Hindle1983a) i **Praške banke stabala zavisnosti** (eng. **Prague Dependency Treebank**) ([Böhmová et al., 2003; Collins et al., 1999]).

O anotacionim standardima primenjenim u bankama (sintaksičkih) stabala biće više reči u odeljku 2.4.

Plitko parsiranje

Za pojedine zadatke automatske obrade teksta (poput ekstrakcije informacija[30]) koji zahtevaju prethodno parsiranje teksta kao jednu od faza prethodne obrade, dovoljno je da se u tekstu prepoznaju samo konstituenti poput imeničkih ili glagolskih sintagmi, a ne i njihova struktura, a ponekad ni njihova uloga u rečenici. U tom slučaju se govori o **plitkom ili delimičnom parsiranju** (eng. **shallow parsing, partial parsing**), za razliku od **dubokog parsiranja** (eng. **deep parsing**).

Plitko parsiranje može biti i uvod u naknadno duboko parsiranje, pri čemu se rad plitkog parsera završava prepoznavanjem sintagmi ili pak pridruživanjem gramatičkih funkcija određenim prepoznatim sintagmama.

Za razliku od dubokih parsera koji koriste formalizme sa manje ograničenja (gramatike zavisnosti, probabilističke ili neprobabilističke kontekstno slobodne gramatike), plitki parseri se najčešće implementiraju kao kaskade konačnih automata/transduktora ([Jurafsky & Martin, 2008: 383]). Na taj način parser donekle gubi na mogućnosti prepoznavanja određenih rekurzivnih konstrukcija, ali zato njegova implementacija dobija na efikasnosti u smislu vremenske i prostorne složenosti primenjenih algoritama prepoznavanja.

Dva tipična primera plitkog parsiranja, implementirana preko kaskada konačnih transduktora, su komadanje i prepoznavanje imenovanih entiteta. Kaskada transduktora je lista transduktora koji se određenim redosledom primenjuju na tekst, tako da izlaz svakog prethodnog transduktora postaje ulaz narednog transduktora iz kaskade.

[30] O zadacima ekstrakcije informacija videti, na primer, [Grishman, 2003].

Komadanje (eng. **chunking**) je najjednostavniji primer plitkog parsiranja([Abney, 1991, 1996]). Naziv za ovu vrstu plitkog parsiranja dolazi od izraza **komad** (eng. **chunk**) koji se koristi da označi „nerekurzivna jezgra ‚glavnih' fraza, tj. NP, VP, PP, AP, AdvP" ([Abney, 1996]). Preciznije ([Busemann, 2012]):

- komadi su delovi rečenice koji se međusobno ne preklapaju;
- komadi ne sadrže jedni druge (nerekurzivni su);
- komadi ne obuhvataju sve reči u rečenici;
- komadi nisu konstituenti, ali obično predstavljaju podnizove reči u okviru konstituenata.

Komadi su najčešće jednostavne sintagme (imenske, glagolske, priloške, itd.).

Prepoznavanje i klasifikacija imenovanih entiteta (eng. **Named Entity Recognition and Classification**, skr. **NERC**) se obično ubraja u zadatke semantičke analize, ali je tokom poslednje decenije pristup rešavanju tog problema pomeren od semantičke analize ka (plitkom) parsiranju.

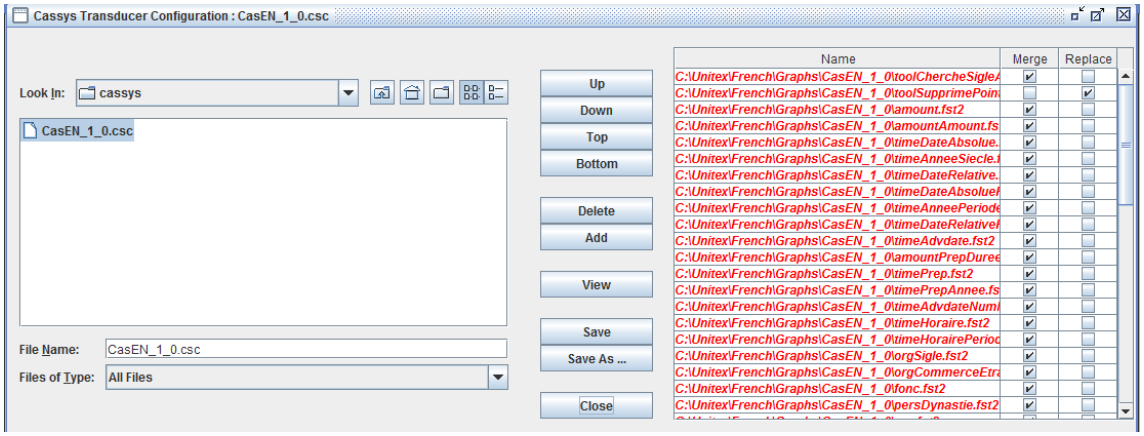
Pod **imenovanim entitetima** (eng. **Named Entities**, skr. **NE**)[31] podrazumevaju se (Chinchor et al. [1999]):

- **imena entiteta** (eng. **Entity Names**), na primer imena osoba, organizacija i lokacija);
- **vremenski izrazi** (eng. **temporal expressions**), na primer datumi, vreme na časovniku, vremenski intervali;
- **brojčani izrazi** (eng. **number expressions**) ili **količine** (eng. **quantities**) koje sačinjavaju procentualni i novčani izrazi, izrazi za mere i kardinalni brojevi[32].

[31] Ovde je navedena jedna od prvih definicija imenovanih entiteta nastala tokom poslednje dve **Konferencije o razumevanju poruka** (eng. **Message Understanding Conferences**, skr. **MUC**), MUC-6 (Chinchor [1995]) i MUC-7 (Chinchor & Marsh [1998]), koja je razmatrala samo tri kategorije imenovanih entiteta (imena, vremenski izrazi, brojčani izrazi). Kasnije su predložene finije hijerarhije imenovanih entiteta sa daleko više kategorija (na primer, 200 kategorija je opisano u [Sekine & Nobata, 2004]), a jedna od realizovanih taksonomija je Prolex, višejezična baza podataka koja obuhvata rečnik vlastitih imena i relacije između njih, definisane na više nivoa ([Maurel, 2008]).

[32] Brojevi u zapisu vremenskih izraza i količina mogu biti i u numeričkom i u alfabetskom obliku, tj. mogu biti zapisani samo pomoću cifara, ili izraženi rečima, ili kombinovanjem i cifara i reči.

Primenu plitkog parsiranja u prepoznavanju imenovanih entiteta ilustrovaćemo na primeru sistema CasSys ([Friburger & Maurel, 2004], [Antoine et al., 2008]). CasSys je najpre razvijen kao samostalni sistem namenjen isključivo za ekstrakciju imenovanih entiteta ([Friburger, 2002]), zasnovan na primeni kaskade konačnih transduktora. U međuvremenu je sistem uopšten i sastavni je deo programskog paketa Unitex ([Paumier, 2011], poglavlje 11) počev od verzije 3.0. Uopštenje sistema CasSys podrazumeva mogućnost da korisnik definiše i ažurira listu proizvoljnih konačnih transduktora čija primena nije ograničena samo na ekstrakciju imenovanih entiteta. Transduktori se potom primenjuju na ulazni tekst u redosledu kojim su navedeni u listi, pri čemu svaki pojedinačni transduktor modifikuje ulazni tekst umetanjem dodatnog teksta ili zamenom nekog dela ulaznog teksta novim tekstom. Grafičko sučelje sistema CasSys (Slika 2.14) omogućava jednostavno održavanje liste transduktora (dodavanje ili brisanje transduktora iz liste, privremeno isključivanje transduktora iz liste, pregled i eventualno ažuriranje lokalne gramatike na osnovu koje je generisan transduktor, promena redosleda izvršavanja transduktora, izbor da li će se izlaz transduktora umetnuti u ulazni tekst ili će zameniti neki postojeći deo ulaznog teksta, da li će se transduktor rekurzivno primenjivati sve dok ima šta da prepozna, itd.).



Slika 2.14: Grafičko sučelje sistema CasSys u okviru programskog paketa Unitex 3.0.

Korisnik ne definiše konačne transduktore direktno, već konstruiše lokalne gramatike u formi grafova, a kompilacijom lokalne gramatike se proizvodi odgovarajući konačni transduktor. **Lokalne gramatike** (eng. **local grammars**) predstavljaju

alternativni pristup sintaksičkoj analizi rečenica prirodnog jezika, sa ciljem da se izbegne formalizam leksikalizovanih kontekstno slobodnih gramatika ogromnih razmera. Naime, formalizam kontekstno slobodnih gramatika pokušava da izgradi uopštenu gramatiku jezika, što za posledicu ima ogroman broj pravila čija desna strana sadrži konkretne reči jezika (otud i naziv *leksikalizovana gramatika*). Konačni automati i transduktori, kao pogodniji formalizmi za implementaciju, ne mogu[33] se iskoristiti za konstrukciju uopštene gramatike jer ne opisuju sve tipove rekurzije (na primer, pravila sa rekurzijom u sredini oblika $A \rightarrow \alpha A \beta$, gde su α i β neprazne niske), već se za zadatu kontekstno slobodnu gramatiku može konstruisati samo konačni automat koji aproksimira jezik generisan tom gramatikom, tj. jezik automata-aproksimacije je nadskup ili podskup jezika polazne gramatike ([Nederhof, 2000]).

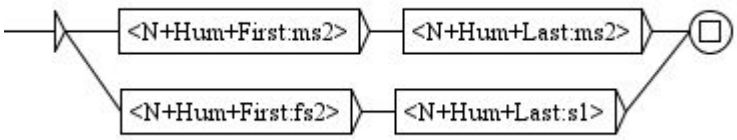
Moris Gros, tvorac lokalnih gramatika, sintaksičku analizu zasniva na konačnim automatima, ali umesto opisa globalne sintakse jezika, cilj je opis određenih lokalnih uslova i ograničenja koje treba da zadovolje susedne niske u prihvatljivim jezičkim iskazima ([Gross, 1993, 1997]). Prilikom kreiranja lokalnih gramatika najpre se opisuju konstrukcije koje se često pojavljuju u tekstu (na primer, imenovani entiteti), tako da se te gramatike mogu ponovo iskoristiti za opis složenijih sintaksičkih konstrukcija. Jedna od prednosti ovog pristupa je što se lokalne gramatike koje opisuju različite sintaksičke fenomene mogu nezavisno konstruisati, održavati i primenjivati.

Druga prednost lokalnih gramatika je mogućnost da se one primenjuju istovremeno sa elektronskim rečnicima čije su odrednice morfosintaksički obeležene.

Slika 2.15 ilustruje jednostavnu lokalnu gramatiku kreiranu u sistemu Unitex za obradu teksta na srpskom jeziku. Gramatika je realizovana kao usmereni graf pri čemu se podrazumeva da su prvi čvor sleva i prvi čvor zdesna početno i završno stanje odgovarajućeg transduktora tim redom. Ostala stanja transduktora su predstavljena

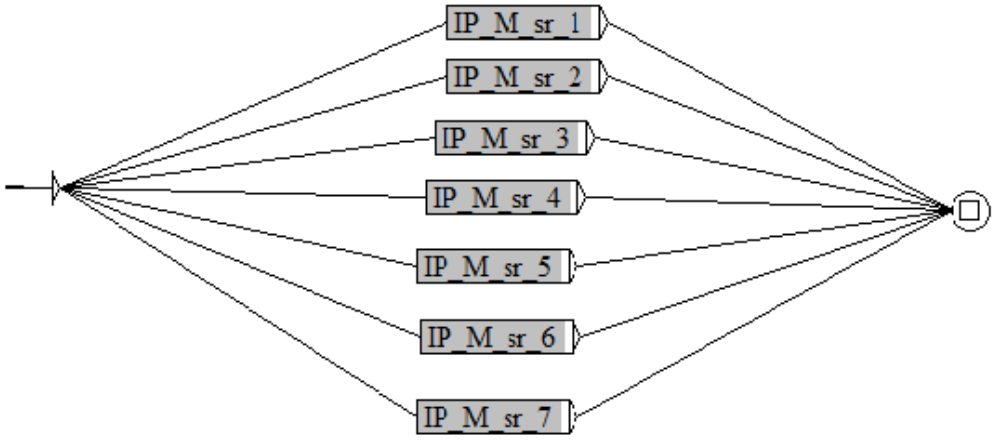
[33]Postoje i suprotna mišljenja koja tvrde da je skup rečenica prirodnog jezika koje su zaista u upotrebi konačan, pa time ne samo kontekstno slobodan, već i regularan, što teorijski omogućava njegov opis konačnim automatom. Ovakav pristup je, pre svega, okrenut ka praktičnoj primeni sintaksičke analize i odbacuje proizvode teorijskih razmatranja poput rekurzivno generisanih rečenica u okviru kojih se proizvoljan broj puta ugnježđuju rečenice određene strukture, čime se odbacuju i glavni argumenti u dokazima da prirodni jezik nije ni regularan ni kontekstno slobodan. Međutim, praktična realizacija konačnog automata koji bi opisao taj „efektivni deo" prirodnog jezika je previše zahtevna i neisplativa, tako da je sa računarske tačke gledišta besmislena iz istih razloga kao i konstrukcija uopštene leksikalizovane kontekstno slobodne gramatike.

neoznačenim granama grafa, dok čvorovi grafa, označeni regularnim izrazima, predstavljaju prelaze iz jednog stanja u drugo. Prelaz iz stanja a u stanje b, obeležen regularnim izrazom r, realizuje se ukoliko je transduktor u stanju a prepoznao na ulazu nisku opisanu regularnim izrazom r. Upotrebljeni regularni izrazi predstavljaju morfosintaksičke opise odrednica iz elektronskog rečnika, i to imenice (`N`) koje označavaju ljudsko biće (`+Hum`). Preciznije, gornji put u grafu (`<N+Hum+First:ms2>` `<N+Hum+Last:ms2>`) označava muško (`m`) ime (`+First`) i prezime (`+Last`) u genitivu (`2`) jednine (`s`), dok donji put u grafu (`<N+Hum+First:fs2>` `<N+Hum+Last:s1>`) prepoznaje žensko (`f`) ime u genitivu jednine i prezime u nominativu jednine (u prvom slučaju postoji slaganje imena i prezimena u rodu, broju i padežu, dok u drugom slučaju postoji samo slaganje u broju, pošto se žensko prezime ne menja po padežima). Analogno se data lokalna gramatika može proširiti tako da prepoznaje muško ili žensko ime i prezime u svim padežima, kao i varijante kada se prezime pojavljuje pre imena, odnosno varijante sa dva prezimena razdvojena razmakom ili crticom. Međutim, u praksi se obično, umesto kreiranja jedne složene lokalne gramatike, kreira više jednostavnih lokalnih gramatika koje se potom mogu uključiti kao podgrafovi različitih grafova. Konkretno, za svaki padež može da se konstruiše lokalna gramatika m_i $(i = 1, \ldots, 7)$ koja prepoznaje sve varijante muškog imena i prezimena u tom padežu, a potom i lokalna gramatika koja sadrži sve gramatike m_i kao podgrafove (Slika 2.16).



Slika 2.15: Lokalna gramatika koja prepoznaje muško ime i prezime u genitivu jednine ili žensko ime i prezime u genitivu jednine.

Složeniji primeri prepoznavanja imenovanih entiteta u tekstovima na srpskom jeziku pomoću kaskada konačnih transduktora mogu se naći u [Krstev, Vitas & Gucul, 2005], [Gucul-Milojević et al., 2008], [Krstev et al., 2011], [Krstev et al., 2012].



Slika 2.16: Lokalna gramatika koja prepoznaje muško ime i prezime u svim padežima. Osenčeni čvorovi grafa predstavljaju podgrafove, tj. nazive datoteke u kojima se nalaze lokalne gramatike koje prepoznaju muško ime i prezime u jednom određenom padežu jednine.

NERC je zadatak čija detaljna analiza izlazi iz okvira ovog rada, a više o tome se može naći, na primer, u [Nadeau & Sekine, 2009].

2.4 Anotacija korpusa

U odeljku 1.3 (pododeljak Anotacija, str. 27) anotacija je definisana kao postupak kojim se dodatne informacije (lingvističke i nelingvističke) pridružuju delovima korpusa (tekstovima, logičkim celinama u okviru teksta, tokenima).

Da bi se uopšte pristupilo anotaciji korpusa, prethodno je neophodno:

(i) precizirati na kom nivou će se anotirati korpus (morfološki nivo, sintaksički nivo, itd.), kao i koje konkretne informacije će tom prilikom biti pridružene odgovarajućim delovima korpusa (lema, vrsta reči, vrednosti flektivnih kategorija, itd.);

(ii) definisati konkretne oznake informacija koje će se pridružiti delovima korpusa;

(iii) doneti odluku da li će se anotacija sprovoditi ručno, automatski ili kombinovanjem ručne i automatske anotacije (poluautomatska anotacija);

(iv) izabrati da li će anotacija biti direktno ugrađena u tekst korpusa ili fizički odvojena u posebnoj datoteci;

(v) obezbediti odgovarajuće programske alate za anotaciju.

Fizičke realizacije anotacije korpusa

Ručna, automatska i poluatomatska anotacija korpusa

U praksi se najčešće primenjuje poluautomatska anotacija korpusa, tj. kombinuju se ručna i automatska anotacija, kao pokušaj da se napravi ravnoteža između utrošenih sredstava (vreme, novac, ljudski resursi) i kvaliteta anotacije korpusa (relativna učestanost grešaka u odnosu na veličinu korpusa).

Ručna anotacija je svakako najskuplja:

- potrebno je vreme da se obuče ljudski resursi koji će obaviti anotaciju;
- ponekad, radi kontrole obavljene anotacije, isti posao obavlja dva ili više anotatora koji ne znaju jedni za druge, a u slučaju da različito anotiraju neki deo korpusa, neophodan je anotator-medijator koji će razrešiti kako treba da izgleda konačna anotacija;
- neophodan je novac za plate ručnih anotatora;
- u poređenju sa automatskom anotacijom, ručna anotacija je daleko sporija.

Automatska anotacija je daleko jeftinija od ručne, ali u slučaju kada programski alat koji vrši anotaciju mora da izabere između više ponuđenih mogućnosti, uprkos „prethodnom iskustvu" zasnovanom na nekoj od tehnika mašinskog učenja, automatska anotacija proizvodi greške sa određenom relativnom učestanošću u odnosu na ukupnu veličinu korpusa. Jasno je da sa porastom veličine korpusa raste i broj pogrešno anotiranih delova korpusa, a jedino se ručno takve greške mogu otkriti i ispraviti.

Poluautomatska anotacija se obično realizuje tako što se najpre ručno anotira neki manji deo korpusa, **skup za treniranje** (eng. **training set, training corpus**), na osnovu kojeg se obučava **programski alat za automatsku anotaciju** (eng. **tagger**). Program za anotaciju potom anotira neanotirani deo korpusa, skup za **testiranje** (eng. **testing set**), čija je veličina približna skupu za treniranje, posle čega se ručno ispravlja skup za testiranje i spaja sa prvobitnim skupom za treniranje. Program za anotaciju se ponovo obučava, koristeći novi skup za treniranje, i postupak se iterativno ponavlja, pri čemu se svaki put vrši evaluacija (v. pododeljak programa za anotaciju Evaluacija). Kontinuirano uvećanje skupa za treniranje i evaluacija programa za automatsku anotaciju imaju smisla sve dok se ne postigne željena preciznost anotacije i dok su troškovi ručne anotacije u okviru planiranih vrednosti.

Skup za treniranje se mora pažljivo konstruisati tako da predstavlja „korpus u malom". U protivnom, ako je skup za treniranje ograničen u pogledu žanra tekstova ili je konstruisan sa ciljem da se anotacija iskoristi u daljoj obradi za neku specifičnu primenu u ograničenom domenu, obuka programa za anotaciju na takvom skupu za treniranje ne može dati zadovoljavajuće rezultate na opštem korpusu sa žanrovima i domenima koji su izostali iz skupa za treniranje.

Evaluacija

Evaluacija programa za automatsku anotaciju korpusa se sprovodi korišćenjem skupa za treniranje, tj. prethodno ručno anotiranih tekstova za korpus. Najjednostavnija evaluacija se zasniva na podeli ručno anotiranih podataka (polazni skup za treniranje) na tri dela u odnosu 8:1:1, pri čemu se najveći deo koristi za obuku programa za anotaciju, drugi deo — za testiranje i fino podešavanje parametara programa za anotaciju, dok treći deo ima ulogu skupa za testiranje ([Jurafsky & Martin, 2008: 153]). Drugi deo se još naziva **skup za razvoj i testiranje** (eng. **development test set**, skr. **dev-test set**). Drugi i treći deo se automatski anotiraju korišćenjem programa za anotaciju, a onda se vrši upoređivanje polazne ručne anotacije i dobijene automatske anotacije. Preciznost anotacije skupa X, gde je X

skup za razvoj i testiranje ili skup za testiranje, računa se kao količnik

$$p = \frac{m}{n} \tag{2.2}$$

gde je m broja tokena skupa X koji su istovetno anotirani ručnim i automatskim postupkom, dok je n ukupan broj tokena skupa X.

Skup za razvoj i testiranje se uvodi kako bi se izbeglo da se konačni skup za testiranje koristi za fino podešavanje parametara programa za automatsku anotaciju, a time i „nameštanje" povoljnije ocene preciznosti. Treći deo se koristi kao pravi skup za testiranje, u smislu da je potpuno „nepoznat" programu za automatsku anotaciju, pa je i ocena greške realističnija. Ukoliko se tokom testiranja programa za anotaciju ne eksperimentiše sa različitim vrednostima parametara kako bi se dobio što bolja ocena preciznosti, može se koristiti i podela u odnosu 9:1, tj. izostaviti skup za razvoj i testiranje.

U praksi se evaluacija češće realizuje postupkom **unakrsnog testa** (eng. **cross-validation test**). Naime, glavni nedostatak izloženog postupka „jednostavne evaluacije" je što su skupovi za treniranje najčešće nereprezentativni, prevashodno zbog svoje nedovoljne veličine. Stoga, umesto jedne podele na skup za treniranje, skup za razvoj i testiranje i skup za testiranje u odnosu 8:1:1 (odnosno, na skup za treniranje i skup za testiranje u odnosu 9:1), prethodno ručno anotirani podaci se opisanim postupkom, na slučajan način, dele više puta, za svaku podelu se računa preciznost programa za anotaciju (jednačinom 2.2), a na kraju se aritmetička sredina dobijenih vrednosti uzima za prosečnu preciznost programa za anotaciju. Pored prosečne vrednosti preciznosti anotacije, kao dodatni rezultati evaluacije se još izračunavaju maksimalna i minimalna preciznost, kao i standardna devijacija preciznosti anotacije. Broj sprovedenih podela je obično 10, zbog čega se ceo postupak još naziva **desetostruki unakrsni test** (eng. **10-fold cross-validation test**).

Ugrađena i odvojena anotacija korpusa

Anotacija može biti direktno ugrađena u tekst korpusa ili fizički odvojena u posebnoj datoteci.

Ugrađena anotacija (eng. **embedded annotation, inline annotation**) deluje kao prirodno rešenje, pogotovo kada se realizuje pomoću nekog od jezika za označavanje zasnovanog na XML-u, jer je XML upravo i zamišljen kao mešavina teksta i oznaka (etiketa). Međutim, problemi nastaju pri pokušaju da se isti tekst anotira korišćenjem različitih anotacija (na primer, na morfološkom i sintaksičkom nivou), koje se međusobno preklapaju na nedozvoljen način (na primer, XML-elementi jedne anotacije se mogu preklopiti sa XML-elementima druge anotacije, te tako dobijeni dokument više nije dobro formirani XML). Slučajevi nedozvoljenih preklapanja XML-anotacija mogu se na razne načine prevazići u okviru ugrađene anotacije ([DeRose, 2004]), ali i primenom odvojene anotacije.

Odvojena anotacija (eng. **stand-off annotation, out-of-line annotation**) ima nekoliko prednosti u odnosu na ugrađenu anotaciju ([Bański, 2010]):

- tekstu korpusa se može pridružiti više različitih anotacija, svaka u posebnoj datoteci, koje ne moraju istovremeno da se koriste;
- tekst može istovremeno da se anotira različitim anotacijama koje zajedno ne bi mogle da se ugrade u istu datoteku zbog nedozvoljenih preklapanja;
- tekst ima status „samo za čitanje", čime se, bez obzira na pridružene anotacije, omogućava da uvek zadrži formu koju je imao i pre anotacije;
- isključivanjem slojeva anotacije koji u datom trenutku nisu od značaja, smanjuje se vremenska složenost prilikom automatske obrade anotiranog korpusa.

Odvojenoj anotaciji se obično zamera potreba za posebnim programskim alatima koji mogu da povežu fizički razdvojeni tekst i odgovarajuću anotaciju. Ukoliko je u pitanju XML-anotacija, uobičajeni mehanizam povezivanja su pokazivači (XPointer ([DeRose et al., 2002]) koje podržava mali broj popularnih XML-alata.

Pregled aktuelnih alata koji omogućavaju rad sa ugrađenom i odvojenom anotacijom korpusa može se naći, na primer, u [Ahn et al., 2006] i [Wilcock, 2009].

Kao alternativa ugrađenoj i odvojenoj anotaciji se nudi **multiplicirani anotirani tekst** (eng. **multiply-annotated text**) koji koristi redundantnu slojevitu

anotaciju, tj. svaki sloj anotacije sadrži istovetnu kopiju izvornog teksta ([Rehm et al., 2010]).

Opšta načela anotacije korpusa

Lič navodi sledeće opšte principe kojima se treba rukovoditi prilikom anotacije korpusa ([Leech, 1993]):

- Omogućiti uklanjanje anotacije po potrebi, tako da se u svakom trenutku može rekonstruisati polazni neanotirani tekst korpusa.
- Omogućiti ekstrakciju same anotacije iz teksta.
- Shemu anotacije zasnovati na smernicama koje su na raspolaganju krajnjem korisniku.
- Jasno naznačiti ko je i kako sproveo anotaciju korpusa.
- Krajnjem korisniku jasno staviti do znanja da anotacija korpusa nije bez grešaka, ali da može da se iskoristi za analizu korpusa.
- Sheme anotacije zasnovati na principima oko kojih je postignuta najširа saglasnost i koji su neutralni u odnosu na različite teorije.
- Nijedna shema anotacije nema *a priori* pravo da se proglasi standardom.

Standardi za anotaciju korpusa

Postoji više standarda koji se primenjuju prilikom anotacije korpusa, ali još uvek nisu retki ni slučajevi da kreatori korpusa ignorišu standarde i izmišljaju svoje formate za predstavljanje anotiranog teksta. Osim nepoznavanja postojećih standarda za anotaciju, jedan od uzroka za njihovo ignorisanje je i činjenica da postoji mnoštvo lingvističkih teorija sa raznovrsnim konceptima, a da istovremeno ne postoji jedan standard koji bi se uspešno primenio na svaku od njih. Takođe, prihvatanje standarda koji su vezani ne samo za anotaciju, već za računarstvo uopšte, ne zavisi samo od kvaliteta njihove specifikacije (dokumentacije, priručnika), već i od raspoloživosti

programskih alata koji će pomoći korisniku u konkretnoj primeni standarda. I samim početnicima koji su zainteresovani da primene standarde u kreiranju, obradi i pretrazi korpusa lakše je da se drže nekog široko prihvaćenog standarda jer će, pored dokumentacije i alata, na raspolaganju imati i iskustva ranijih korisnika u vidu najčešćih pitanja i odgovora na elektronskim diskusionim grupama (eng. forum), elektronskim dopisnim listama (eng. mailing list) i društvenim mrežama.

Još uvek ne postoji jedan opšteprihvaćeni standard koji se primenjuje prilikom anotacije korpusa. Ovde će biti predstavljeni neki od trenutno najzastupljenijih „nezvaničnih standarda" koje koristi zajednica korpusnih lingvista. Ono što je zajedničko za sve njih je da u osnovi koriste iste, opšte standarde za kodiranje teksta (ISO-8859, Unicode) i za označavanje teksta (XML). O standardima za kodiranje teksta je već bilo reči u odeljku 2.2, tako da ovde sledi opis samo opštih i specifičnih standarda za označavanje teksta, odnosno anotaciju korpusa.

XML

Tokom 1969. godine počeo je rad na razvoju **Generalizovanog jezika za označavanje** (eng. **Generalized Markup Language**, skr. **GML**) kako bi se poboljšalo upravljanje složenim industrijskim elektronskim dokumentima u okviru korporacije IBM. GML je omogućio zapis strukture elektronskog teksta na način koji je nezavisan od potonje obrade tog teksta, odnosno razdvojio je strukturu dokumenta (na primer, informacije o tome da deo teksta predstavlja naslov, pasus, sliku, listu, itd.) i njegovu vizuelnu reprezentaciju, tj. formatiranje (na primer, veličina i težina fonta za prikaz naslova i pasusa, tip poravnanja margina, itd.). Dodatne prednosti pristupa koje nudi GML su mogućnosti:

- da se isti delovi dokumenta ponovo iskoriste u drugom kontekstu;
- da se na delove dokumenta referiše zahvaljujući sistemu jedinstvenih identifikatora;
- da se delovi dokumenta opciono uključuju ili isključuju iz dokumenta;

Stoga se najpre ANSI zainteresovao za GML (1978. godine), a potom i ISO, tako da su tokom osamdesetih godina XX veka nastale dve verzije standarda, ISO

8879:1986 i ISO 8879:1988, kojima je definisan **Standardni generalizovani jezik za označavanje** (eng. **Standard Generalized Markup Language**, skr. **SGML**), kao uopštenje GML-a. Iako se kaže da je SGML jezik za označavanje, on je zapravo *meta-jezik* jer omogućava definisanje konkretnih specifičnih jezika za označavanje u skladu sa određenim pravilima. Ono što je zajedničko svim jezicima definisanim na osnovu standarda SGML je mogućnost kreiranja tekstuelnih dokumenata (SGML-dokumenata) u kojima su međusobno izmešani **sadržina** (eng. **content**) i njeno **označavanje** (eng. **markup**).

SGML je stekao svoju popularnost, pre svega, zbog podrške raznih programskih alata (uređivači teksta, programi namenjeni pripremi za štampu, parseri, sistemi za upravljanje dokumentima, itd.) i vodećih programskih jezika koji su uspešno prepoznali navedene pogodnosti.

Međutim, početkom devedesetih godina XX veka SGML gubi na popularnosti, pre svega zbog pokušaja da bude i sveobuhvatan i dovoljno fleksibilan, što je, s jedne strane, odbijalo početnike, a sa druge strane otežavalo njegovu implementaciju.

Popularnost HTML-a, jednog od konkretnih jezika za označavanje zasnovanih na SGML-u, kao i ekspanzija interneta, sprečili su da SGML završi u zapećku, ali su se istovremeno pojavili zahtevi da se SGML pojednostavi i iskoristi kao format za razmenu podataka na internetu. Tako je 1996. godine nastao **proširiv jezik za označavanje** (eng. **eXtensible Markup Language**, skr. **XML**) sa idejom da zameni SGML, u čemu je u potpunosti i uspeo. Njegov tekstuelni format, podjednako čitljiv i za čoveka i za računar, sa podrškom za Unicode, zahvaljujući svojoj jednostavnosti i opštosti, postao je jedna od najpopularnijih internet-tehnologija.

XML realizuje označavanje preko **etiketa** (eng. **tag**) i entiteta. Etikete se prepoznaju kao delovi teksta između simbola `<` i `>` i mogu biti:

- početne ili otvorene (na primer `<div>`),
- završne ili zatvorene (na primer `</div>`),

Entiteti su delovi teksta između simbola `&` i prvog sledećeg simbola `;` i mogu predstavljati:

- jedan Unicode-karakter (karakterski entiteti),

- nisku proizvoljne dužine (opšti unutrašnji entiteti),
- ili sadržaj neke tekstuelne datoteke (opšti spoljašnji entiteti).

Za karaktere koji u okviru XML-dokumenta imaju specijalno značenje, koristi se sledećih pet entiteta:

- < umesto <
- > umesto >
- & umesto &
- ' umesto apostrofa '
- " umesto navodnika "

Standard XML ([Bray et al., 2008]) definiše opšta pravila koja mora da zadovolji **dobro formiran** (eng. **well-formed**) XML-dokument, koja ćemo razmotriti na primeru (Slika 2.17). Svaki dobro formirani XML-dokument se sastoji iz prologa (linija 1) i jednog elementa (linije 3–15) koji se naziva **koreni element** (eng. **root element**), a koji sadrži, neposredno ili posredno, sve ostale elemente u dokumentu. Pored toga, u dokumentu su mogu pojaviti i komentari (linija 2) koji se navode između niski `<!--` i `-->`.

Prolog na početku sadrži obaveznu XML-deklaraciju koja predstavlja jednu od mogućih **instrukcija obrade** (eng. **processing instruction**) u XML-dokumentu. Instrukcije obrade se navode između niski `<?` i `?>`, a na početku se instrukcije se navodi njena *„meta"* ili *cilj* (eng. *target*). „Meta" identifikuje odgovarajući (ciljani) program koji ignoriše preostale instrukcije obrade čije „mete" ne prepoznaje. XML-deklaracija koristi „metu" `xml`, tj. navodi se između `<?xml` i `?>`).

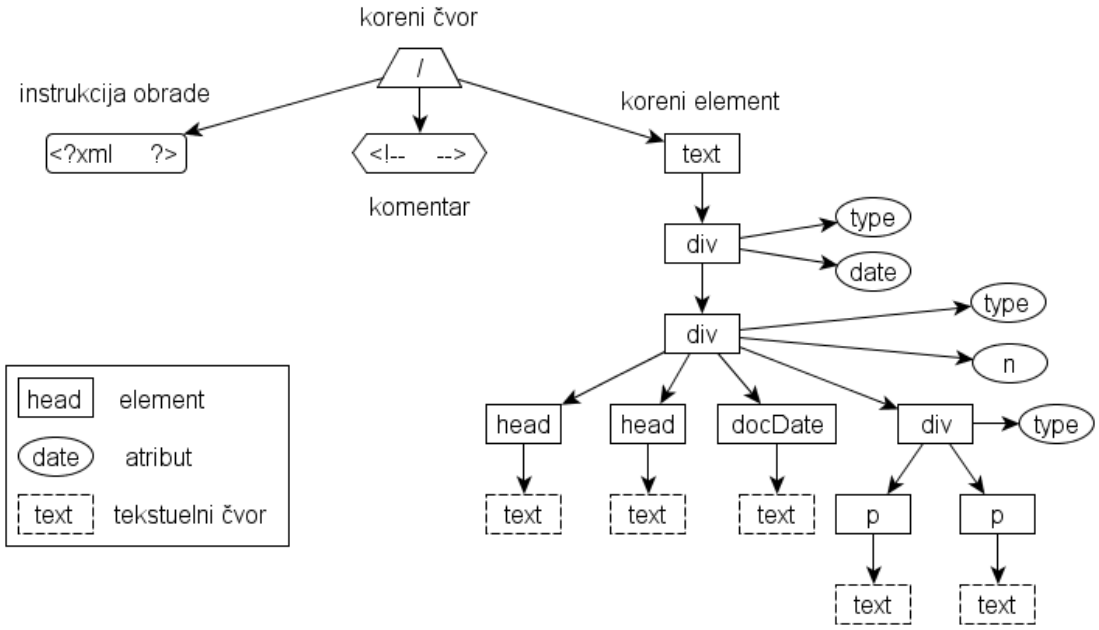
Svaki element je omeđen početnom i odgovarajućom završnom etiketom i nosi isti naziv kao i njegove etikete. U datom primeru koreni element ima naziv `text` i omeđen je etiketama `<text>` i `</text>`. Pored elementa `text`, u dokumentu se pojavljuju još tri elementa `div`, dva elementa `head`, element `docDate` i dva elementa `p`. Da bi se razumeo odnos između ovih elemenata, treba napomenuti da se svaki dobro formirani XML-dokument može predstaviti u obliku drveta koje podseća na

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- novinski članak iz sportske rubrike -->
3  <text>
4    <div type="issue" date="02022009">
5      <div type="article" n="1">
6        <head>Nadal spreman za Dejvis kup</head>
7        <head>Prvi teniser sveta Španac Rafael Nadal
             izjavio je da se posle trijumfa na Australijan
             Openu okreće mečevima u Dejvis kupu protiv
             reprezentacije Srbije.</head>
8        <docDate>Beograd, 2. februar</docDate>
9        <div type="section">
10         <p>"Moj cilj je Dejvis kup, u kojem zbog povrede
               nisam igrao prošle godine", rekao je Nadal.
11         </p>
12         <p>"To je više san nego cilj. Ne želim sada da
               stanem", dodao je on. Nadalova pobeda u
               Melburnu je treća uzastopna nad Federerom u
               finalima velikih turnira.
13         </p>
14         </div>
15     </div>
16   </div>
17 </text>
```

Slika 2.17: Primer dobro formiranog XML-dokumenta

porodično stablo (Slika 2.18). Koreni čvor drveta, koga ne treba mešati sa korenim elementom dokumenta, predstavlja zapravo sam dokument. U datom primeru koreni čvor sadrži XML-deklaraciju, komentar i koreni element `text`. Element `text` sadrži jedan element `div` (Slika 2.17, linije 4–16) koji takođe sadrži jedan element `div` (Slika2.17, linije 5–15), pri čemu se još kaže da je prvi element `div` dete elementa `text`, kao što je drugi element `div` dete prvog elementa `div`. Drugi element `div` ima četvoro dece: dva elementa `head`, element `docDate` i treći element `div` (Slika 2.17, linije 9–14).

Sadržaji elemenata ne smeju da se preklapaju, jedino je dozvoljeno da jedan element bude unutar drugog. U praksi to znači da poslednja otvorena etiketa mora prva da se zatvori, odnosno da prva otvorena etiketa mora poslednja da se zatvori.



Slika 2.18: Dobro formirani XML-dokument (Slika 2.17) predstavljen kao drvo.

Elementi `head`, `docDate` i `p` u navedenom primeru ne sadrže druge elementa kao decu, već isključivo tekst. Tekstuelna sadržina svakog od tih elemenata ponaosob predstavlja jedan **tekstuelni čvor** (eng. **text node**) koji je ujedno i dete odgovarajućeg elementa. Tekst može sadržati i entitete koji se interpretiraju tokom analize (parsiranja) XML-dokumenta (na primer, `&quot;` se interpretira kao `"`), pa se stoga tekstuelni čvorovi još nazivaju i **parsirani karakterski podaci** (eng. **Parsed Character DATA**, skr. **#PCDATA**).

Posebnu vrstu elemenata predstavljaju **prazni elementi** (eng. **empty element**), tj. elementi koji nemaju nikakav sadržaj. Radi jednostavnije forme dokumenta, ako je, na primer, `el` prazan element, umesto `<el></el>` može se pisati prosto `<el/>`[34].

Elementi `div` sadrže i atribute (`type`, `date`, `n`) koji se navode u okviru otvorene etikete u formatu `imeAtributa=vrednost`, pri čemu se vrednost atributa uvek piše između jednostrukih ili dvostrukih znakova navoda (apostrofa ili navodnika). Pojedinačni atributi elemenata predstavljaju **atributske čvorove** (eng. **attribute**

[34]Na primer, `<datum dan='9' mesec='5' godina='2012'/>`.

node).

U okviru XML-deklaracije se koriste tri atributa: `version`, `encoding`, `standalone`[35]. Obavezni atribut `version` definiše verziju XML-a koju koristi dokument i ta vrednost je najčešće 1.0[36]. Opcioni atribut `encoding` precizira kodnu shemu koju koristi XML-dokument (u datom primeru u pitanju je UTF-8 što je i podrazumevana kodna shema ukoliko se `encoding` ne navede). Opcioni atribut `standalone` ima dve moguće vrednosti, „yes" i „no", koje određuju da li je XML-dokument samostalan ili ne (detaljnije objašnjenje sledi u istom odeljku).

Konkretni jezici za označavanje, pored opštih uslova koje mora da zadovolji dobro formirani XML-dokument, nameću i svoje specifične uslove. Definicija konkretnog jezika za označavanje podrazumeva:

- preciziranje **vokabulara** (eng. **vocabular**), tj. skupa raspoloživih etiketa i entiteta, kao i
- utvrđivanje **gramatike** (eng. **grammar**), tj. pravila po kojima se različite etikete mogu kombinovati u dokumentu.

Objedinjeni opis vokabulara i gramatike se naziva **XML-šema** (eng. **XML schema**). Još je SGML omogućio zadavanje šeme pomoću **definicije tipa dokumenta** (eng. **Document Type Definition**, skr. **DTD**), a isti mehanizam definiše i specifikacija XML-a ([Bray et al., 2008]). Međutim, DTD nije dovoljno izražajan prilikom opisa sadržaja tekstuelnih čvorova, jer sve te sadržaje tretira na isti način kao parsirane karakterske podatke. Stoga su, pored DTD-a, razvijeni i drugi jezici za zadavanje šeme, a najpopularniji su W3C XML Schema ([Biron & Malhotra, 2004; Fallside & Walmsley, 2004; Thompson et al., 2004]) i ISO RELAX NG ([Clark, 2001]). Ova dva jezika, imaju dve bitne prednosti u odnosu na DTD:

- njihove šeme su takođe XML-dokumenti, te se isti softver koji se koristi za obradu XML-dokumenata može koristiti i za obradu šema;

[35] Atributi instrukcija obrade, pa time i XML-deklaracije, nisu atributski čvorovi.

[36] Postoji i verzija XML 1.1, ali u ovom trenutku ona još uvek nije dovoljno softverski podržana, tako da se praktično ne koristi.

- njihove šeme su izražajnije u odnosu na DTD-šeme jer njihove specifikacije definišu različite tipove podataka (celobrojne tipove, datume, niske, itd.) i ograničenja koja se mogu nametnuti vrednostima tih tipova podataka (najmanja i najveća vrednost, obrazac u obliku regularnog izraza koji opisuje dozvoljene vrednosti niske, itd.), što se koristi za precizniji opis sadržaja tekstuelnih čvorova i vrednosti atributa.

Ako opcioni atribut `standalone` XML-deklaracije ima vrednost „yes", to znači da XML-dokument ne zavisi od neke šeme, dok podrazumevana vrednost „no" označava da takva zavisnost možda postoji.

Ukoliko dobro formirani XML-dokument x zavisi od neke šeme s i zadovoljava sva njena ograničenja, tada je XML-dokument x **validan** u odnosu na šemu s. Programi koji proveravaju validnost XML-dokumenta u odnosu na neku šemu nazivaju se **XML-validatori** ili **XML-parseri**. XML-validatori su danas sastavni deo svih naprednijih programa za uređivanje XML-dokumenata, kao i biblioteka različitih programskih jezika koje omogućavaju izradu aplikacija za obradu XML-dokumenata.

Pored navedenih prednosti koje su učinile XML jednom od najpopularnijih internet-tehnologija, treba istaći da zaslugu za to, kao i u slučaju SGML-a, imaju i mnogobrojni programski alati, kao i najuticajniji programski jezici (Java, C#, Perl, Python, itd.) koji podržavaju obradu XML-dokumenata. Tu treba posebno istaći i nove tehnologije razvijene pomoću samog XML-a, poput **Proširivog jezika stilskih listova** (eng. **Extensible Stylesheet Language**, skr. **XSL**) i **XML-upitnog jezika** (eng. **XML Query Language**, skr. **XQuery**). XSL ([Clark, 1999]) omogućava transformaciju XML-dokumenta bilo u drugi XML-dokument, ili u (X)HTML-dokument, ili pak u format čistog teksta. XQuery ([Boag et al., 2010]) se koristi za pretragu i ažuriranje kolekcija XML-dokumenata (XML-baza podataka). Zajedničko za XSL i XQuery je korišćenje jezika XPath ([Clark & DeRose, 1999]) za adresiranje delova XML-dokumenta.

Formati anotacije korpusa koji nisu zasnovani na XML-tehnologiji

Iako se XML tokom svog dosadašnjeg postojanja nametnuo kao dominantni format za anotaciju teksta, treba ipak istaći da postoji nekoliko popularnih anotacijskih

formata koji nisu zasnovani na XML-tehnologiji, kao što su horizontalni i vertikalni (vertikalizovani) format teksta. Oba formata su ilustrovana na primeru anotacije rečenice (1):

(1) Bio je vedar i hladan aprilski dan.

Horizontalni format podrazumeva uobičajenu reprezentaciju rečenice koju u tekstu koriste evropski jezici, pri čemu se uz svaki token dopisuje i niska sa odgovarajućim pridruženim informacijama. Ako bi, na primer, koristili horizontalni format da svakom tokenu pridružimo njegovu vrstu reči (ako je u pitanju korpusna reč) ili oznaku da je u pitanju interpunkcija, rezultat anotacije rečenice (1) bi bila rečenica (2)[37]:

(2) `Bio_V je_V vedar_A i_C hladan_A aprilski_A dan_N ._PUNC`

Sa stanovišta korisnika horizontalni format je kompaktan i čitljiv u slučaju kada tekst nije višestruko anotiran. Međutim, ako su tokenima pridružene informacije sa različitih nivoa (na primer, morfološkog, sintaksičkog, semantičkog, itd.) pogodnije je koristiti **vertikalni (vertikalizovani) format**. Vertikalni format koristi zapis po kolonama, obično razdvojenim tabulatorom. U prvoj koloni se navode tokeni rečenice, po jedan u svakom redu, dok se u ostalim kolonama, u odgovarajućem redu, zapisuju pridružene informacije, pri čemu svakom nivou anotacije odgovara po jedna kolona. Tako se, prilikom anotacije rečenice (1) u vertikalnom formatu (Slika 2.19), svakom tokenu (prva kolona) može pridružiti informacija o vrsti reči (druga kolona), kao i informacija o lemi (treća kolona).

Vertikalni tekst predstavlja najčešći ulazni i izlazni format alata za automatsku anotaciju korpusa (na primer, [Brants, 2000; Schmid, 1994]), kao i ulazni format za pojedine alate za indeksiranje korpusa ([Evert & The OCWB Development Team, 2010a]).

[37] U datom primeru (prva rečenica romana *1984* Džordža Orvela) obeležja `V`, `A`, `C`, `N`, `PUNC` redom označavaju glagol, pridev, veznik, imenicu i znak interpunkcije.

```
Bio       V      biti
je        V      jesam
vedar     A      vedar
i         C      i
hladan    A      hladan
aprilski  A      aprilski
dan       N      dan
.         PUNC   .
```

Slika 2.19: Vertikalni format anotirane rečenice (1).

TEI

Za razliku od prethodnih pokušaja nametanja standarda za kodiranje i anotaciju mašinski čitljivog teksta, **Smernice Inicijative za obeležavanje teksta** (eng. **Text Encoding Initiative Guidelines**, skr. **TEI Guidelines**) su prve naišle na masovan pozitivan prijem i opstale do danas, zahvaljujući mnogobrojnim korisnicima koji su našli interes da ih praktično sprovode. Projekat **Inicijativa za obeležavanje teksta** (eng. **Text Encoding Initiative**, skr. **TEI**) je nastao krajem osamdesetih godina XX veka u akademskom okruženju i pod patronatom asocijacija ACH (Association of Computer in the Humanities), ALLC (Association of Literary and Linguistic Computing) i ACL (Association for Computational Linguistics), a tokom 1999/2000. godine je formiran poseban neprofitni konzorcijum (TEI Consortium[38]) sa ciljem da promoviše, održava i razvija TEI.

Tokom svog postojanja, TEI je objavio pet predloga (verzija) svojih Smernica. Prvi predlog (TEI P1) je objavljen 1990. godine, a aktuelni peti predlog (TEI P5) se pojavio 2007. godine ([Burnard & Bauman, 2009]). Smernice opisuju jezik za označavanje koji se tokom prve tri verzije oslanjao na SGML, a predlozi nastali posle 2002. godine (verzija TEI P4 i kasnije) su zasnovani na specifikaciji XML-a.

Smernice TEI teže da budu sveobuhvatne i da opišu anotaciju što više različitih tipova elektronskog teksta. Otuda i njihova obimnost, pa tako Smernice predloga TEI P5 na oko hiljadu i petsto strana objašnjavaju upotrebu skoro pet stotina različitih XML-elemenata (TEI-elemenata). Da bi se izbeglo ponavljanje definicija zajedničkih sadržaja elemenata i atributa, uključujući i specifikacije zajedničkih

[38] http://www.tei-c.org

vrednosti atributa, koriste se klase, makroi i tipovi podataka ([Burnard & Bauman, 2009], dodaci A, B i E). Klase mogu biti atributske (grupišu elemente sa zajedničkim atributima) i klase modela (grupišu elemente koji se mogu pojaviti na istoj lokaciji u okviru TEI-dokumenta). Makroi su skraćena imena za modele sadržaja (ili delove modela sadržaja) koji se često pojavljuju kao delovi definicija elemenata. Posebnu vrstu makroa predstavljaju tipovi podataka čiji se nazivi često koriste u definicijama atributa umesto navođenja svih mogućih vrednosti atributa.

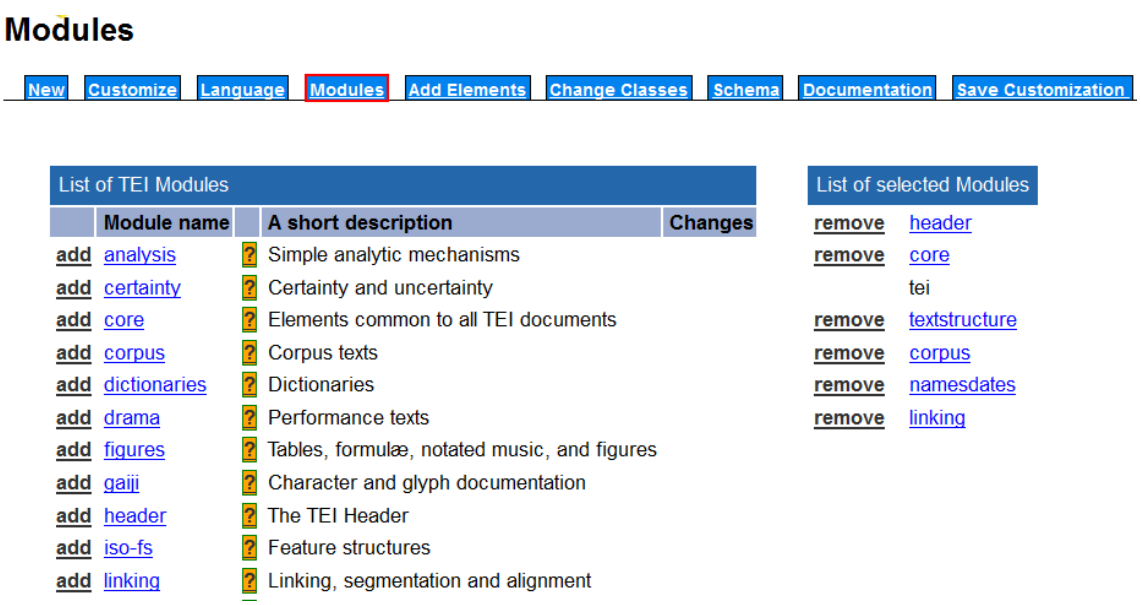
S obzirom da većinu korisnika zanima svega nekoliko različitih tipova elektronskog teksta (ili čak samo jedan), definicije TEI-elemenata su organizovane modularno sa hijerarhijom nasleđivanja elemenata i atributa, što omogućava da korisnici izaberu podskup jezika za označavanje koji će koristiti. U tu svrhu je razvijen i poseban alat Roma, dostupan na sajtu TEI-konzorcijuma[39]. Roma omogućava korisniku da konstruiše svoj jezik za označavanje zasnovan na postojećem TEI-jeziku, dodavanjem ili brisanjem kako modula, tako i pojedinačnih elemenata i atributa, a takođe i da elemente i atribute preimenuje, redefiniše njihov sadržaj, itd. (Slika 2.20). Na kraju Roma omogućava korisniku da snimi XML-šemu pomoću koje će proveravati da li prilikom primene izabranog podskupa TEI-jezika kreira validne XML-dokumente. XML-šema se može snimiti u raznim formatima: DTD, W3C XML Schema, RELAX NG (XML-sintaksa i kompaktna sintaksa), itd.

Prilikom izbora modula korisniku su na raspolaganju šabloni sastavljeni od unapred izabranih modula, pri čemu korisnik može dalje da prilagođava izabrani šablon dodavanjem ili brisanjem modula, elemenata, atributa, itd. Pored obaveznog modula `tei`, šabloni uglavnom sadrže i one module koji su zastupljeni u svim ili gotovo svim TEI-dokumentima:

- `core`
- `header`
- `textstructure`

Modul `tei` se sastoji od deklaracija klasa, tipova podataka i makroa koji su na raspolaganju svim TEI-modulima ([Burnard & Bauman, 2009], poglavlje 4).

[39] `http://www.tei-c.org/Roma`



Slika 2.20: Roma: stranica za izbor TEI-modula koji će formirati korisnikov jezik za označavanje.

Modul `core` čine elementi zajednički za sve TEI-dokumente, tj. elementi koji se mogu koristiti za opis proizvoljnog tipa teksta ([Burnard & Bauman, 2009], poglavlje 3).

Modul `header` je sastavljen od elemenata kojima se specifikuje zaglavlje TEI-dokumenta, tj. elementa `teiHeader` i njegove dece ([Burnard & Bauman, 2009], poglavlje 2). Zaglavlje TEI-dokumenta sadrži bibliografske podatke o elektronskoj i izvornoj verziji teksta, opisuje kako je izvorna verzija teksta transformisana u konačnu elektronsku verziju, navodi istoriju izmena u dokumentu, kao i odgovorne osobe za pojedine faze kreiranja i ažuriranja elektronske verzije teksta, nosioce autorskih prava, osobe zadužene za diseminaciju i distribuisanje TEI-dokumenta, itd.

Modul `textstructure` obuhvata elemente kojima se opisuje podrazumevana struktura visokog nivoa većine tipova teksta ([Burnard & Bauman, 2009], poglavlje 4). Podrazumevana struktura TEI-dokumenta se realizuje na dva načina (Slika 2.21):

- preko korenog elementa `TEI` koji se sastoji iz TEI-zaglavlja (element `teiHeader`) i teksta (element `text`), ili
- preko korenog elementa `teiCorpus` koji se sastoji iz TEI-zaglavlja korpusa

(element `teiHeader`) i jednog ili više elemenata `TEI`.

```
<!-- 1. način -->
<TEI>
  <teiHeader>
<!-- zaglavlje dokumenta (obavezno) -->
  </teiHeader>
  <text>
<!-- tekst dokumenta -->
  </text>
</TEI>
```

```
<!-- 2. način -->
<teiCorpus>
 <teiHeader>
<!-- zaglavlje korpusa (obavezno) -->
 </teiHeader>
 <TEI>
<!-- obavezan -->
  <teiHeader>
<!-- zaglavlje jednog teksta (obavezno)-->
  </teiHeader>
  <text>
<!-- tekst dokumenta -->
  </text>
 </TEI>
<!-- ... -->
 <TEI>
<!-- obavezan -->
  <teiHeader>
<!-- zaglavlje jednog teksta (obavezno)-->
  </teiHeader>
  <text>
<!-- tekst dokumenta -->
  </text>
 </TEI>
</teiCorpus>
```

Slika 2.21: Podrazumevana struktura TEI-dokumenta

Pored navedenih, najčešće korišćenih modula, TEI P5 nudi i:

- modul `analysis` koji omogućava kreatoru TEI-dokumenta da pridruži tekstu ili delu teksta željene semantičke, sintaksičke ili morfološke interpretacije.

Prema tome, ovaj modul podržava lingvističku anotaciju na različitim nivoima, koristeći elemente `s` (rečenica), `phr` (sintagma), `w` (korpusna reč), kao i atribut `ana` (vrednost atributa je pridružena analiza, na primer, vrsta reči);

- `corpus`, posvećen specifičnostima anotacije elektronskih korpusa (([Burnard & Bauman, 2009], poglavlje 15)) koje nisu pokrivene najčešće korišćenim modulima (`tei`, `core`, `header`, `textstructure`).

- modul `linking`, namenjen **povezivanju** (eng. **linking**), segmentaciji i **poravnanju** ili **uparivanju** (eng. **alignment**) TEI-dokumenata. Ovaj modul je posebno značajan za odvojenu anotaciju korpusa, kao i za anotaciju jedinica prevođenja paralizovanih korpusa (v. odeljak 6.5);

- modul `spoken` čiji se elementi koriste za anotaciju transkribovanog govora;

- modul `namesdates` čiji se elementi koriste za anotaciju imenovanih entiteta poput imena ljudi, geografskih lokacija i organizacija, kao i za označavanje izraza kojima se opisuju datum i vreme;

- modul `msdescriptions`, kojim se opisuju rukopisi kao primarni izvori elektronskog teksta;

- modul `transcr` transkripcija tekstuelnih komponenenti primarnih izvora elektronskog teksta;

- modul `dictionaries`, namenjen opisu elektronskih rečnika;

- modul `verse`, koji se primenjuje za anotaciju poetskih tekstova.

Kako bi se početnicima omogućilo da se na jednostavan način upoznaju sa osnovama Smernica TEI, kako bi u što kraćem roku mogli i praktično da ih primene, autori Smernica TEI su 1995. godine izdvojili standardizovani podskup najčešćih ili najvažnijih TEI-elemenata i atributa — TEI-Lite ili „laku verzija TEI" ([Burnard & Sperberg-McQueen, 2012]). Većina istraživača koji imaju nameru da prouče Smernice TEI kako bi izgradili svoj model anotacije, koriste upravo TEI-Lite kao polaznu osnovu.

Za dokument nastao primenom TEI-anotacije kaže se da je **usaglašen sa TEI** (eng. **TEI Conformant**) ako i samo ako ([Burnard & Bauman, 2009], odeljak 23.3):

(T1) predstavlja dobro formirani XML-dokument;

(T2) validan je u odnosu na XML-šemu koja prati Smernice TEI;

(T3) usaglašen je sa Apstraktnim modelom TEI;

(T4) ispravno koristi prostor imena koji definiše TEI, kao i relevantne prostore imena;

(T5) postoji dokumentacija koja ga opisuje u formatu ODD i u skladu sa Smernicama TEI;

(T6) može da se transformiše u dokument koji zadovoljava uslove (T1)–(T5) nekom procedurom koju definiše TEI.

Prvi značajniji primer primene anotacije TEI-anotacije u korpusnoj lingvistici je anotacija Britanskog nacionalnog korpusa u čemu su direktno učestvovali i pojedini tvorci standarda TEI. Tom prilikom je, kao shema anotacije, primenjen **Format za razmenu korpusnih dokumenata** (eng. **Corpus Document Interchange Format**, skr. **CDIF**) [Burnage & Dunlop, 1993; Burnard, 1992], zasnovan na verziji TEI P3 ([Sperberg-McQueen & Burnard, 1994]) i standardu SGML. Aktuelna verzija Britanskog nacionalnog korpusa ([Burnard, 2007]) je anotirana verzijom CDIF-a koja je usklađena sa TEI P4 ([Sperberg-McQueen & Burnard, 2004]) i specifikacijom XML-a.

CES/XCES

I pored široke prihvaćenosti Smernica TEI među korpusnim istraživačima, od samog početka su na njihov račun stizale i ozbiljne kritike ([Lehmberg & Wörner, 2008]):

- pokušaj Smernica TEI da budu „sveobuhvatne" čini ih glomaznim skupom etiketa, bez obzira na modularnu organizaciju, što odbija istraživače koji tek počinju da ih koriste;

- Smernice TEI omogućavaju da se isti fenomen anotira na različite načine, što praktično znači da se ne garantuje njegova konzistentna anotacija u korpusu, a što bi pravi standard morao da obezbedi;
- Pojedini mehanizmi TEI-anotacije su na tako visokom nivou apstrakcije, da se više ne može reći da su korisni;
- Višeznačnost i hijerarhija ne omogućavaju uvek efikasnu validaciju anotacije, što je od posebne važnosti za korpus i računarsku obradu;
- TEI daje prednost ugrađenoj anotaciji, dok podrške za odvojenu anotaciju nema ili je u začetku.

Navedene kritike su uticale na pojedine korpusne istraživače da se ne opredele za anotaciju pomoću TEI. Drugi, poput savetodavnog tela EAGLES, su odlučili da kreiraju sopstveni predlog standarda za anotaciju korpusa, usaglašen sa Smernicama TEI.

Stručne savetodavne grupe za standarde jezičkog inženjerstva (eng. **Expert Advisory Groups on Language Engineering Standards**, skr. **EAGLES**) je formirala Evropska komisija (EK) 1993. godine sa ciljem da razvije jezičke standarde koji bi se odnosili na zvanične jezike Evropske zajednice (EZ) i koristili u budućim projektima EZ. U okviru svojih smernica iz 1996. godine, EAGLES predlažu **Standard za kodiranje korpusa** (eng. **Corpus Encoding Standard**, skr. **CES**). CES je usaglašen sa Smernicama TEI i predstavlja kombinaciju podskupa postojećih TEI-elemenata i atributa, relevantnih za anotaciju korpusa, kao i definicije novih elemenata i atributa. S obzirom da je u tom trenutku TEI koristio SGML, na isti način je definisan i CES ([Ide, 1998]), a posle odluke da se u Smernicama TEI primeni XML kao jezik za označavanje teksta, nastaje i **XCES** ([Ide et al., 2000]), XML-verzija standarda CES. Definiciji verzija CES i XCES doprineli su i pojedini saradnici na izradi Smernica TEI.

Za razliku od standarda TEI, CES i XCES se bave isključivo anotacijom korpusa, pokušavajući istovremeno da nadomeste nedostatke koji se zameraju TEI-anotaciji. U tom smislu, CES i XCES teže da eliminišu višeznačnost i obezbede konzistentno

sprovođenje anotacije preciznijim definicijama značenja i upotrebe svojih elemenata i atributa. Takođe, CES i XCES pružaju podršku za odvojenu anotaciju.

Od tipova anotacije navedenih u odeljku 1.3, pododeljak Anotacija, CES i XCES podržavaju i lingvističku i nelingvističku anotaciju. Kad je reč o lingvističkoj anotaciji, CES posebno detaljno definiše mehanizme morfosintaksičke anotacije, kao i anotaciju paralelnih korpusa, dok XCES uvodi podršku i za anotaciju diskursa, govora i rečnika. U pogledu nelingvističke anotacije, CES/XCES, kao i TEI, definišu zaglavlje korpusa i pojedinačnih tekstova[40]

Jedan od prvih korpusa koji je primenio XCES je Američki nacionalni korpus ([Reppen & Ide, 2004]).

Prva verzija XCES-a je uživala popularnost pravog standarda. Međutim, u sledećoj verziji iz 2003. godine dotadašnja konkretna morfosintaksička shema je zamenjena uopštenim mehanizmom **stukture (gramatičkih) kategorija** (eng. **feature structure**) koji nije usklađen sa odgovarajućim standardom ISO 24610-1 o predstavljanju stukture (gramatičkih) kategorija (eng. ISO Feature Structure representation standard, skr. FSR). To je među korpusnim istraživačima shvaćeno kao „korak nazad" ([Bański & Przepiórkowski, 2010]), jer zahteva više posla na konkretizaciji anotacione sheme.

Pored navedenog, standardima CES i XCES se može zameriti i sledeće ([Przepiórkowski, 2009]):

- nepotpuna dokumentacija, posebno za XCES;
- od konkretnih shema anotacije, razrađena je jedino morfosintaksička anotaciona shema;
- za validaciju XCES-dokumenata ponuđene su dve vrste shema, DTD i XSD, pri čemu nije navedeno koje su bitne razlike između njih, što može dovesti do nekonzistentne primene anotacije;
- ne postoje mehanizmi za anotaciju razdvojenih elemenata teksta kao celine (eng. discontinuity);

[40]Dok TEI koristi elemente `teiCorpus` i `TEI` za anotaciju strukture korpusa, kao i `teiHeader` za opis zaglavlja korpusa i njegovih pojedinačnih tekstova, CES i XCES koriste elemente `cesCorpus`, `cesAna` i `cesHeader`.

- neusaglašenost sa TEI P5;
- nemogućnost alternativne anotacije.

Poslednja zamerka je upravo ono što je (X)CES hteo da izbegne, tj. da omogući da se anotacija uvek sprovede jednoznačno. Što se tiče neusaglašenosti sa TEI P5, trenutno je u izradi nova verzija standarda XCES, usklađena sa aktuelnom verzijom predloga TEI-anotacije. S obzirom da je DTD potisnut od strane XSD-sheme i RELAXNG-sheme, zamerka o konfuziji u korišćenju shema za validaciju se može zaobići primenom naprednije sheme, dakle, XSD-sheme. Međutim, nepotpunost dokumentacije i nedostatak primera na zvaničnom sajtu[41] su svakako glavni razlog zašto je XCES izgubio na popularnosti poslednjih godina.

ISO/TC 37

Tehnički komitet ISO/TC Međunarodne organizacije za standardizaciju (ISO) je zadužen za „standardizovanje principa, metoda i aplikacija koji se odnose na terminologiju i druge jezičke resurse i resurse sadržaja u kontekstu višejezičke komunikacije i kulturoloških različitosti"[42], a poseban potkomitet za upravljanje jezičkim resursima, ISO/TC /SC [43], bavi se izradom standarda za anotaciju ([Pustejovsky & Stubbs, 2012: 62]). Do sada je ISO/TC /SC objavio ukupno dvanaest standarda i to[44]:

ISO 24610-1:2006 Upravljanje jezičkim resursima — Strukture (gramatičkih) kategorija — Deo 1: Reprezentacija strukture (gramatičkih) svojstava (eng. Language resource management — Feature structures — Part 1: Feature structure representation);

[41] `http://www.xces.org`

[42] Republika Srbija učestvuje u radu ISO/TC 37 preko Komisije za standarde A037 (Terminologija) Instituta za standardizaciju Srbije (ISS). Navedeni opis područja rada ove komisije preuzet je sa zvanične prezentacije ISS-a: `http://www.iss.rs/tc/?national_committee_id=552` 26. IV 2013. godine.

[43] Republika Srbija, odnosno ISS, prisutni su u potkomitetu ISO/TC 37/SC 4 samo u svojstvu posmatrača.

[44] Prevod naziva objavljenih standarda nije zvaničan prevod Instituta za standardizaciju Srbije (ISS), već autora rada, pošto srpska verzija navedenih standarda još uvek nije objavljena.

ISO 24610-2:2011 Upravljanje jezičkim resursima — Strukture (gramatičkih) kategorija — Deo 2: Deklaracija sistema (gramatičkih) kategorija (eng. Language resource management — Feature structures — Part 2: Feature system declaration);

ISO 24611:2012 Upravljanje jezičkim resursima — Okvir za morfosintaksičku anotaciju (eng. Language resource management — Morpho-syntactic annotation framework (MAF));

ISO 24612:2012 Upravljanje jezičkim resursima — Okvir za lingvističku anotaciju (eng. Language resource management — Linguistic annotation framework (LAF));

ISO 24613:2008 Upravljanje jezičkim resursima — Okvir za leksičko označavanje (eng. Language resource management - Lexical markup framework (LMF));

ISO 24614-1:2010 Upravljanje jezičkim resursima — Segmentacija na reči pisanih tekstova — Deo 1: Osnovni koncepti i opšti principi (eng. Language resource management — Word segmentation of written texts — Part 1: Basic concepts and general principles);

ISO 24614-2:2011 Upravljanje jezičkim resursima — Segmentacija na reči pisanih tekstova — Deo 2: Segmentacija na reči za kineski, japanski i korejski jezik (eng. Language resource management — Word segmentation of written texts — Part 2: Word segmentation for Chinese, Japanese and Korean);

ISO 24615:2010 Upravljanje jezičkim resursima — Okvir za sintaksičku anotaciju (eng. Language resource management — Syntactic annotation framework (SynAF));

ISO 24616:2012 Upravljanje jezičkim resursima — Okvir za višejezične informacije (eng. Language resources management — Multilingual information framework);

ISO 24617-1:2012 Upravljanje jezičkim resursima — Okvir za semantičku anotaciju — Deo 1: vreme i događaji (eng. Language resource management —

Semantic annotation framework (SemAF) — Part 1: Time and events (SemAF-Time, ISO-TimeML));

ISO 24617-2:2012 Upravljanje jezičkim resursima — Okvir za semantičku anotaciju — Deo 2: dijaloški činovi (eng. Language resource management — Semantic annotation framework (SemAF) — Part 2: Dialogue acts);

ISO 24619:2011 Upravljanje jezičkim resursima — Trajna identifikacija i održivi pristup (eng. Language resource management — Persistent identification and sustainable access (PISA));

Okvir za lingvističku anotaciju (eng. **Linguistic Annotation Framework**, skr. **LAF**) je ISO-standard razvijan od 2005. godine i odobren 2012. godine kao ISO 24612. LAF se nadovezuje na nezvanične standarde za anotaciju, TEI i (X)CES, sa ciljem da ih zameni. LAF nudi apstraktan model podataka koji se može primeniti na svim nivoima lingvističke anotacije, a koristi graf za reprezentaciju strukture podataka. Poseban **Format za anotaciju grafova** (eng. **Graph Annotation Format**, skr. **GrAF**), zasnovan na XML-u, koristi se za interni zapis LAF-anotacije. LAF omogućava da se u praksi koriste i drugi formati lingvističke anotacije (vertikalni, horizontalni, zasnovani na XML-u) pod uslovom da se direktno mogu transformisati u GrAF. GrAF je napravljen sa ciljem da se koristi kao međuformat prilikom automatske konverzije jednog formata lingvističke anotacije u drugi.

Ostale važnije karakteristike LAF-anotacije su ([Ide & Romary, 2006]):

- LAF koristi odvojenu anotaciju;
- svaki nivo LAF-anotacije se čuva u posebnoj datoteci čime se omogućava da se nezavisno kreira i ažurira u odnosu na ostale nivoe anotacije, kao i da se koriste alternativne anotacije istog lingvističkog fenomena;
- prilikom spajanja različitih slojeva LAF-anotacije i originalnog teksta u jednu GrAF-datoteku, rezultat je dobro formirani XML-dokument;

- LAF nema svoje kategorije podataka koje bi dodatno opisale značenje lingvistički anotiranog teksta, već koristi standard ISO 12620:2009[45] i njegovu implementaciju ISOcat[46].

S obzirom da izrada ISO standarda teče po fazama[47], tako da prođe po nekoliko godina od predloga do prihvatanja i objavljivanja standarda, većina korpusnih istraživača je izbegavala da primeni LAF za anotaciju korpusa, kao i druge standarde koje je razmatrao potkomitet ISO/TC 37/SC 4, pogotovo dok su još bili u statusu predloga. Kako je LAF usvojen tek 2012. godine[48], još uvek je rano analizirati efekte njegove primene, kao i odgovoriti na pitanje u kojoj meri je prihvaćen od zajednice korpusnih lingvista.

Morfosintaksička anotacija korpusa

Morfosintaksička anotacija (**morfološka anotacija** ili **gramatička anotacija**) predstavlja posebnu vrstu lingvističke anotacije kojom se, u opštem slučaju, svakom tokenu teksta pridružuju odgovarajuće sledeće informacije:

- vrsta reči,
- kanonski oblik ili lema,
- vrednosti morfoloških kategorija poput roda, broja, padeža, lica, itd.

U užem smislu, morfosintaksičkom anotacijom se tokenu pridružuje samo neka od navedenih informacija. Ukoliko je to samo vrsta reči, proces pridruživanja se još naziva **etiketiranje vrstom reči** (eng. **Part of Speech Tagging**, skr. **PoS tagging**), mada se ponekad engleski termini *PoS tagging* i *tagging* koriste da označe morfosintaksičku anotaciju uopšte. Ukoliko se tokenu pridružuje samo informacija o lemi, proces pridruživanja se naziva **lematizacija** (eng. **lemmatization**).

[45]Standard ISO 12620:2009 opisuje **Registar kategorija podataka** (eng. **Data Category Registry**, skr. **DCR**). ISO 12620:2009 i DCR razvija potkomitet ISO/TC 37/SC 3.9.

[46]`http://www.isocat.org`

[47]Do zvaničnog prihvatanja, svaki standard mora da prođe kroz fazu projekta, fazu predloga, fazu prednacrta, fazu nacrta, fazu javne rasprave, fazu definitivnog teksta nacrta, faza objavljivanja, fazu preispitivanja, a u slučaju objavljivanja novog izdanja standarda ili potpuno novog standarda sa istim predmetom, i fazu povlačenja prethodne verzije standarda ([ISS, 2012]).

[48]Nepunih godinu dana pre no što je nastao ovaj tekst.

Polazeći od „glasovnog (fonemskog) oblika reči, od njihovih gramatičkih oblika, od njihove službe (funkcije) u jeziku i rečenici i od njihovog gramatičkog (najopštijeg) značenja" sve reči prirodnog jezika se mogu podeliti na klase koje se nazivaju **vrste reči** (eng. **Part of Speech**, skr. **PoS**) ili **leksičke klase** (eng. **lexical class, word class**) ([Stanojčić & Popović, 2008]). Podela na vrste reči varira od jednog do drugog prirodnog jezika, ali postoje i vrste reči koje su zajedničke za većinu jezika. U srpskom jeziku se razlikuje deset vrsta reči: imenice, zamenice, pridevi, brojevi, glagoli, prilozi, predlozi, veznici, uzvici i rečce (partikule).

Vrste reči mogu biti promenljive i nepromenljive. Promenljive vrste reči se realizuju u tekstu kroz različite gramatičke (flektivne) oblike zavisno od njihove funkcije u rečenici, odnosno od gramatičkog značenja. Skup različitih gramatičkih (flektivnih) oblika iste reči predstavlja njenu (flektivnu) paradigmu.

Leksičko značenje, vrsta reči, paradigma, itd. predstavljaju informacije koje obuhvata jedan osnovni element uobičajenih rečnika u papirnatom obliku — leksička reč ili leksema. Leksema je skup oblika koji imaju istu osnovu, pripadaju istoj vrsti reči i imaju isto značenje. Umesto navođenja svih oblika u rečniku, paradigma lekseme je redukovana na jednog predstavnika, **kanonski oblik** ili **lemu** (eng. **lemma**), kojim je leksema predstavljena u rečniku[49]. U slučaju nepromenljivih vrsta reči, jedini oblik lekseme je ujedno i njena lema. Pojam leme se može uopštiti tako da svaki token, uključujući i znakove interpunkcije, ima lemu. U slučaju znakova interpunkcije, sam token se najčešće uzima za lemu, mada je moguće i da više različitih znakova interpunkcije sa sličnim značenjem imaju zajedničku lemu (na primer, različiti karakteri koji predstavljaju znake navoda).

Svaka vrsta reči se odlikuje nekim skupom morfoloških kategorija koje mogu biti karakteristične za više vrsta reči (na primer, padež je morfološka kategorija imenica, prideva, zamenica), dok su neke morfološke kategorije specifične za pojedine vrste reči (na primer, stepen poređenja za prideve, ili glagolski oblik za glagole).

Pre no što se pristupi morfosintaksičkoj anotaciji, neophodno je utvrditi oznake pojedinačnih morfoloških informacija, **morfološke deskriptore** ili **etikete** (eng.

[49]Na primer, lema imenica je oblik u nominativu jednine (ako postoji, kontraprimer je imenica `vrata`), dok su glagoli u rečniku predstavljeni infinitivom kao lemom (ako postoji, kontraprimer je glagol `velim`).

tag), i precizno definisati njihovo značenje[50]. Najjednostavniji **skup etiketa** (eng. **tagset**) sadrži samo oznake uobičajenih leksičkih klasa (imenice, pridevi, glagoli, itd.), dok izrazito detaljni skup etiketa, pored vrste reči i vrednosti morfoloških kategorija, sadrže i informacije o podvrstama reči (na primer, zamenice u srpskom jeziku mogu biti imeničke i pridevske; pridevske zamenice mogu biti prisvojne, pokazne, upitno-odnosne, odrične, neodređene, određene) i informacije o klasifikacionim kategorijama (na primer, rod imenice). Prema tome, veličina skupa etiketa ne samo da se menja od jednog do drugog prirodnog jezika, već je i za isti prirodni jezik moguće kreirati skupove etiketa različite veličine. Na veličinu skupa etiketa direktno utiče namena anotiranog korpusa, tj. šta korpusni istraživači žele da pronađu u korpusu. Detaljnije anotirane morfološke informacije u korpusu zahtevaju veći broj etiketa, između kojih ponekad postoje samo suptilne razlike, što otežava proces anotacije, bilo da je u pitanju ručna, automatska ili poluautomatska anotacija.

Automatska morfosintaksička anotacija

Osnovni problemi sa kojim se suočava automatska morfosintaksička anotacija su ([Vitas, 2007]):

- višeznačnost i
- obrada „nep(rep)oznatih" reči.

Višeznačnost (eng. **ambiguity**) nastupa kada se jednom tokenu može pridružiti više morfosintaksičkih opisa koji se međusobno isključuju, kao što je to slučaj sa tokenom `jedu` koji može biti oblik imenice `jed` (primer (1)) ili treće lice prezenta glagola `jesti` (primer (2)).

(1) „Koliko mu je samo puta svirao dok je ovaj davao oduška svom golemom jedu i besu;" (Milisav Savić, *Ujak naše varoši*, pripovetka *Kapetan Vuk*)

(2) „Veštice ne jedu beli luk! - rekla je ujna." (Milisav Savić, *Ujak naše varoši*, pripovetka *Odbrana varoši od veštica*)

[50]U slučaju leme to nije neophodno, tj. svaka lema je ujedno i oznaka te informacije.

Ukoliko program za anotaciju dodeljuje tokenu samo vrstu reči, mora da odluči između dve mogućnosti. Međutim, u slučaju da se tokenu pridružuju i vrednosti morfoloških kategorija (na primer, broj i padež), a program za anotaciju analizom eliminiše drugu mogućnost (glagolski oblik), još uvek treba da razreši da li se radi o dativu jednine ili lokativu jednine imenice `jed`. Za eliminaciju morfosintaksičke višeznačnosti neophodna je analiza šireg konteksta tokena koji se anotira, najčešće rečenice koja sadrži token, ali ponekad i logičkih delova teksta na višem nivou, uključujući i sam tekst, odnosno domen kome tekst pripada. Uz sve to, čovek koristi i svoje celokupno znanje i zdrav razum, što je daleko više od skupa informacija i mehanizama za izvođenje zaključaka kojima raspolažu programi za morfosintaksičku anotaciju.

„Nep(rep)oznate" reči (eng. **unknown words**) su tokeni o kojima program za anotaciju ne poseduje nikakvu informaciju. U najjednostavnijim slučajevima se resursi koji se konsultuju tokom anotacije mogu dopuniti nedostajućim informacijama, ili program za anotaciju može na osnovu konteksta tokena odlučiti koju će etiketu pridružiti pravljenjem analogije sa obavljenom anotacijom u istovetno anotiranim kontekstima. Rešenju problema „nep(rep)oznatih" reči odmažu i *hapaksi* (grč. $\alpha\pi\alpha\xi$ $\lambda\varepsilon\gamma o\mu\varepsilon\nu o\nu$), obično jednom upotrebljene reči, skovane samo za tu priliku primenom nekog poznatog tvorbenog mehanizma.

U kombinaciji sa višeznačnošću i „nep(rep)oznatim" rečima, obimniji skup sa etiketama, koje se razlikuju u finim detaljima, posebno utiče na smanjenje preciznosti anotacije, jer je programu za anotaciju teže da odluči između više sličnih mogućnosti.

Programi za automatsku morfosintaksičku anotaciju se mogu podeliti u tri grupe:

- programi zasnovani na pravilima;
- probabilistički ili stohastički programi;
- programi koji kombinuju prethodna dva pristupa.

Programi za automatsku morfosintaksičku anotaciju **zasnovani na pravilima** (eng. **rule-based taggers**) koriste listu ručno izrađenih pravila koja se koriste

za razrešavanje višeznačnosti. Pravila se oslanjaju na lingvističko znanje i neposredno zavise od konkretnog prirodnog jezika. Nedostaci ovog pristupa su potreba za stručnim ljudskim resursima, kao i znatno vreme i napor da se lista pravila kreira i održava. Kao glavna prednost anotacije zasnovane na pravilima, najčešće se navodi visoka preciznost postignuta za pojedine jezike[51].

Probabilistički ili stohastički (eng. **probabilistic/stochastic taggers**) programi za automatsku morfosintaksičku anotaciju koriste skup za treniranje kako bi izračunali verovatnoću da se datom tokenu u datom kontekstu pridruži određena etiketa, a potom dodeljuju onu etiketu za koju je izračunata najveća verovatnoća. Ovaj koncept se realizuje kroz različite tehnike mašinskog učenja koje su detaljno opisane u [Jurafsky & Martin, 2008; Popović, 2008, 2010; Sečujski, 2009], pri čemu **skriveni Markovljevi modeli** (eng. **Hidden Markov models**, skr. **HMM**) predstavljaju „najrasprostranjeniji pristup etiketiranju vrstom reči s obzirom na njihovu tačnost i veliku brzinu obrade" ([Schmid, 2008: 540]).

Ukoliko je skup za treniranje prethodno ručno anotiran, u pitanju je tehnika nadgledanog mašinskog učenja. U slučaju neanotiranog skupa za treniranje, program sam generiše etikete tako što tokene skupa za treniranje grupiše po određenim kriterijumima sličnosti, a onda svakoj grupi dodeljuje određenu etiketu. Tako generisane etikete se koriste tokom morfosintaksičke anotacije korpusa primenom istih kriterijuma za svrstavanje tokena u grupe. Praksa pokazuje da tehnike nadgledanog mašinskog učenja postižu veću preciznost prilikom anotacije u odnosu na tehnike nenadgledanog mašinskog učenja, pogotovo ako je skup za treniranje obimniji ([Merialdo, 1994]).

Posebnu grupu predstavljaju programi za morfosintaksičku anotaciju koji kombinuju pravila i mašinsko učenje. Suštinska razlika u odnosu na prvi pristup je što se pravila za razrešavanje višeznačnosti ne kreiraju ručno, već se nekom tehnikom mašinskog učenja generišu na osnovu skupa za treniranje. Tipičan predstavnik ovog pristupa je Brilov program za morfosintaksičku anotaciju ([Brill, 1995]).

[51] Za sistem *EngCG*, morfološki anotator tekstova na engleskom jeziku zasnovan na pravilima, navodi se da postiže preciznost preko 99% ([Karlsson et al., 1995; Samuelsson & Voutilainen, 1997]).

Lematizacija Algoritmi lematizacije se mogu svrstati u dve grupe u zavisnosti od toga da li su zasnovani na korišćenju rečnika ili ne ([Fitschen & Gupta, 2008]).

Lematizacija koja nije zasnovana na rečniku se ponekad naziva **određivanje osnove reči** (eng. **stemming**), mada, strogo govoreći, lema i osnova su dva različita pojma, tako da su i lematizacija i određivanje osnove suštinski različiti procesi. Određivanje osnove reči redukuje flektivni oblik na osnovu reči koja se u jezicima poput engleskog često poklapa sa lemom, što i dovodi do pomenute terminološke zabune. Ovaj pristup se više koristi u sistemima za **pronalaženje informacija** (eng. **Information Retrieval**, skr. **IR**), nego u korpusnoj lingvistici. Najpoznatija implementacija ovog pristupa je Porterov algoritam ([Porter, 1980]), zasnovana na pravilima koja brišu, odnosno zamenjuju, sufikse oblika kako bi proizveli osnovu. Iako je algoritam jednostavan, ima efikasnu implementaciju pomoću kaskade konačnih transduktora i poboljšava pretragu informacija u dokumentima na engleskom jeziku, nije pogodan za jezike sa razvijenijom morfologijom, a time ni za srpski, mada postoje pristupi za jezike sa razvijenijom morfologijom čija evaluacija daje kao rezultat tačnost od 79% ([Kešelj & Šipka, 2008]).

Algoritmi lematizacije zasnovani na rečniku se takođe mogu podeliti u dve grupe u zavisnosti od toga da li rečnik predstavlja spisak svih oblika popisanih lema (eng. full-form listing) ili rečnik sadrži samo spisak lema i mehanizam na osnovu kojeg se za svaku lemu rečnika mogu generisati svi oblici njene paradigme. Opis paradigme se obično zadaje regularnim izrazom, tako da se rečnik lema i pravila za generisanje paradigmi implementiraju kao konačni transduktori koji za izlaz imaju lematizirane tokene ulaznog teksta. Detaljniji primer ovakvog rečnika je opisan u odeljku LADL/DELA, str. 145.

Formati morfosintaksičke anotacije za tekstove na srpskom jeziku

LADL/DELA *Laboratorija za automatsku dokumentaciju i lingvistiku* (fr. *Laboratoire d'Automatique Documentaire et Linguistique*, skr. *LADL*) je razvila formate morfosintaksičkih opisa, **LADL-elektronski rečnici** (fr. **Dictionnaire Électronique du LADL**, skr. **DELA**), koji se koriste u morfološkim elektronskim rečnicima najpre francuskog jezika ([Courtois & Silberztein, 1990]), a potom i dvadesetak dru-

gih jezika[52], uključujući i srpski (Krstev & Vitas [2005]).

DELA zapravo predstavlja skup nekoliko formata (DELAS, DELAF, itd.). DELAS je format rečnika lema, dok je DELAF format rečnika flektivnih oblika. DELAF se neposredno koristi u morfološkoj analizi i anotaciji teksta, dok se DELAS koristi za automatsko generisanje flektivnih oblika. Zapis jedne odrednice u formatu DELAF (Tabela 2.9), pored obaveznog oblika leksičke reči, informacije o lemi i vrsti reči, može sadržati flektivne kategorije, kao i sintaksičko-semantičke markere na osnovu kojih se iz rečnika mogu ekstrahovati odrednice sa određenim semantičkim ulogama (agens, pacijens, instrument, itd.), izgovorom (dijalekt), odrednice koje predstavljaju određeni tip tvorbe (prisvojni/relacioni pridev, mocija roda, deminutiv, augmentativ) ili određenu vrstu imenovanih entiteta (geografski pojam, naziv organizacije, itd.)[53]. DELAS (gornji deo Tabele 2.9) se razlikuje od formata DELAF (donji deo Tabele 2.9) po odsustvu flektivnih oblika i vrednosti flektivnih kategorija, a oznaka vrste reči (u ovom primeru `A`) sadrži i dodatnu oznaku (u ovom slučaju broj `1`) tako da zbirna oznaka (`A1`) predstavlja flektivnu klasu. Flektivna klasa je realizovana kao regularni izraz (v. odeljak 3.1 str. 175).

Svako veliko i malo slovo engleskog alfabeta predstavlja oznaku najviše jedne vrednosti neke flektivne kategorije, tako da se za vrednosti flektivnih kategorija koristi nepoziciona notacija, odnosno zapisi `:adms4v` i `:m4sadv` imaju isto značenje. Nažalost, u rečnicima različitih jezika primenjuju se različite oznake za vrste reči i vrednosti morfoloških kategorija.

LADL je razvio programski alat INTEX ([Silberztein, 1999]) namenjen obradi korpusa pomoću morfoloških elektronskih rečnika u formatima DELA. Posle razlaza sa tvorcem INTEX-a, Maksom Silberštajnom, LADL je razvio novi alat — Unitex ([Paumier, 2011]). Unitex podržava dve varijante morfosintaksičke anotacije u formatu DELAF (Slika 2.22). Prva varijanta je zasnovana na pravilima koja su opisana pomoću mehanizma ELAG ([Laporte & Monceaux, 1998]), a potonja na

[52]Na adresi `http://www-igm.univ-mlv.fr/~unitex/index.php?page=5` su navedeni podaci o leksičkim resursima javno dostupnim u formatima LADL/DELA za engleski, francuski, finski, starogruzijski, nemački, grčki i starogrčki, italijanski, latinski, norveški, portugalski, ruski, srpski, španski, tajlandski (taj).

[53]Izuzimajući semantičke uloge, ostali pomenuti sintaksičko-semantički markeri su implementirani u elektronskom morfološkom rečniku srpskog jezika u formatu DELA (v. na primer [Krstev, 2008]).

Tabela 2.9: Primer zapisa u formatima DELAS i DELAF u morfološkom elektronskom rečniku srpskog jezika.

Primer zapisa u rečniku lema (format DELAS)		
`studentov,A1+Pos+Der+Hum`		
Morfosintaksički opis odrednice (format DELAF)		
`studentovog,studentov.A+Pos+Der+Hum:adms4v`		
Objašnjenje morfosintaksičkog opisa		
`studentovog`	leksička reč (element rečnika), flektivni oblik	
`studentov`	lema (kanonski oblik leksičke reči)	
`A`	vrsta reči (**pridev**)	
`+Pos+Der+Hum`	Sintaksičko-semantički markeri	
	`+Pos`	**prisvojni pridev**
	`+Der`	**izvedenica**
	`+Hum`	izveden od **osobe**
`:adms4v`	Flektivne kategorije	
	`a`	**pozitiv** (stepen poređenja)
	`d`	**određeni** (vid)
	`m`	**muški** (rod)
	`s`	**jednina** (broj)
	`4`	**akuzativ** (padež)
	`v`	**animatnost**/inanimatnost

nadgledanom mašinskom učenju ([Paumier, 2011], poglavlje 7).

MULTEXT-East U trenutku kad su se Smernice TEI već nametnule kao nezvanični standard za anotaciju teksta, a EAGLES već uveliko radile na standardu CES, usledio je niz projekata: MULTEXT ([Ide & Véronis, 1994]), MULTEXT-East ([Erjavec, Krstev, Petkevič, Simov, Tadić & Vitas, 2003], Erjavec [2010]), TELRI I–II ([Erjavec, Lawson & Romary, 1998]), CONCEDE ([Erjavec, Evans, Ide & Kilgarriff, 2003]). Prvi u nizu, pod nazivom **Višejezični alati i korpusi** (eng. **Multilingual Tools and Corpora**, skr. **MULTEXT**) je sebi postavio sledeće ciljeve koje su potonji projekti dalje razradili i proširili:

- razvoj standarda i specifikacija za kodiranje i obradu lingvističkih korpusa, koji bi proširili standard TEI i ujedno ga testirali na podacima realne veličine;

```
 1 {S}
 2 {Glava,glava.N:fs1q}
 3 {prva,prvi.A+Ord:aefs1g}
 4 {S}{Kako,kako.CONJ}
 5 {je,jesam.V+Imperf+It+Iref+Aux:Pzsi}
 6 {Kandid,Kandid.N+NProp+Hum+Fict:ms1v}
 7 {odrastao,odrasti.V+Perf+It+Iref:Gsm}
 8 {u,u.PREP+p7}
 9 {jednom,jedan.NUM+v1:ms7g}
10 {lepom,lep.A+Ek:adms7g}
11 {zamku,zamak.N:ms7q}
12 {i,i.CONJ}
13 {kako,kako.CONJ}
14 {je,jesam.V+Imperf+It+Iref+Aux:Pzsi}
15 {iz,iz.PREP+p2}
16 {njega,on.PRO+PrsJG:msz2r}
17 {bio,biti.V+Imperf+Tr+Iref:Gsm}
18 {isteran,isterati.V+Perf+Tr+Iref+Ek:Tms}
19 {S}
```

Slika 2.22: Odlomak iz poluautomatski anotirane verzije Volterovog romana *Kandid* (format DELAF). Sa {S} su označeni separatori rečenica.

- razvoj programskih alata i resursa pomoću kojih bi se izgrađeni standardi i specifikacije primenili u praksi;
- rad na uspostavljanju softverskog standarda koji bi omogućio ponovno korišćenje kreiranih programskih komponenti (eng. reusability) za obradu korpusa.

Projekti MULTEXT su među prvima primenili tek objavljeni standard CES prilikom anotacije višejezičnog korpusa JOC, sastavljenog od tekstova zvaničnog časopisa Evropske zajednice (eng. Official Journal of European Community) na pet zapadnoevropskih jezika (engleski, francuski, italijanski, nemački i španski). JOC je anotiran informacijom o vrsti reči, a tekstovi su paralelizovani na nivou rečenice (v. odeljak 6.5, str. 290).

Ciljevi i postignuti rezultati su inicirali novi projekat, MULTEXT-East ([Erjavec, 2010]), usmeren ka proširivanju jezičkih i programskih resursa, metodologija i iskustava projekata MULTEXT na srednjoevropske i istočnoevropske jezike. Zvanično, MULTEXT-East je trajao od 1995. do 1997. godine, ali je nastavio da održava i

proširuje proizvedene specifikacije, jezičke i programske resurse u skladu sa promenama koje su nastupile (uvođenje standarda XML umesto SGML-a, resursi za novopridružene jezike) i da objavljuje nove verzije rezultata 1998, 2002, 2004. i 2010. godine. Resursi za srpski jezik su takođe zastupljeni u resursima koje je proizveo MULTEXT-East ([Krstev et al., 2004])[54].

MULTEXT-East koristi pozicionu notaciju za morfosintaksičke opise (Slika 2.23), tj. svaka pozicija u opisu predstavlja određenu vrstu informacije (vrstu reči, morfološke ili klasifikacione kategorije) koja se naziva atributom. Svaka vrednost atributa je označena nekim slovom engleske abecede ili dekadnom cifrom, dok se crtica (-) na koristi na određenoj poziciji kao specijalna oznaka kojom se naglašava da dati token ne poseduje atribut sa te pozicije. Pošto je notacija poziciona, moguće je upotrebiti istu oznaku na više pozicija, pri čemu značenje oznake zavisi od pozicije na kojoj se nalazi. Npr, oznake u opisu `Afcfsa` za oblik `manju` redom označavaju da je u pitanju pridev (`A`), i to opisni (`f`), u komparativu (`c`), ženskog roda (`f`), u jednini (`s`), u akuzativu (`a`).

2.5 Indeksiranje i kompresija teksta

Jedan od jednostavnijih i neefikasnijih algoritama za pretraživanje elektronskog teksta jeste sekvencijalna pretraga koja se svodi na ispitivanje jednog po jednog tokena u redosledu u kom se pojavljuju u tekstu. U slučaju kolekcija elektronskog teksta ogromnih razmera takav pristup je neefikasan, odnosno značajno usporava odziv sistema za pretragu. Kako bi se popravila brzina i efikasnost pretrage, pribegava se postupku koji se naziva **indeksiranje** (eng. **indexing**).

Indeksiranjem se kreira naročita struktura podataka, **indeks** (eng. **index**), koja omogućava brz i direktan pristup delovima teksta koji su od značaja za pretragu (reči/tokeni, rečenice, odeljci, kompletan tekstuelni dokument). Indeks zahteva određene prostorne kapacitete koji obično prevazilaze veličinu originalnog teksta, a nekoliko desetina najučestalijih reči prirodnog jezika može zauzimati „oko 30%

[54]Neki resursi za srpski jezik su bili dostupni još u resursima projekta TELRI (v. odeljak 6.1 i [Erjavec, Lawson & Romary, 1998]).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TEI SYSTEM "tei_mte.dtd">
<TEI xmlns="http://www.tei-c.org/ns/1.0" xmlns:xsi="http
    ://www.w3.org/2001/XMLSchema-instance">
 <teiHeader>
  ...
 </teiHeader>
 <text>
  <body>
   <div type="chapter" n="37">
    <head>
     ...
    </head>
    <p>
    <seg>
    <w lemma="zaista" ana="Rgp">Zaista</w>
    <c>,</c>
    <w lemma="ko" ana="Pq---sn">ko</w>
    <w lemma="ne" ana="Q-">ne</w>
    <w lemma="biti" ana="Vaca3s-an-n---p">bi</w>
    <w lemma="obići" ana="Vmps-sman-n---e">obišao
    </w>
    <w lemma="svet" ana="Ncmsa---n">svet</w>
    <w lemma="i" ana="C-s">i</w>
    <w lemma="za" ana="Spsa">za</w>
    <w lemma="mali" ana="Afcfsa">manju</w>
    <w lemma="cena" ana="Ncfsa---n">cenu</w>
    <c>?</c>
    </seg>
    </p>
      </div>
    </body>
  </text>
</TEI>
```

Slika 2.23: MULTEXT-East (morfosintaksička anotacija poslednje rečenice srpske verzije romana Žila Verna *Put oko sveta za 80 dana*).

referenci koje indeks koristi" ([Witten et al., 1999: 14]). Iz tog razloga su algoritmi za indeksiranje teksta u uskoj vezi sa algoritmimima za kompresiju podataka. Kompresija podataka je postupak kojim se digitalni podaci transformišu tako da njihova rezultujuća reprezentacija koristi manji broj bitova u odnosu na originalnu reprezentaciju.

Treba naglasiti da indeksiranje i kompresija imaju i svojih nedostataka koji se ogledaju, pre svega, u implementaciji osnovnih operacija sa indeksom kao što su kreiranje, ažuriranje i korišćenje indeksa[55]. Kreiranje indeksa zahteva značajno vreme, a različiti programi za indeksiranje najčešće nisu međusobno kompatibilni, tako da po kreiranju indeks mogu da koriste samo određeni programski paketi, tj. oni koji sadrže biblioteku potprograma pomoću koje je indeks napravljen. U pojedinim slučajevima, kada se kolekcija teksta izmeni dodavanjem novih ili brisanjem postojećih dokumenata, jedini način da se indeks ažurira jeste da se ponovo napravi, što znači da je i ažuriranje indeksa vremenski zahtevan proces. Pored toga, ako je indeks ogromnih razmera, on se mora komprimovati, a pre korišćenja i dekomprimovati, što zahteva dodatno vreme i odgovarajuće performanse računarskog sistema koji će izvršavati algoritme kompresije i dekompresije.

Detaljni pregled algoritama za indeksiranje i kompresiju elektronskog teksta i slika je dat u [Witten et al., 1999]. Ovde će se ukratko razmotriti samo problematika indeksiranja teksta za elektronske korpuse.

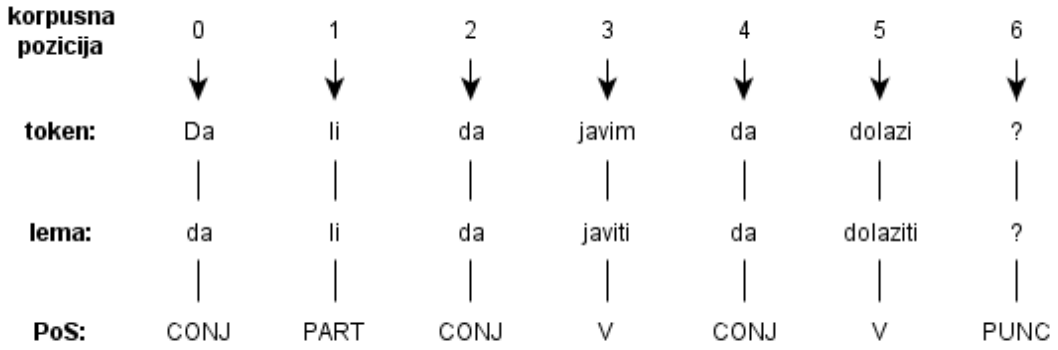
Radi efikasnijeg pretraživanja, neophodno je obezbediti direktan pristup svakom tokenu elektronskog korpusa, odnosno napraviti **indeks kompletnog teksta** (eng. **full-text index**). U tu svrhu se tekst korpusa posmatra ne kao niz karaktera, već kao niz pozicija tokena u korpusu. Pozicija tokena u korpusu se još naziva **korpusna pozicija** (eng. **corpus position**) i obično se realizuje kao nenegativan[56] ceo broj koji predstavlja redni broj tokena u korpusu.

Svakoj korpusnoj poziciji se pridružuje određen broj **pozicionih atributa** (eng. **positional attribute**). Prvi pozicioni atribut je obavezan i on predstavlja vrednost

[55] Na adresi `http://webdeptos.uma.es/filifa/spanish/profesorado/personal/amoreno/teaching/cl/ctools.html` su detaljnije razmotreni navedeni nedostaci indeksiranja i kompresije teksta, kao i alati za indeksiranje i kompresiju teksta (datum pristupa 29. maj 2013. godine.

[56] To obično znači da je prva korpusna pozicija označena nulom, druga jedinicom, itd.

tokena na odgovarajućoj korpusnoj poziciji. Ostali pozicioni atributi, ako postoje, mogu se iskoristiti kao elementi anotacije, tj. njihove vrednosti mogu biti neke dodatne informacije koje se pridružuju tokenu poput njegove leme, vrste reči, identifikatora teksta u korpusu koji sadrži taj token, itd. (Slika 2.24).



Slika 2.24: Neka se rečenica `Da li da javim da dolazi?` nalazi na početku anotiranog korpusa, u kome su svakom tokenu pridružene informacije o lemi i vrsti reči. Tada tokeni rečenice zauzimaju korpusne pozicije numerisane brojevima od 0 do 5, a svaka korpusna pozicija ima tri pozicionа atributa čije su vrednosti token, njegova lema i njegova vrsta reči (PoS). Vrednosti CONJ, PART, V atributa PoS redom označavaju vrste reči veznik, rečcu (partikulu), glagol, dok PUNC predstavlja oznaku znaka interpunkcije).

U slučaju ogromnih pretraživih kolekcija tekstova, poput elektronskih korpusa, najpogodnija struktura za predstavljanje indeksa je **invertovani indeks** (eng. **inverted index**) ili **invertovana datoteka** (eng. **inverted file**).

Invertovani indeks se koristi zajedno sa **leksikonom** (eng. **lexicon**) — skupom različitih termina koji se pojavljuju u tekstu. Za svaki termin leksikona se formira **invertovana lista** (eng. **inverted list**), tj. lista pokazivača na sva pojavljivanja termina u tekstu.

U slučaju indeksiranja teksta za korpus, leksikon se sastoji od različitih tokena koji se pojavljuju u korpusu, a invertovana lista je lista korpusnih pozicija na kojima se pojavljuje token, na čijem početku se navodi ukupan broj korpusnih pozicija u listi. Prema tome, invertovani indeks je zapravo skup trojki oblika $(t, |l_t|, l_t)$, pri čemu je:

- t element leksikona (termin, odnosno, token),
- l_t je invertovana lista za token t,
- a $|l_t|$ je dužina invertovane liste l_t, odnosno broj pojavljivanja (frekvencija, učestanost) tokena t.

Da bi se omogućila kompresija indeksa, kao i efikasno konsultovanje leksikona i samog korpusa, elementi leksikona i korpusa se kodiraju pomoću celih brojeva. Na taj način se na osnovu reprezentacije leksikona i korpusa, kao niza tokena u tekstuelnoj datoteci, konstruišu **celobrojne reprezentacije** (eng. **integerized representation**) leksikona i korpusa kao niza nenegativnih celih brojeva u binarnoj datoteci, pri čemu različitim pojavljivanjima istog tokena odgovara isti ceo broj, a različitim tokenima odgovaraju različiti celi brojevi. Drugim rečima između skupa tokena leksikona i skupa nenegativnih celih brojeva uspostavlja se jedno $1-1$ preslikavanje, celobrojni kôd tokena, koje se potom koristi za konstruisanje binarne celobrojne reprezentacije korpusa.

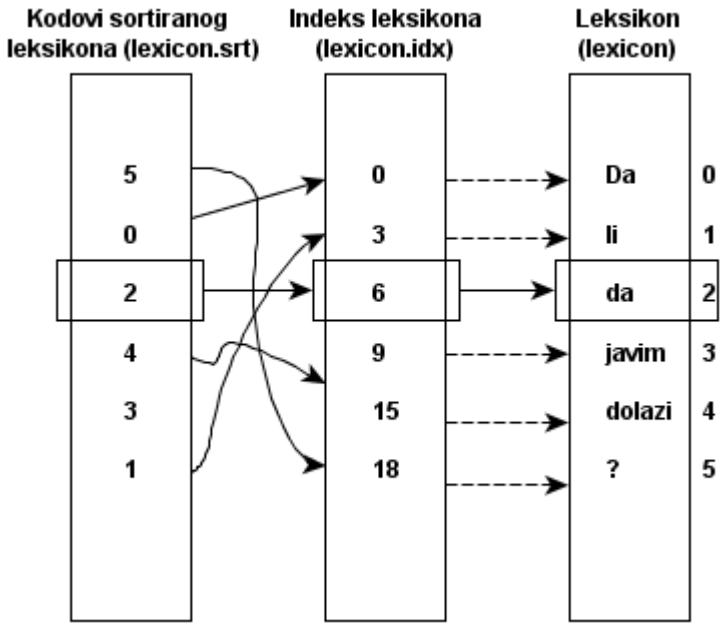
Implementaciju pozicionih atributa, invertovanog indeksa, leksikona i celobrojne reprezentacije korpusa ilustrovaćemo na primeru jedne od ranih verzija alata IMS CWB ([Christ, 1994b]).

Jednim prolazom kroz datoteku `corpus.vrt` sa tekstuelnom reprezentacijom korpusa konstruišu se leksikon (datoteka `lexicon`) i celobrojna reprezentacija korpusa (datoteka `corpus`). Leksikon predstavlja niz L različitih tokena korpusa razdvojenih nula-bajtom[57]. Neka je $|L|$ oznaka za broj elemenata niza L (broj različitih tokena u korpusu) i $L[i]$ oznaka za token na poziciji i u leksikonu, pri čemu je $0 \leq i < |L|$. U celobrojnoj reprezentaciji korpusa svaki token je predstavljen svojim rednim brojem u leksikonu ($0 \leq i < |L|$). Oznaka $f[i]$, $0 \leq i < |L|$ se u daljem tekstu koristi za ukupan broj pojavljivanja tokena $L[i]$ u korpusu.

Dve operacije koje su neophodne prilikom korišćenja celobrojne reprezentacije korpusa su izračunavanje celobrojnog koda zadatog tokena i određivanje tokena kome odgovara zadati celobrojni kôd. Kako se te operacije često izvršavaju, radi njihove efikasnije implementacije konstruišu se još dve pomoćne binarne datoteke

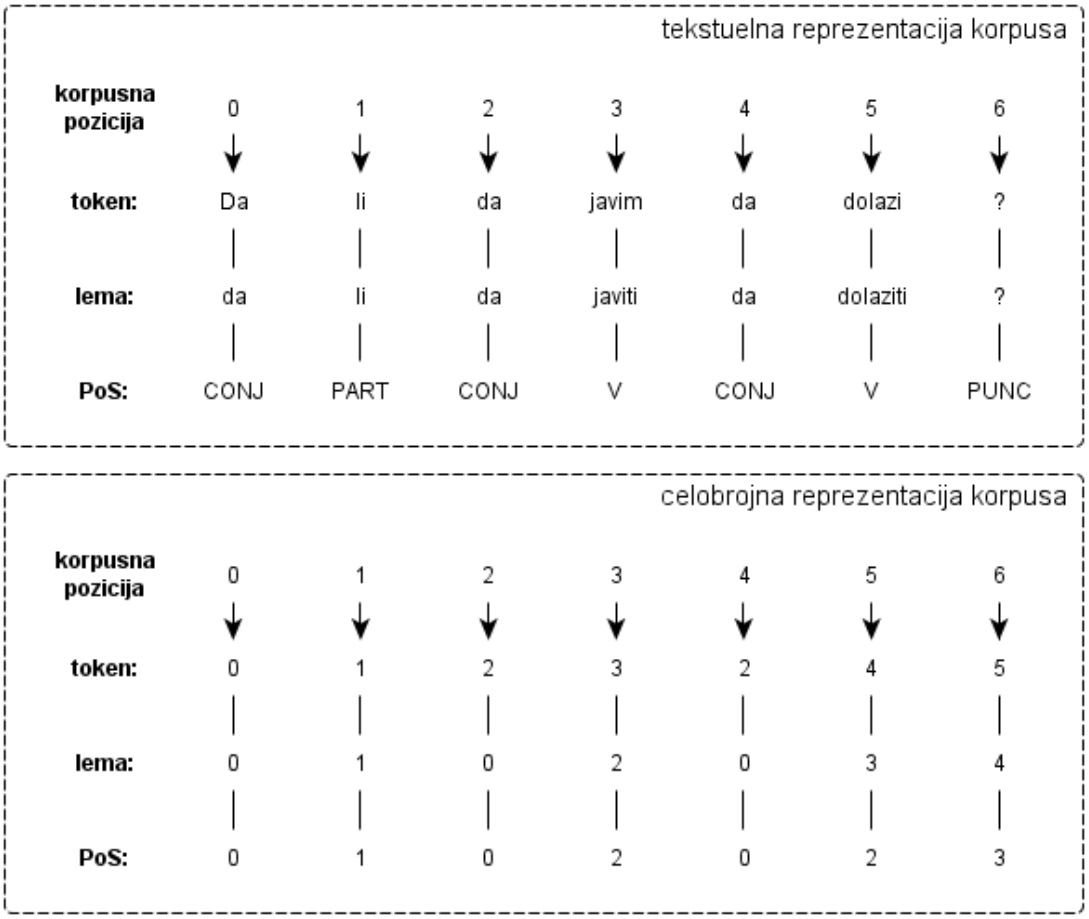
[57] U pitanju je prvi ASCII-karakter '\0' čiju bitovsku reprezentaciju sačinjavaju samo nule.

za konsultovanje leksikona (Slika 2.25). Prva pomoćna datoteka (`lexicon.srt`) predstavlja niz celobrojnih kodova tokena sortiranog leksikona (u oznaci S_L). U drugoj pomoćnoj datoteci (`lexicon.idx`) nalazi se niz pokazivača na tokene leksikona (u oznaci I_L), pri čemu redosled pokazivača odgovara redosledu tokena u leksikonu. Pokazivač na token t leksikona je zapravo *pomeraj* (eng. *offset*) u odnosu na početak datoteke `lexicon`, izražen u bajtovima.



Slika 2.25: Postupak određivanja celobrojnog kôda za token `da`.

Određivanje celobrojnog koda tokena se svodi na pronalaženje pozicije na kojoj se token nalazi u nizu L leksikona. Umesto da se leksikon (koji ne mora biti sortiran) direktno pretražuje, koristi se binarna pretraga datoteke `lexicon.srt`. Ako je k, $0 \leq k < |L|$, indeks središnjeg elementa u tekućem pretraživanom podnizu datoteke `lexicon.srt`, zadati token se ne može upoređivati sa elementom niza $S_L[k]$ (celobrojnim kodom tokena sortiranog leksikona), već je potrebno odrediti element leksikona čiji je celobrojni kôd jednak $S_L[k]$. U tu svrhu se koristi datoteka `lexicon.idx`, tj. zadati token se upoređuje sa elementom leksikona čiji je pomeraj jednak $I_L[S_L[k]]$. Opisanim postupkom se za token `da` utvrđuje da je njegov celobrojni kôd broj 2 (Slika 2.25).



Slika 2.26: Tekstuelna i celobrojna reprezentacija korpusa: prvi red predstavlja korpusne pozicije, dok drugi, treći i četvrti red predstavljaju vrednosti pozicionih atributa (token, lema i vrsta reči).

Pronalaženje svih pozicija na kojima se u korpusu pojavljuje zadati token realizuje se pomoću četiri datoteke:

- datoteka `corpus` sadrži celobrojnu reprezentaciju korpusa kao niz C celobrojnih kodova tokena, tj. $C[i]$ je celobrojni kôd tokena na korpusnoj poziciji i (Slika 2.26);
- datoteka `corpus.cnt` sadrži niz $f[i]$, $0 \leq i < |L|$, tj. niz učestanosti pojedinačnih tokena u korpusu u istom redosledu u kom se ti tokeni pojavljuju kao elementi leksikona. Dakle, $f[i]$ je učestanost tokena $L[i]$ čiji je celobrojni kôd i;

- datoteka `corpus.rev` predstavlja jednodimenzionalni niz R kao uniju podnizova $t[i][-]$, $0 \leq i < |L|$. Podniz $t[i][-]$ za fiksiranu vrednost i predstavlja rastući niz svih korpusnih pozicija na kojima se pojavljuje token $L[i]$[58], a dužina podniza je zapravo učestanost tokena $L[i]$, tj. $f[i]$. Proizvoljni element niza R može se opisati sledećom formulom:

$$R[\sum_{k=0}^{i-1} f[k] + j] = t[i][j] \ (0 \leq i < |L|, 0 \leq j < f[i]).$$

  Nizovi C i R su iste dužine, tj. $|C|{=}\sum_{0 \leq i < |L|} f[i] = |R|$;

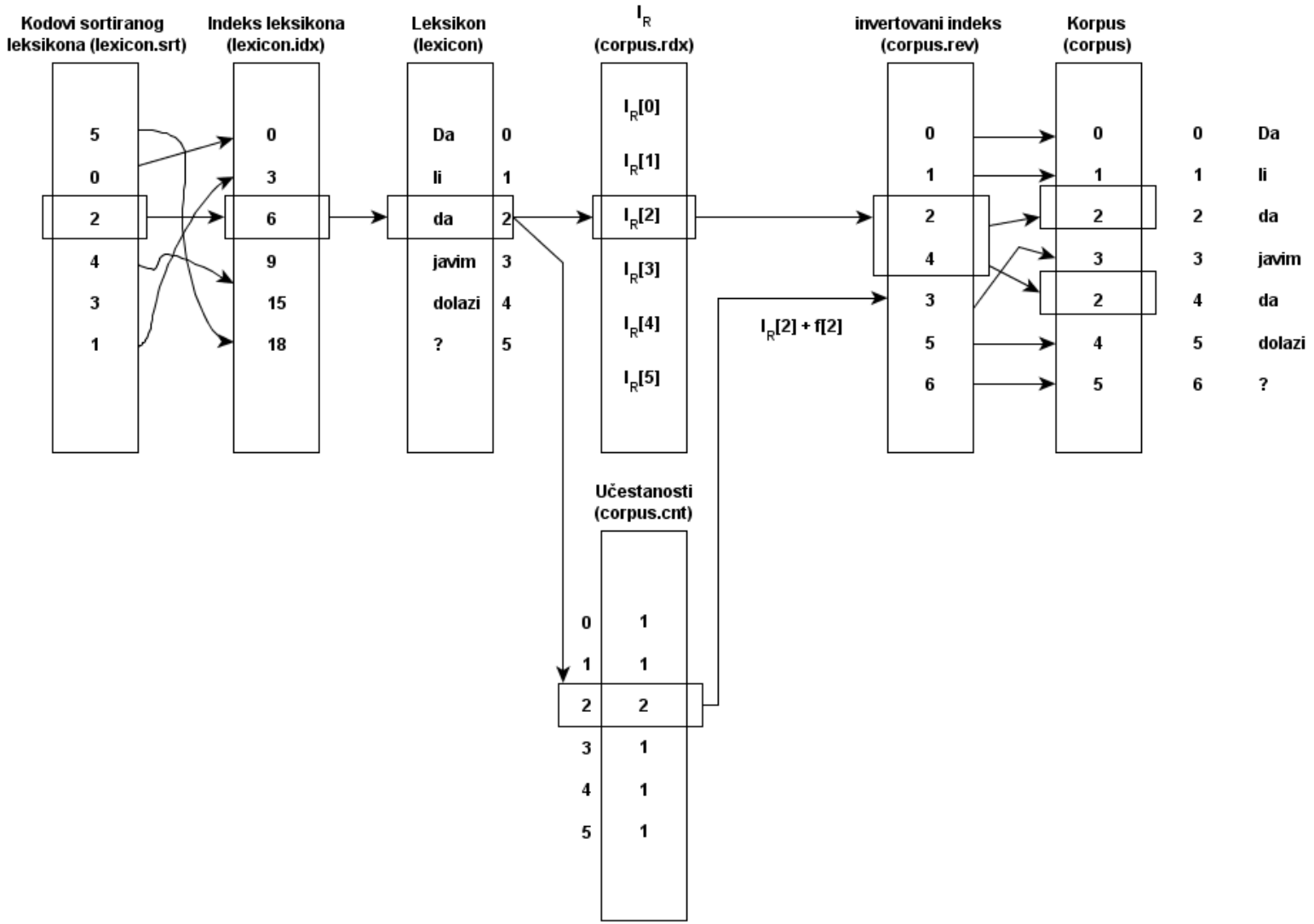
- datoteka `corpus.rdx` sadrži niz pokazivača I_R na pozicije u datoteci `corpus.rev`. Ako je i celobrojni kôd tokena $L[i]$, $0 \leq i < |L|$, tada je $I_R[i]$ pomeraj (u bajtovima) u odnosu na početak datoteke `corpus.rev` do početka rastućeg podniza korpusnih pozicija na kojima se nalazi celobrojni kôd i tokena $L[i]$. Dakle, $I_R[i]$ je pokazivač na početak podniza korpusnih pozicija $t[i][-]$.

Navedenim postupkom se na osnovu celobrojnog koda tokena `da` (broj 2) najpre određuju vrednost pokazivača $I_R[2]$ na podniz $t[2][-]$ korpusnih pozicija na kojima se taj token pojavljuje, kao i učestanost tokena `da`, tj. $f[2]$ (Slika 2.27). Vrednosti $t[2][0] = 2$ i $t[2][1] = 4$ predstavljaju korpusne pozicije na kojima se pojavljuje celobrojni kôd tokena `da`, tj. indekse elemenata niza C čija je vrednost kôd tokena `da` (broj 2).

U programskom sistemu IMS CWB svaki pozicioni atribut je imenovan, pri čemu je naziv prvog pozicionog atributa uvek `word`, dok ostale pozicione atribute imenuje administrator korpusa (Slika 2.26). Za svaki pozicioni atribut se posebno konstruiše sedam datoteka koje omogućavaju korišćenje leksikona i invertovanog indeksa, pri čemu se u njihovom nazivu navodi i ime atributa kao prefiks. Tako se za atribut `word` konstruišu sledeće datoteke:

- `word.lexicon`,
- `word.lexicon.idx`,

[58] Podniz $t[i][j]$ za fiksiranu vrednost celobrojnog koda i predstavlja rastući niz svih pozicija na kojima se u nizu C pojavljuje kôd i.



Slika 2.27: Postupak određivanja korpusnih pozicija na osnovu celobrojnog kôda za token `da`.

- `word.lexicon.srt`,
- `word.corpus`,
- `word.corpus.cnt`,
- `word.corpus.rev`,
- `word.corpus.rdx`,

a analogna imena imaju datoteke ostalih pozicionih atributa.

3

Pretraga i analiza korpusa

Sav trud oko kreiranja korpusa kao reprezentativnog uzorka jezika je uzaludan ukoliko ne postoji mogućnost pretrage i analize njegovih tekstova. U ovom odeljku će najpre biti razmotreni regularni izrazi kao notacija za zadavanje upita nad korpusom, kao i konkordance kao format za predstavljanje rezultata pretrage. Na kraju će posebno biti razmotrene mogućnosti statističke analize korpusa, pre svega liste učestanosti i n-grami.

3.1 Regularni izrazi u obradi i pretrazi teksta

Regularni izrazi u računarstvu: kratka istorija

Jedna od prvih praktičnih primena regularnih izraza u računarstvu, ugradnja kompilatora regularnih izraza u programe za uređivanje teksta, opisana je u radu Kena Tompsona ([Thompson, 1968]). Korisnicima programa za uređivanje teksta je tako omogućeno da prilikom pretrage zadaju **obrasce** (eng. **pattern**) u formi regularnih izraza i da kao rezultat dobiju linije teksta koje zadovoljavaju zadate obrasce. Komanda kojom su korisnici zadavali obrazac postala je poznata pod nazivom *grep*[1]

[1]Sintaksa komande kojom su korisnici zadavali obrasac u prvoj verziji imala je oblik `g/regular expression/p` (srp. `g/regularni izraz/p`), što su korisnici tumačili kao *Global Regular Expression Print*, odnosno skraćeno *grep*. Između kosih crta korisnici su regularnim izrazom zadavali obrazac, opcijom `g(lobal)` su zahtevali sve pojave obrasca u tekstu, dok je uloga opcije `p(rint)`

i postala je sastavni deo operativnog sistema Unix u čijoj je izradi Tompson takođe učestvovao.

Mnogi programski alati (*sed*, *tr*, *flex*, *yacc*, *awk*, itd.) i jezici (Perl, PHP, Java, C#, Python, Ruby, i dr.) takođe su prepoznali pogodnosti regularnih izraza u obradi teksta, pa su obezbedili opcije, operatore i funkcije za proveru da li neka niska pripada jeziku koga opisuje zadati regularni izraz. Takođe, opcija pretraživanja i modifikacije (automatske zamene) teksta regularnim izrazima je postala obavezna u **integrisanim razvojnim okruženjima** (eng. **Integrated Development Environment**, skr. **IDE**) i naprednim programima za uređivanje teksta (Microsoft Word, Emacs, *vi*, PSPad, Notepad++, UltraEdit, XML Spy, WinEdt, Microsoft Visual Studio, SharpDevelop i mnogi drugi).

Od prve verzije komande *grep* do danas sintaksa i semantika regularnih izraza se menjala i proširivala, a gotovo svaki programski jezik ili program za uređivanje teksta koji je omogućio njihovo korišćenje je unosio po neki dodatak ili izmenu. Počev od 1986. godine, pojavom standarda **Prenosivo sučelje operativnog sistema** (eng. **Portable Operating System Interface**, skr. **POSIX**), započinju pokušaji da se standardizuju sintaksa i semantika regularnih izraza. POSIX je pokušao da sve popularne mogućnosti regularnih izraza reorganizuje definišući dve klase regularnih izraza, **osnovne regularne izraze** (eng. **Basic Regular Expressions**, skr. **BREs**) i **proširene regularne izraze** (eng. **Extended Regular Expressions**, skr. **EREs**). Aplikacije koje koriste regularne izraze i implementiraju POSIX-standard uglavnom podržavaju jednu od tih dveju klasa regularnih izraza.

Osim standarda POSIX, danas su najuticajnije međusobno različite implementacije regularnih izraza programskih jezika Perl, Java, kao i jezika koje obuhvata .NET (C#, C++/CLI, Visual Basic, J#, F#, itd.).

Istorijat regularnih izraza, priručnik za upotrebu i uporedni pregled njihove sintakse, semantike u okviru različitih implementacija (*grep*, *egrep*, Perl, Java, .NET) detaljno su opisani u [Friedl, 2002], dok se u ([Jurafsky & Martin, 2008: dodatak A]) može naći uporedni pregled sintakse regularnih izraza koji (uz Perl i *grep*) uključuje i Microsoft Word.

bila ispisivanje linija koje su zadovoljavale obrazac.

POSIX: prošireni regularni izrazi

U ovom odeljku biće opisana sintaksa i semantika proširenih regularnih izraza definisanih standardom POSIX 1003.2. Upravo te regularne izraze koristi većina upitnih jezika savremenih alata za obradu i pretragu korpusa, među njima i **Korpusni upitni jezik** (eng. **Corpus Query Language**, skr. **CQL**) o kom će biti više reči u poglavlju 7.

Azbuka nad kojom se definišu POSIX-regularni izrazi zavisi od **lokalnih podešavanja operativnog sistema** (eng. **locale**). Ova podešavanja opisuju jezik i konvencije koje se odnose na zapis datuma, časovnog vremena, novčanih jedinica, interpretaciju karaktera tekućeg kodnog rasporeda, itd. Uloga lokalnih podešavanja operativnog sistema je da omoguće internacionalizaciju različitih aplikacija, a ne samo regularnih izraza. Kada su regularni izrazi u pitanju, uticaj lokalnih podešavanja se svodi na svojstva karakterskog skupa koji odgovara aktivnom (tekućem) kodnom rasporedu. Međutim, ako je aktivan bilo koji od kodnih rasporeda opisanih u odeljku 2.2 (izuzimajući YUSCII), koji se koristi za čuvanje elektronskog teksta na srpskom, azbuka regularnih izraza sadrži karakterski skup ASCII kao podskup.

Elementi azbuke POSIX-regularnih izraza se dele na dve grupe: **obične karaktere** ili **literale** (eng. **literal**) i **metakaraktere** (eng. **metacharacter**). Dok literali predstavljaju sami sebe, dotle metakarakteri imaju specijalno značenje. Iako su metakarakteri `|` i `*`, kojima se označavaju regularne operacije, dovoljni da opišu svaki regularni skup, POSIX uvodi dodatne metakaraktere kojima se olakšava i skraćuje zapis regularnih izraza.

Značenje POSIX-metakaraktera zavisi od tipa obrasca u okviru kog se koriste. Sledi pregled važnijih tipova obrazaca:

Unija (disjunkcija, alternacija). Ovaj tip obrasca i metakarakter `|` opisani su u Definiciji 2.6, tačka (iv) i ilustrovani primerima 2.2–2.4.

Dopisivanje (proizvod, konkatenacija). Ovaj tip obrasca, opisan u Definiciji 2.6, tačka (v), ne koristi nijedan metakarakter.

Klasa karaktera i negativna klasa karaktera. Klasa karaktera `[...]` i negativna klasa karaktera `[^...]` koriste metakaraktere `[` i `]`, tj. srednje zagrade, dok negativna klasa karaktera koristi i metakarakter `^`.

Klasa karaktera omogućava pogodniji zapis konačne unije skupova čiji su elementi niske dužine 1, tj. karakteri. Ako je n pozitivan ceo broj, $c_1, c_2, \ldots, c_n$ karakteri i c_1 nije metakarakter `^`, tada je `[`$c_1c_2\cdots c_n$`]` regularni izraz (klasa karaktera) koji označava regularni skup $\{c_1, c_2, \ldots, c_n\}$.

Primer 3.1. Regularni izraz `[aeiou]` označava regularan skup koji sadrži pojedinačne karaktere navedene u klasi, tj. skup $\{a, e, i, o, u\}$, pa je ekvivalentan regularnom izrazu `(a|e|i|o|u)`. Međutim, skup $\{pismo, pisma\}$ iz Primera 2.4 ne može se predstaviti klasom karaktera jer su njegovi elementi niske dužine veće od 1.

Karakter `-` (ASCII-crtica) isključivo unutar klase karaktera dobija specijalno značenje, ali samo ako se ne nađe ni na početku ni na kraju klase karaktera, tj. uz srednje zagrade. Naime, crticom se definiše opseg ili interval karaktera (kolaciona sekvenca), tj. zapis c_1`-`c_2 predstavlja skup svih karaktera čija je kodna pozicija veća ili jednaka od kodne pozicije karaktera c_1 i istovremeno manja od kodne pozicije karaktera c_2.

Primer 3.2. Crtica omogućava kondenzovani zapis klase karaktera što je posebno korisno kada su u pitanju često korišćeni podskupovi: dekadne cifre, velika slova engleskog alfabeta i mala slova engleskog alfabeta. Naime, karakteri ovih podskupova karakterskog skupa ASCII zauzimaju susedne kodne pozicije (v. Sliku 2.2, strana 73). Stoga se dekadne cifre (kodne pozicije 48–57) mogu predstaviti regularnim izrazom `[0-9]`, velikim slovima engleskog alfabeta (kodne pozicije 65–90) odgovara regularni izraz `[A-Z]`, a malim slovima engleskog alfabeta (kodne pozicije 97–122) regularni izraz `[a-z]`. Primetimo da regularni izraz `[A-z]` ne predstavlja samo proizvoljno (veliko ili malo) slovo engleskog alfabeta, već i karaktere sa pozicija 91–96 koji nisu slova; korektan regularni izraz za slovo engleskog alfabeta je `[A-Za-z]` ili `[a-zA-Z]`.

Ukoliko se klasom karaktera želi predstaviti skup karaktera među kojima je i crtica (kao literal), ona se mora navesti ili kao prvi ili kao poslednji karakter u klasi.

Primer 3.3. Regularni izraz `[az-]` predstavlja skup $\{a, z, -\}$.

Većina metakaraktera unutar klase karaktera gubi svoje specijalno značenje, izuzetak su jedino `]`, `\` i crtica kada se ne nalazi na početku ili kraju klase.

Prvi karakter klase karaktera ne sme biti metakarakter `^` jer se u protivnom radi o negativnoj klasi karaktera.

Svakoj klasi karaktera odgovara negativna klasa karaktera koja se dobija dodavanjem metakaraktera `^` posle otvorene srednje zagrade. Ako klasa karaktera označava skup karaktera X, tada negativna klasa karaktera označava komplement skupa X u odnosu na azbuku, odnosno tekući karakterski skup.

Primer 3.4. Klasi karaktera `[aeiou]` odgovara negativna klasa karaktera `[^aeiou]` koja označava skup svih karaktera tekućeg karakterskog skupa izuzev karaktera `a, e, i, o, u`.

U okviru negativne klase karaktera takođe se može koristiti crtica za zadavanje opsega karaktera.

Primer 3.5. Regularni izraz `[^a-z]` predstavlja skup svih karaktera tekućeg kodnog rasporeda od kojih nijedan nije malo slovo engleskog alfabeta.

Kvantifikovanje. Kvantifikovanje koristi metakaraktere `*`, `+`, `?`, `{` i `}`.

Kvantifikator `*` je metakarakter kojim se označava Klinijevo zatvorenje. Kao što ilustruje Primer 2.5, metakarakter `*` označava da se karakter ili izraz koji mu prethodi u obrascu pojavljuje nula ili više puta.

Na sličan način, kvantifikator `+` je metakarakter koji označava da se karakter ili izraz koji mu prethodi u obrascu pojavljuje jednom ili više puta. Ako je `r` regularni izraz, tada je `r+` samo kraći zapis za `rr*`. Ako je X skup označen regularnim izrazom `r`, skup kome odgovara regularni izraz `r+` se još naziva **pozitivno zatvorenje** skupa X i označava sa X^+.

Primer 3.6. Ako je B regularni skup iz Primera 2.5, tada regularni izraz `u*` označava skup $B^* = \{\varepsilon, u, uu, uuu, \ldots\}$, dok `u+` predstavlja regularni skup $B^* \setminus \{\varepsilon\} = \{u, uu, uuu, \ldots\} = BB^* = B^+$.

Kvantifikator `?` je metakarakter kojim se označava opciono pojavljivanje karaktera ili izraza koji mu prethodi u obrascu, tj. prethodni karakter ili izraz može imati jedno ili nijedno pojavljivanje. Ako je `r` regularni izraz koji označava skup X, tada regularni izraz `r?` označava skup $X \cup \{\varepsilon\}$.

Primer 3.7. Regularni izraz `h?leb` označava skup $\{hleb, leb\}$.

Neka su `m` i `n` pozitivni celi brojevi. Kvantifikatori `{n}`, `{n,m}` i `{n,}` redom označavaju da se karakter ili izraz koji im prethodi u obrascu pojavljuje tačno `n` puta (`{n}`), između `n` i `m` puta (`{n,m}`), odnosno bar `n` puta (`{n,}`).

Primer 3.8. Ako je `u` element regularnog skupa B iz Primera 2.5, tada:

- `u{4}` označava regularni skup $\{uuuu\}$;
- `u{2,5}` označava regularni skup $\{uu, uuu, uuuu, uuuuu\}$;
- `u{5,}` označava regularni skup $\{uuuuu, uuuuuu, uuuuuuu, \ldots\}$.

„Džoker-znak". Metakarakter `.` predstavlja regularni izraz kojim se opisuje proizvoljan karakter, izuzimajući znak za novi red.

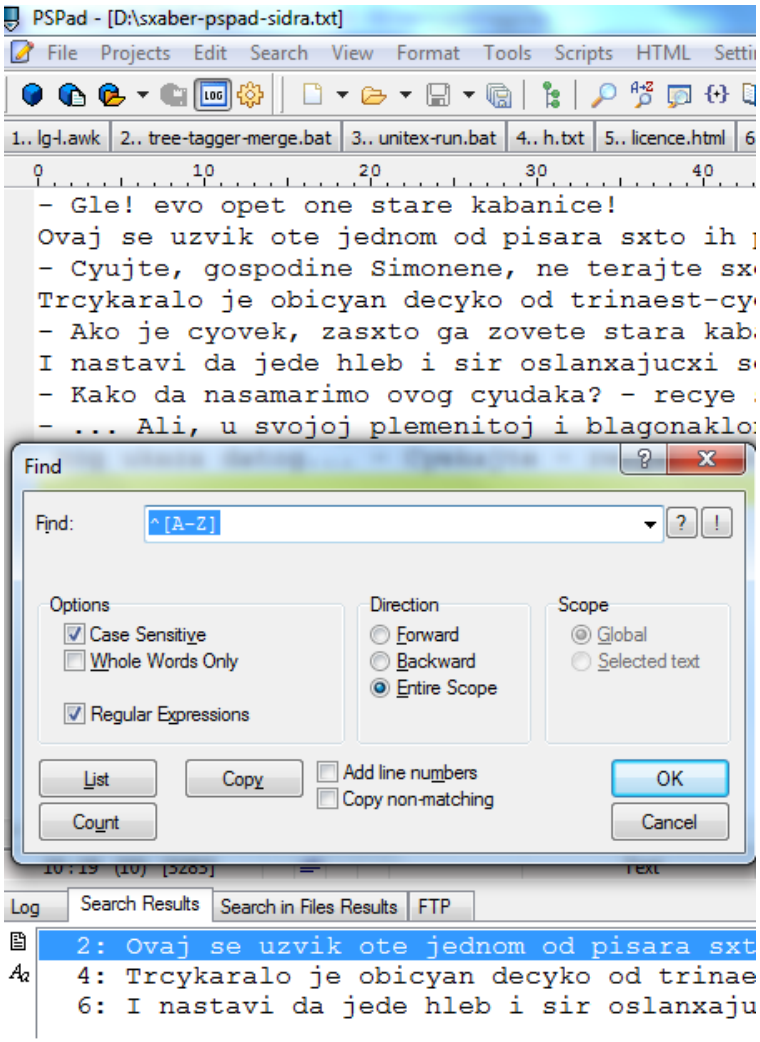
Primer 3.9. Regularni izraz `.{2}` predstavlja skup svih niski dužine 2 nad tekućim karakterskim skupom izuzev onih koje sadrže znak za novi red.

Sidra. Sidra su predstavljena metakarakterima `^` i `$`[2]. U programskim jezicima i programima za uređivanje teksta koji koriste regularne izraze, prilikom pretrage se češće ispituje da li neki povezani deo niske (podniska) odgovara zadatom regularnom izrazu[3]. U tom slučaju sidra omogućavaju da se niske odgovarajućeg regularnog skupa preciznije opišu na pozicijama kao što su početak (`^`) i kraj (`$`) niske.

[2] U nekim klasifikacijama i „džoker-znak" `.` se tretira kao sidro.

[3] Prilikom testa da li neku podnisku zadate niske opisuje dati regularni izraz, može se desiti da na istoj poziciji počinje više različitih podniski koje odgovaraju datom opisu. Na primer, ako se u okviru niske `Godina␣2011.` traži podniska koja odgovara regularnom izrazu `[0-9]+`, tada na poziciji 8 (karakter `2`) počinju četiri podniske koje odgovaraju tom regularnom izrazu: `2`, `20`, `201`, `2011`. Većina implementacija regularnih izraza koristi **pohlepni algoritam** (eng. **greedy algorithm**) koji kao rešenje navodi **prvu sleva najdužu uparenu podnisku** (eng. **leftmost longest match**) koju opisuje regularni izraz (u ovom slučaju `2011`).

Primer 3.10. U programu za uređivanje teksta PSPad (Slika 3.1) regularni izraz `^[A-Z]` pronalazi sve linije koje počinju velikim slovom engleskog alfabeta, regularni izraz `!$` pronalazi sve linije koje se završavaju uzvičnikom, dok regularni izraz `^[A-Z].*!$` pronalazi sve linije koje imaju oba svojstva.



Slika 3.1: Rezultat pretrage regularnim izrazom `^[A-Z]` u programu PSPad)

Ovde treba primetiti da metakarakter `^` može imati dva različita značenja, kao sidro i kao negacija klase karaktera. S obzirom da su konteksti u kojima se realizuju ova dva značenja jasno razdvojeni navođenjem metakaraktera unutar ili izvan srednjih zagrada, ne može doći do zabune, tako da se u istom regularnom izrazu `^` može pojaviti u oba značenja.

Primer 3.11. Regularni izraz `^[^A-Z]` u programu PSPad pronalazi sve linije koje počinju karakterom koji nije veliko slovo engleskog alfabeta.

Alati za obradu i pretragu korpusa uglavnom pretražuju korpus kao niz prethodno izdvojenih tokena. Iz tog razloga sidra `^` i `$` nisu neophodna, pa ih neki upitni jezici (na primer, CQL, v. poglavlje 7) ne koriste kao metakaraktere. Međutim, ako takvi upitni jezici koriste biblioteke regularnih izraza koje poštuju standard POSIX (kao što to čini CQL), simboli `^` i `$` i dalje imaju specijalno značenje, te ako je potrebno da se u regularnom izrazu pojave kao literali, njihovo specijalno značenje treba eliminisati metakarakterom \ (v. pododeljak „Vračara”.).

„Vračara”. Metakarakter \ se navodi ispred pojedinačnog karaktera i koristi se na dva načina:

1. metakarakterima[4]

   ```
   | * + ?  { } [ ] . ^ $ ( ) \
   ```

   uklanja specijalno značenje, tako da predstavljaju samo sebe kao obične karaktere. Na primer, `\.` predstavlja običnu tačku, tj. samo karakter `.` i nijedan drugi, a analogno se ponašaju i sekvence

   ```
   \| \* \+ \?  \{ \} \[ \] \^ \$ \( \) \\;
   ```

2. pojedinim karakterima daje specijalno značenje. Tako `\t` i `\n` redom predstavljaju tabulator i znak za novi red.

Dakle, metakarakter \ se ponaša kao **„vračara”** (eng. **escape character**), tj. uklanjanjem i dodeljivanjem specijalnog značenja karakterima „skida i baca čini”. Sekvence karaktera koje počinju metakarakterom \ i imaju stoga posebno značenje nazivaju se **metasekvence** (eng. **metasequence, escaped character, escape sequence**)[5].

[4] Prilikom nabrajanja metakaraktera je kao separator korišćen razmak umesto zapete radi bolje preglednosti.

[5] *Escaped character* se radije koristi u prvom slučaju (uklanjanje specijalnog značenja metakarakterima), dok se *escape sequence* češće koristi za dodeljivanje specijalnog značenja karakterima (`\t`, `\n`, itd.)

Grupisanje i prioritet. Kao što je već primećeno na strani 91, neophodno je uvesti prioritet operacija označenih nabrojanim metakarakterima, kako bi se svaki regularni izraz jednoznačno tumačio (Tabela 3.1).

Tabela 3.1: Lista metakaraktera u regularnim izrazima prema opadajućem prioritetu

Obrasci	**Metakarakteri**
Klase karaktera	`[...-...] [^...-...]`
Metasekvence	`\`
Grupisanje (zagrade)	`()`
Kvantifikatori	`* + ?  {}`
Dopisivanje	
Sidra	`^ $`
Unija (alternacija)	`|`

Obične zagrade `(` i `)` omogućavaju zaobilaženje prioriteta grupisanjem odgovarajućih delova regularnog izraza.

Primer 3.12. Ako je potrebno odrediti regularni izraz koji bi opisao skup A^* iz Primera 2.5, `ha*` bi bilo nekorektno rešenje. Naime, `ha*` ima isti efekat kao i `h(a)*`, pošto kvantifikator `*` ima veći prioritet u odnosu na dopisivanje, tako da `ha*` odgovara skupu

$$C = \{h, ha, haa, haaa, \ldots\} \neq \{\varepsilon, ha, haha, hahaha, \ldots\} = A^*.$$

Jedno moguće rešenje za opis skupa A^*, `(ha)*`, koristi zagrade da bi izbeglo prioritet.

Primer 3.13. Pretpostavimo da regularnim izrazom treba predstaviti skup rednih brojeva dana u mesecu, ili skup rednih brojeva meseci u godini, pri tome ne koristeći uniju kad god je moguće koristiti klasu karaktera. Najčešća početnička greška su „rešenja" `[1-31]`, odnosno `[1-12]`, koja zapravo predstavljaju skupove $\{1, 2, 3\}$, odnosno {1,2}. Ovakva „rešenja" se biraju zbog nekorektnog poistovećivanja niski poput `12` i `31` sa karakterima, odnosno zbog pogrešno uspostavljene analogije između intervala rednih brojeva i intervala kodnih pozicija karaktera. Posle pažljive analize definicije 2.6 i tvrđenja 2.1 skup rednih brojeva dana u mesecu može se predstaviti

na sledeći način:

$$\begin{aligned}
\{1,2,\ldots,31\} &= \{1,2,\ldots,9\} \cup \\
&\cup \{10,11,12,\ldots,19\} \cup \\
&\cup \{20,21,22,\ldots,29\} \cup \\
&\cup \{30,31\} = \\
&= \{1,2,\ldots,9\} \cup \\
&\cup \{1\}\{0,1,2,\ldots,9\} \cup \\
&\cup \{2\}\{0,1,2,\ldots,9\} \cup \\
&\cup \{3\}\{0,1\} = \\
&= \{1,2,\ldots,9\} \cup \\
&\cup (\{1\} \cup \{2\})\{0,1,2,\ldots,9\} \\
&\cup \{3\}\{0,1\} = \\
&= \{1,2,\ldots,9\} \cup \{1,2\}\{0,1,2,\ldots,9\} \cup \{3\}\{0,1\}.
\end{aligned}$$

Odatle neposredno sledi da se proizvoljan redni broj dana u mesecu može opisati regularnim izrazom `([1-9])|([12][0-9])|(3[01])`, odnosno, ako se izostave zagrade (pošto klasa karaktera i dopisivanje imaju veći prioritet od unije), izrazom `[1-9]|[12][0-9]|3[01]`. Slično, proizvoljan redni broj meseca u godini može se opisati regularnim izrazom `[1-9]|1[0-2]`.

Primena regularnih izraza u obradi teksta

Tokenizacija U odeljku 2.3, pododeljak Tokenizacija, primerima 2.6-2.8 je ilustrovano kako izbor separatorskog skupa utiče na proces izdavajanja tokena u tekstu. Regularni izraz je najčešći mehanizam kojim se tokenizatoru zadaje separatorski skup.

Primer 3.14. Separatorski skup koji čine samo beline može se opisati regularnim izrazom `[␣\t\n]+`, pri čemu je simbolom ␣ označen razmak.

Ovde treba napomenuti da postoji još metasekvenci koje se tretiraju kao beline (Tabela 3.2). Programski jezici koji podržavaju standard POSIX uglavnom koriste

posebnu metasekvencu `\s` kao regularan izraz kojim opisuju proizvoljnu belinu, tj. `\s` je ekvivalentno sa `[␣\t\n\r\f\v]`.

Tabela 3.2: Metasekvence kojima se opisuje razne vrste belina u tekstu

Metasekvenca	**Naziv**	**Naziv u originalu**
`\n`	novi red	newline character, line feed
`\t`	tabulator	(horizontal) tab
`\r`	„vraćanje kolica”	carriage return
`\f`	nova strana	form feed
`\v`	vertikalni tabulator	vertical tab

Ukoliko je obrada teksta linijski-orijentisana, najpre se tekst podeli na linije, a onda se svaka linija podeli na **polja** (eng. **field**), pri čemu se sekvence razmaka i tabulatora koriste kao separatori polja[6].

Različiti operativni sistemi različito implementiraju terminator linije: Windows koristi dva karaktera kojima u karakterskom skupu ASCII odgovaraju pozicije 13 (*carriage return*) i 10 (*line feed*), dok Linux koristi samo potonji karakter (*line feed*). Karakterski skup Unicode, pored pomenutih ASCII-karaktera i sekvenci, definiše tri posebna karaktera za terminator linije, čije su heksadekadne kodne pozicije U+0085 (NEXT LINE), U+2028 (LINE SEPARATOR) i U+2029 (PARAGRAPH SEPARATOR). Pošto je većina aplikacija, uključujući i programe za uređivanje teksta, linijski-orijentisana, korisnik uglavnom ne mora da vodi računa o pravom terminatoru linije. U slučaju da korisnik sam mora da podeli tekst na linije, u većini slučajeva dovoljno je da linije razdvoji regularnim izrazom `\n`.

Međutim, kada se tekst obrađuje regularnim izrazima u **višelinijskom režimu** (eng. **multiline**), tj. kada treba prepoznati deo teksta koji se proteže kroz dve ili više linija, od implementacije konkretne aplikacije zavisi kako će se opisati kraj linije. Perl, na primer, koristi sidra (^ i $) i ne vodi računa o konkretnom terminatoru linije, dok phreplace[7] (dodatak za PSPad) i TextCrawler[8] zahtevaju preciznu sekvencu karaktera koji čine novi red (`\r\n` za Windows-tekstuelne datoteke).

Primer 3.15. U slučaju tokenizacije teksta na prirodnom jeziku separatorski skup se najčešće proširuje svim nealfanumeričkim karakterima (tj. karakterima koji nisu

[6] U slučaju programskih jezika awk i Perl ovo je podrazumevana obrada ulaznog teksta

[7] `http://www.phdesign.com.au`

[8] `http://www.digitalvolcano.co.uk/content/textcrawler`

ni slova ni cifre), ali se svi separatori, sa izuzetkom belina, tretiraju kao pojedinačni tokeni. Stoga se tokenizacija vrši u dve faze:

- U prvoj fazi se tekst modifikuje tako da se oko svakog separatora koji nije belina ubacuju beline koje ga razdvajaju od okolnog teksta (konkretno, po jedan razmak ispred i iza separatora). Takođe, svaki povezani niz cifara se tretira kao celina (broj) i na isti način se odvaja razmacima od okolnog teksta.
- U drugoj fazi se iz modifikovanog teksta izdvajaju tokeni, pri čemu separatorski skup čine samo beline.

Ovakva tokenizacija može se ilustrovati sledećim naredbama u programskom jeziku C#:

```
//dodavanje razmaka ispred i iza nealfanumeričkih karaktera
niska = Regex.Replace(niska, "([^A-Za-z0-9␣\t\n])", "␣$1␣");
//dodavanje razmaka ispred i iza niza cifara
niska = Regex.Replace(niska, "([0-9]+)", "␣$1␣");
//izdvajanje tokena, separatori su beline
String [] tokeni = Regex.Split(niska, "[␣\t]+");
```

Regularni izraz za opis nealfanumeričkih karaktera (`[^A-Za-z0-9␣\t\n]`) je korektan za tekstove na engleskom jeziku, ali ne i za tekstove na srpskom ili bilo kom drugom jeziku čija sva slova ne mogu da se predstave ne-ASCII-karakterima. Da bi se taj problem prevazišao, moguća su dva rešenja. Prvo rešenje podrazumeva da se tekst kodira tako da koristi samo ASCII karaktere, pri čemu ne-ASCII-slova moraju biti zamenjena sekvencom od jednog ili više ASCII-karaktera (v. kôd YUSCII, odeljak 2.3, ili kôd aurora, odeljak 6.2).

Drugi izlaz je da se za tekst koristi neki od kodnih rasporeda kojim se kodira karakterski skup Unicode (v. kodove UTF-16 i UTF-8, odeljak 2.3), a da se regularni izraz modifikuje tako da negativna klasa karaktera isključuje ne samo slova engleske abecede (`[^A-Za-z]`), već sve slovne karaktere u skupu Unicode. Upravo takvu mogućnost daju .NET-regularni izrazi koje koristi programski jezik C# zato što podržavaju Unicode, a u okviru klase karaktera i negativne klase karaktera mogu

koristiti opštu kategoriju `[^\p{L}]` koja označava skup svih Unicode-slova. Stoga sledeća modifikacija regularnog izraza u prvoj naredbi nudi drugo rešenje problema:

```
niska = Regex.Replace(niska, "([^\p{L}0-9␣\t\n])", "␣$1␣");
```

Segmentacija na rečenice U odeljku 2.3, pododeljak Identifikacija kraja rečenice, su opisana dva osnovna pristupa problemu segmentacije teksta na rečenice:

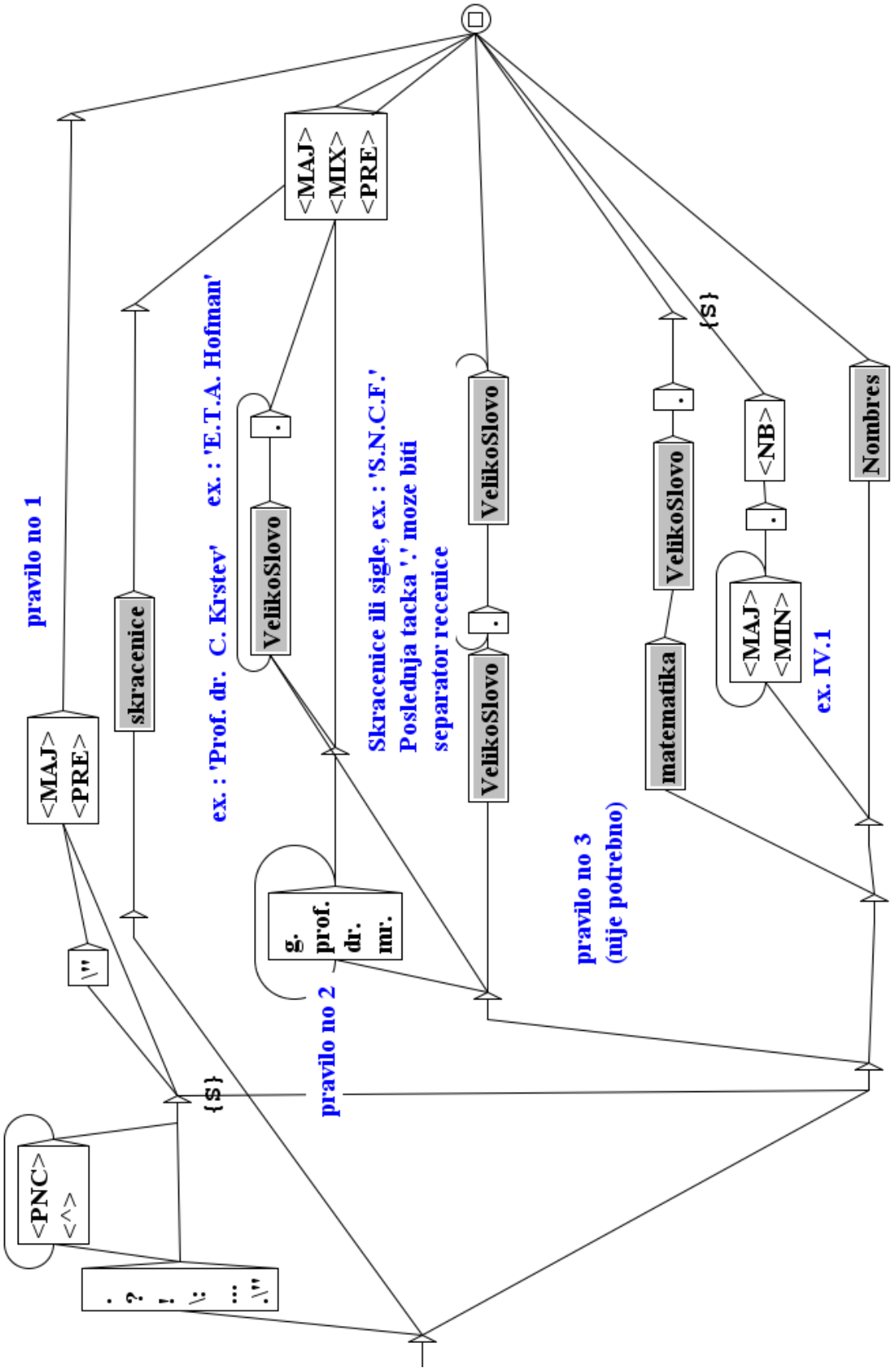
- segmentacija teksta na rečenice zasnovana na pravilima i
- segmentacija teksta na rečenice zasnovana na statističkim metodama.

Prvi pristup, zasnovan na pravilima, u slučaju tekstova na srpskom jeziku ilustrovaćemo na primeru sistema Unitex ([Paumier, 2011]) i njegovog modula za srpski jezik ([Vitas et al., 2003], [Krstev & Vitas, 2005], [Krstev, 2008]).

Unitex koristi regularne izraze za opis razovrsnih lingvističkih fenomena (morfoloških, sintaksičkih, semantičkih). Slovni karakteri azbuke se definišu u posebnoj datoteci koja se naziva **datoteka azbuke** (eng. **alphabet file**).

Međutim, što su lingvistički fenomeni složeniji, to i regularni izrazi koji ih opisuju postaju kompleksniji i teško čitljivi, a često se u opisu nekog fenomena na više mesta pojavljuje isti regularni izraz. Stoga Unitex, uz regularne izraze, primenjuje dodatni formalizam — kolekcije grafova i podgrafova. Svaki graf se čuva u posebnoj datoteci i na njega se može referisati po imenu, čime se omogućava da se jedan graf koristi više puta kao podgraf nekog složenijeg grafa. Slika 3.2 ilustruje jedan graf sistema Unitex čiji su podgrafovi predstavljeni osenčenim čvorovima (`skracenice`, `VelikoSlovo`, itd.) koji se nalaze u istoimenim datotekama tipa `.grf` (`skracenice.grf`, `VelikoSlovo.grf`, itd.).

U sistemu Unitex postoje razne vrste grafova: za prethodnu obradu teksta, automatsko generisanje rečnika oblika promenljivih vrsta reči na osnovu rečnika njihovih lema (flektivni grafovi), pretragu teksta (sintaksički grafovi, v. odeljak 4.3), razrešavanje višeznačnosti, itd. Ovde će biti reči o grafovima za prethodnu obradu teksta, preciznije, o grafu za segmentaciju teksta na rečenice koji koristi modul za srpski.



Slika 3.2: Radna verzija grafa sistema Unitex za segmentaciju teksta na rečenice.

Graf sistema Unitex je usmereni graf koji ima jedan početni i jedan završni čvor. Svi čvorovi sadrže neki regularni izraz, pri čemu prazni čvorovi (uključujući i početni i završni čvor) predstavljaju praznu nisku. Čvorovi u grafu su povezani usmerenim lukovima kojima se grafički predstavlja operacija dopisivanja. U slučaju da luk ulazi u isti čvor iz koga izlazi, u pitanju je grafički prikaz pozitivnog zatvorenja, dok mogućnost da iz istog čvora izlazi više lukova omogućava predstavljanje i operacije unije.

U zavisnosti od režima u kome se vrši prethodna obrada teksta, osnovne jedinice regularnih izraza su ili karakteri ili tokeni. Svaki put od početnog do završnog čvora predstavlja jedan složeni regularni izraz i unija svih tih izraza odgovara skupu niski koje graf prepoznaje.

Unitex-grafovi imaju mogućnost da, kada prepoznaju neki deo teksta, emituju izlazne niske i to u dva režima: režimu umetanja i režimu zamene. U prvom slučaju se tekst modifikuje tako što se posle prepoznate niske umeće odgovarajuća izlazna niska. U režimu zamene se prepoznata niska zamenjuje odgovarajućom izlaznom niskom.

Tokom prethodne obrade teksta, modul za srpski jezik sistema Unitex segmentira tekst na rečenice koristeći graf u režimu umetanja, pri čemu su osnovne jedinice tokeni, a ne karakteri (Slika 3.2). Segmentacija se obavlja tako što se u tekst umeće separator rečenica `{S}`. U grafu se koriste specijalni regularni izrazi[9] ([Paumier, 2011: 37–38]):

- `<PNC>` : prepoznaje interpunkcijske simbole `; , ! ? : ¡ ¿` i neke interpunkcijske simbole azijskih jezika;
- `<MAJ>` : prepoznaje proizvoljni niz velikih slova;
- `<MIN>` : prepoznaje proizvoljni niz malih slova;
- `<MOT>` : prepoznaje proizvoljni niz slova;
- `<PRE>` : prepoznaje proizvoljni niz slova koji počinje velikim slovom;

[9]Ako graf za prethodnu obradu teksta radi u režimu karaktera, pored navedenih regularnih izraza može koristiti i regularni izraz `<L>` koji prepoznaje proizvoljno slovo definisano u datoteci azbuke.

- `<NB>` : prepoznaje proizvoljni niz cifara (na primer, 1234, ali ne i 1 234);
- `<E>` : prepoznaje praznu nisku;
- `<^>` : prepoznaje znak za novi red;
- `#` : zabranjuje razmak na zadatoj poziciji.

Prvo pravilo implementira algoritam „tačka-razmak-veliko slovo" umetanjem separatora rečenica `{S}` na određenoj poziciji, pri čemu:

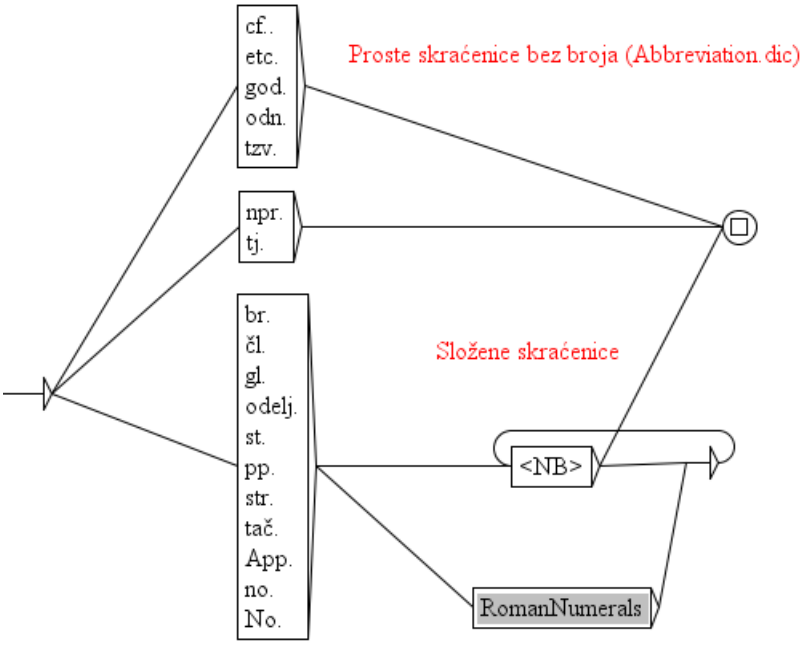
- levi kontekst pozicije predstavlja neki od simbola `. ? ! : " ...` iza kog je ili opcioni znak interpunkcije (`<PNC>`) ili opcioni znak za novi red (`<^>`);
- desni kontekst pozicije je ili proizvoljni niz velikih slova (`<MAJ>`) ili proizvoljni niz slova koji počinje velikim slovom (`<PRE>`), ispred kojih je opcioni navodnik;
- pošto put u grafu ne sadrži regularni izraz #, između uzastopnih tokena može biti proizvoljan broj razmaka.

Drugo pravilo sprečava segmentiranje rečenica na pozicijama koje prepoznaje prvo pravilo ako levi kontekst predstavlja neka od skraćenica koje se završavaju tačkom, uključujući akronime i sigle. Za prepoznavanje skraćenica se koristi poseban podgraf (Slika 3.3) koji u sebi sadrži još jedan podgraf (RomanNumerals) koji prepoznaje rimske brojeve.

Treće pravilo pokriva razne slučajeve koji se odnose na brojeve (celi brojevi sa separatorom hiljada, brojevi sa decimalama, redni brojevi odeljaka u referencama, itd.).

Treba napomenuti da je ovde radi ilustracije predstavljena samo pojednostavljena verzija opštijeg grafa, a za pojedine slučajeve tek treba razviti posebne podgrafove (na primer, podgraf za XML-dokumente koji bi sprečio umetanje separatora rečenica unutar XML-etiketa).

Trenutno u okviru grafova za prethodnu obradu teksta nije moguće korišćenje informacija iz rečnika, kao ni drugih tipova regularnih izraza koji su dostupni u morfološkom režimu, što bi svakako unapredilo pravila za prepoznavanje skraćenica i vlastitih imena. S obzirom da Unitex primenjuje rečnike isključivo na segmentiran



Slika 3.3: Graf sistema Unitex za prepoznavanje skraćenica

tekst, nije verovatno da će se takva mogućnost pojaviti u budućim implementacijama.

Morfološka analiza INTEX ([Silberztein, 1999]), Unitex ([Paumier, 2011]) i NooJ [Silberztein, 2003] su primeri programskih sistema koji regularnim izrazima opisuju pravila za generisanje flektivne paradigme polazeći od leme. S obzirom da je svaka flektivna paradigma konačan skup niski, to je i regularan skup, pa je opisiva regularnim izrazom. Ukoliko se azbuka paradigme proširi i oznakama operatora za pozicioniranje u okviru niske, brisanje i umetanje karaktera, moguće je opisati pravilo kojim se na osnovu leme generišu svi njeni flektivni oblici.

Tako u sistemu INTEX sledeći regularni izraz[10] (deo pravila koje opisuje flektivnu klasu `A1`)

```
<E>/:akms1g + og/:adms2g + om/:adms3g + og/:adms4v
```

[10]U ovom slučaju je alternacija označena sa + umesto sa |, a `<E>` označava praznu nisku. Svaki deo regularnog izraza između dva uzastopna znaka alternacije opisuje generisanje jednog flektivnog oblika i podeljen je kosom crtom (/) na dva dela. Deo izraza levo od kose crte ukazuje kako treba izmeniti lemu da bi se dobio flektivni oblik i u navedenom primeru svuda predstavlja nisku koja se dopisuje na kraj leme. Deo izraza desno od kose crte predstavlja vrednosti flektivnih kategorija koje se dopisuju na konačni zapis flektivnog oblika zajedno sa zapisom leme.

polazeći od leme[11]

```
studentov.A1+Pos+Der+Hum
```

generiše deo flektivne paradigme prideva `studentov`

```
studentov,studentov.A+Pos+Der+Hum:akms1g
studentovog,studentov.A+Pos+Der+Hum:adms2g
studentovom,studentov.A+Pos+Der+Hum:adms3g
studentovog,studentov.A+Pos+Der+Hum:adms4v
```

Obrada „ne(pre)poznatih" reči Prilikom obrade teksta pomoću morfološkog elektronskog rečnika u sistemu INTEX, Unitex i NooJ, kao jedan od rezultata se dobija i lista „ne(pre)poznatih" reči, tj. alfabetskih tokena kojih nema u primenjenim morfološkim rečnicima. Takvi tokeni mogu biti bilo pogrešno otkucane reči u tekstu, bilo ispravne reči koje iz nekog razloga nisu zabeležene u rečniku. U nekim slučajevima se bez konteksta u kome se token pojavio i ne može odrediti šta je od navedenog u pitanju. U svakom slučaju, da bi se eventualna greška ispravila, neophodno je pronaći pojavljivanje „ne(pre)poznate" reči u tekstu. Uobičajeni dijalozi za pretragu teksta nisu od velike pomoći ukoliko je u pitanju kratka sekvenca slova koja se često pojavljuje kao podniska u tekstu. Međutim, pošto se zna da je „ne(pre)poznata" reč kompletan token, na osnovu skupa separatora tokena se može definisati regularni izraz koji neposredno pronalazi prvo pojavljivanje „ne(pre)poznate" reči. Na primer, ako je separator tokena bilo koji karakter koji nije slovo engleske azbuke, a traga se za „ne(pre)poznatim" rečima `st` i `al`, sledeći regularni izraz u programu PSPad se može upotrebiti da pronađe sva njihova pojavljivanja:

```
(^|[^A-Za-z])(st|al)([^A-Za-z]|$)
```

Navedeni regularni izraz pronalazi u tekstu sva pojavljivanja niski `st` i `al` ispred kojih je ili početak reda ili neki neslovni karakter, a iza kojih sledi ili neslovni karakter ili kraj reda. Drugim rečima traže se zadate niske između dva uzastopna separatora tokena. Pošto je pretraga linijski orijentisana (tj. pretražuje se linija po linija, a znak za novi red se ignoriše), regularni izraz

[11] Lema je navedena u formatu DELA koji je detaljno objašnjen u odeljku 2, str. 145.

`[^A-Za-z](st|al)[^A-Za-z]`

ne bi pronašao tražene niske na početku ili kraju reda, jer znak za novi red nije obuhvaćen regularnim izrazom

`[^A-Za-z]`.

Plitko parsiranje Naredni primeri će ilustrovati prepoznavanje nekih elemenata imenovanih entiteta kao što su brojevi (kao delovi novčanih iznosa, merenja), postepeno će se izložiti konstruisanje složenog regularnog izraza za prepoznavanje datuma.

Primer 3.16. Regularni izraz `[1-9][0-9]*` označava proizvoljan prirodan broj.

Primer 3.17. Regularni izraz `[+-](0|[1-9][0-9]*)` označava proizvoljan ceo broj.

Primer 3.18. Regularni izraz `[+-]?([1-9][0-9]*|0),[0-9]*` označava zapis proizvoljnog realnog broja u nepokretnom zarezu.

Primer 3.19. U Primeru 3.13 je pokazano da se redni broj dana u mesecu (1–31) može predstaviti regularnim izrazom `[1-9]|[12][0-9]|3[01]`. Ovo je naravno korektno za januar, mart, maj, jul, avgust, oktobar i decembar, ali za ostale mesece treba napisati posebne regularne izraze. Za mesece april, jun, septembar i novembar, tj. za opseg 1–30, odgovarajući regularni izraz je `[1-9]|[12][0-9]|30`, dok februaru mogu odgovarati dva regularna izraza — `[1-9]|1[0-9]|2[0-8]` (opseg 1–28) i `[1-9]|[12][0-9]` (opseg 1–29) — sve u zavisnosti je li godina prestupna ili ne.

Primer 3.20. Pretpostavimo da regularnim izrazom treba opisati prestupnu godinu iz opsega 1000–2099. Godine koje se ne završavaju sa dve nule su prestupne ako su deljive sa četiri (na primer, 2004, 2008, 2012. itd.). Godine koje se završavaju sa dve nule su prestupne jedino ako su deljive sa 400, odnosno, ako je broj koji se dobija brisanjem poslednje dve nule u zapisu deljiv sa 4. Prema tome, problem se svodi na opis prirodnih brojeva koji su deljivi sa četiri. Može se pokazati da je prirodan broj deljiv sa četiri, ako je dvocifren broj sastavljen od dve njegove poslednje cifre deljiv sa 4. Time je problem dodatno sužen na opis dvocifrenih brojeva koji su deljivi sa 4. Oni se mogu predstaviti na sledeći način:

Tabela 3.3: Dvocifreni brojevi deljivi sa četiri

00	04	08	12	16
20	24	28	32	36
40	44	48	52	56
60	64	68	72	76
80	84	88	92	96

Iz tabele se neposredno zaključuje da se brojevi deljivi deljivi sa 4 mogu predstaviti kao unija dva skupa (razdvojena u tabeli vertikalnom linijom), pri čemu jedan skup čine brojevi koji počinju ciframa 0, 2, 4, 6 ili 8, a završavaju se ciframa 0, 4 i 8, dok drugi skup čine brojevi koji počinju ciframa 1, 3, 5, 7 ili 9, a završavaju se ciframa 2 i 6. Odatle se lako konstruiše regularni izraz za opis dvocifrenih brojeva deljivih sa četiri: `[02468][048]|[13579][26]`.

Ukoliko želimo da eliminišemo nisku 00, odgovarajući regularni izraz bi imao oblik `0[48]|[2468][048]|[13579][26]`.

U slučaju godina koje se završavaju sa dve nule (1000, 1100,..., 2000, 2100), brisanjem tih nula dobijaju se brojevi iz opsega 10–21, pri čemu prestupnim godinama odgovaraju brojevi deljivi sa četiri (12, 16 i 20). Stoga bi definitivni regularni izraz za opis prestupnih godina iz opsega 1000–2100 imao oblik[12]:

```
(12|16|20)([02468][048]|[13579][26])
|(1[01345789]|21)(0[48]|[2468][048]|[13579][26]).
```

Anotacija Regularni izrazi se mogu iskoristiti i za automatsku i poluautomatsku anotaciju logičke strukture teksta. Svi primeri regularnih izraza, koji će biti navedeni u ovom pododeljku, testirani su u programu PSPad. U pojedinim programskim jezicima (awk, Perl) i programima za uređivanje teksta (MS Word, PSPad) regularni izrazi koriste zagrade ne samo za grupisanje i izbegavanje prioriteta, već i za memorisanje delova prepoznate niske na koje se u daljoj obradi može referisati pomoću specijalnih promenljivih (eng. backreferencing). Broj specijalnih promenljivih za memorisanje je obično ograničen na deset, koliko ih ima i PSPad (u oznaci $0, $1,

[12]Izraz je prelomljen u dve linije pošto ne može da stane u jednu liniju, a u stvari treba da bude zapisan u jednoj liniji bez razmaka između prikazanih delova. Prva linija prikazuje prestupne godine koje počinju sa 12, 16 i 20, a završavaju se dvocifrenim brojem deljivim sa 4, uključujući i 00. Druga linija prikazuje godine iz opsega 1000–2100 koje ne počinju sa 12, 16 i 20, a završavaju se dvocifrenim brojem deljivim sa 4, isključujući nisku 00.

. . . $9). Promenljiva $0 označava celu nisku prepoznatu regularnim izrazom, dok $1, . . . $9 redom odgovaraju podizrazima, tj. parovima zagrada, gledano sleva nadesno. Tako za regularni izraz `[A-Z]([a-z](:)[a-z])` promenljiva $1 odgovara podizrazu `([a-z](:)[a-z])`, dok promenljiva $2 odgovara podizrazu `(:)`.

Ukoliko je tekst kodiran tako da svaka linija predstavlja jedan pasus, tj. tako da je znak za novi red unošen isključivo na kraju pasusa, pasus se može anotirati TEI-elementom `p` (tj. njegova sadržina ograničiti između etiketa `<p>` i `</p>`), ukoliko se u dijalogu za pretragu i zamenu navede regularni izraz

```
^(.*)$
```

a kao tekst zamene

```
<p>$1</p>
```

Ukoliko su naslovi na neki način istaknuti u tekstu (sastoje se isključivo od velikih slova i znakova interpunkcije ili se u njihovom nazivu pojavljuje karakteristična niska, na primer `GLAVA`), to se može iskoristiti ne samo za anotaciju naslova TEI-elementom `head`, već i za anotaciju odeljaka na čijem početku se pojavljuju ti naslovi primenom TEI-elementa `div`. Na primer, ako se naslov sastoji od tokena `GLAVA` i rimskog broja glave, pri čemu je ukupan broj glava manji od pedeset, naslovi i odeljci se mogu anotirati tako što se u dijalogu za pretragu i zamenu navede regularni izraz

```
^(GLAVA)␣([IVXL]+)$
```

a kao tekst zamene

```
</div>\n<div type='chapter' n='$2'>\n<head>$1 $2</head>
```

S obzirom da se pri kucanju i formatiranju teksta mogu pojaviti dodatne beline na početku, kraju ili između komponenti naslova, precizniji regularni izraz bi izgledao ovako:

```
^[␣\t]*(GLAVA)␣[␣\t]*([IVXL]+)[␣\t]*$
```

a kao odgovarajući tekst zamene

```
</div>\n<div type='chapter' n='$2'>\n<head>$1 $2</head>
```

Na taj način bi naslov

```
GLAVA XIV
```

bio anotiran na sledeći način:

```
  </div>
```

```
<div type='chapter' n='XIV'>
<head>GLAVA XIV</head>
```

Naravno, ovakav postupak bi zahtevao i ručnu korekciju prve i poslednje glave, tj. premeštanje prve zatvorene etikete `</div>` na kraj poslednje glave, kako bi TEI-dokument bio dobro formiran.

3.2 Konkordance

U opštem slučaju **konkordance** (eng. **concordances**) predstavljaju „alfabetski uređenu listu (ili indeks) reči (ili ideja; ponekad ograničenih na ključne ili značajne reči) u tekstu ili grupi tekstova (ili u delima određenog autora), zajedno sa pozicijama u tekstu (poglavlje, stih, čin, scena, redni broj linije, itd.) svakog pojavljivanja reči, kao i citatima ili izvodima iz konteksta" ([Brown, 2005], odrednica *Concordance*). Smatra se da naziv *konkordanca* potiče od latinskog glagola *concordo, concordare, concordavi, concordatus* sa značenjem *slagati (se).*

U korpusnoj lingvistici konkordance se koriste kao metod za vizuelizaciju podataka iz korpusa koji odgovaraju zadatom upitu korisnika ([Wynne, 2008]). Korisnik najčešće zadaje upit u formi regularnog izraza, tako da se u opštem slučaju traga za pojavljivanjem skupa niski, pri čemu jedna niska može biti sastavljena od jednog ili više tokena. Regularnim izrazom korisnik može da traži određenu tekstuelnu (korpusnu) reč, paradigmu leme, višečlanu leksemu, frazu ili pak korpusne reči sa određenim afiksima. Uz sve to, ako je korpus anotiran, u konkordancama se mogu uz svaki token prikazati i odgovarajuće pridružene informacije, na primer vrsta reči ili lema. Stoga pojedini autori ([McEnery & Hardie, 2012: 35]) dobro primećuju da pojmovi **ključna reč** (eng. **keyword/key word**) i **tražena reč** (eng. **search-word**) nisu najsrećnije odabrani da označe element regularnog skupa opisanog korisnikovim upitom, pogotovo što kod različitih autora postoje dileme, na primer u slučaju kad se razmatra traženje fraze, da li se tim pojmovima označava cela fraza (kao jedna ključna/tražena reč) ili pak tokeni koji sačinjavaju tu frazu (u tom slučaju se o frazi govori kao o „nizu ključnih reči"). Činjenica je da su vremenom pojmovi *ključna reč* i *tražena reč* postali standardni termini korpusne lingvistike, pa će i ovde biti

korišćeni u odgovarajućem značenju, pri čemu će se fraza tretirati kao niz ključnih reči.

Programski alat kojim se automatski generišu konkordance naziva se **konkordancer** (eng. **concordancer**). U korpusnoj lingvistici naziv *konkordancer* se ponekad koristi u širem značenju, tj. da označi sistem za kreiranje i analizu korpusa, pri čemu je modul za generisanje konkordanci, tj. konkordancer u užem smislu te reči, samo jedna od osnovnih komponenti tog sistema.

Formati konkordanci

Kao rezultat pretrage korpusa, konkordance se mogu formatirati na različite načine.

Najčešće korišćeni format konkordanci je format **ključne reči u kontekstu** (eng. **Key Word In Context**, skr. **KWIC**). Format KWIC objasnićemo na primeru konkordanci (Slike 3.4 i 3.5) za upit zadat regularnim izrazom

```
naprotiv ([A-Za-z]+)?  \.
```

kojim se aproksimiraju sva pojavljivanja lekseme `naprotiv` na kraju izjavne rečenice (`\.`), pri čemu se između te lekseme i tačke može naći najviše jedna tekstuelna reč (`([A-Za-z]+)?`)[13].

U formatu KWIC svaka konkordanca je predstavljena jednom linijom u kojoj su ključne reči istaknute, bilo naročitim formatiranjem — kurziv, podvlačenje, podebljana slova (Slika 3.4) — bilo upotrebom posebnih karaktera kao separatora, na primer uglastim zagradama < i > (Slika 3.5). Na taj način su ključne reči jasno odvojene od odgovarajućeg levog i desnog konteksta, a linije su poravnate po granicama između ključnih reči i odgovarajućeg konteksta, tako da se konkordance mogu posmatrati kao tekst u tri kolone i ključnim rečima u srednjoj koloni.

Dužina levog i desnog konteksta konkordanci u KWIC formatu se obično bira tako da dužina cele linije ne prelazi granice medijuma na kojima se nalaze (papir, ekran). Konkordanceri uglavnom imaju opciju za istovremeno ili odvojeno

[13]Konkordance su proizvedene pomoću alata *cqp* ([Evert & The OCWB Development Team, 2010b]), a na osnovu verzije SrpKor2013 Korpusa savremenog srpskog jezika (`http://www.korpus.matf.bg.ac.rs`).

lingvistiku , već ju je ***naprotiv*** **bodrila .** Danas , međutim , nakon
a se nije stišala , već ***naprotiv*** **razbesnela .** Zvonjenje njegovog mobil
posvađali ? ” “ Ne , ***naprotiv*** **.** Rekao sam Fani da mi nis
te krčme ? - Ništa , ***naprotiv*** **.** - Znate li je ? - Ne , i
stva . . . Izvolite . . . ***naprotiv*** **.** . . Gospođa Rigo ulazi
Ja neću da jurišam , ***naprotiv*** **.** - Onda ćete vi dati vas
. Tiče me se mnogo , ***naprotiv*** **.** Ja sam veran podanik Nje
neće naneti neko zlo , ***naprotiv*** **.** . . Na trkama će , gosp
isključuje moralnost , ***naprotiv*** **.** Revolveri su bez sumnje
ih vas tešio , nego baš ***naprotiv*** **.** - Šta hoćete da kažet

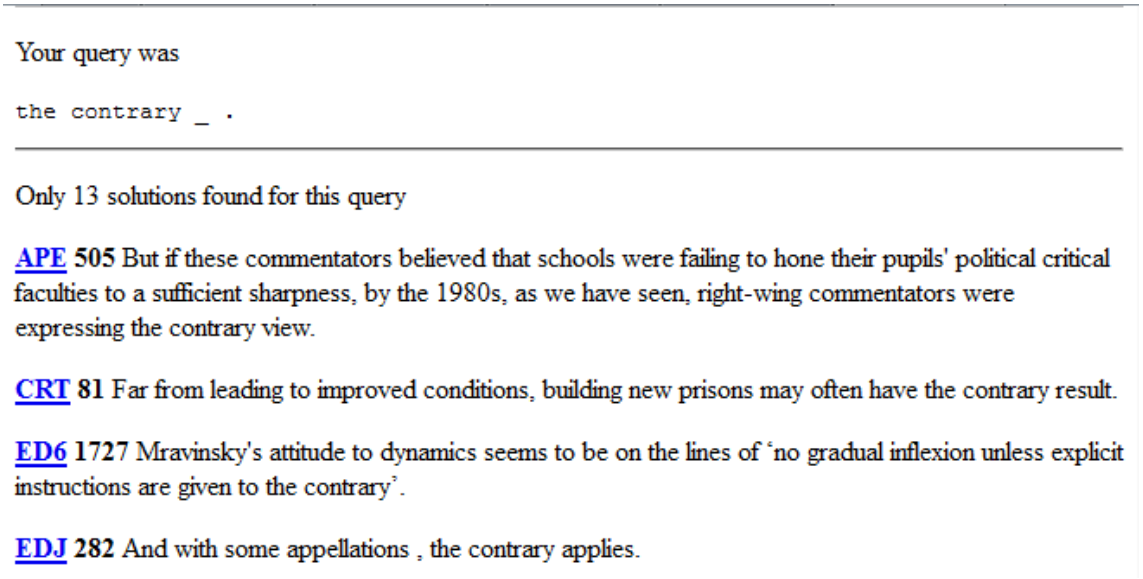
Slika 3.4: Konkordance u formatu KWIC (LaTeX)

```
lingvistiku , već ju je <naprotiv bodrila .> Danas , međutim ,
a se nije stišala , već <naprotiv razbesnela .> Zvonjenje
 posvađali ?  ”  “ Ne , <naprotiv .> Rekao sam Fani da mi nis
iv te krčme ? - Ništa , <naprotiv .> - Znate li je ? - Ne , i
va . . . Izvolite . . . <naprotiv .> . . Gospođa Rigo ulazi
 . Ja neću da jurišam , <naprotiv .> - Onda ćete vi dati vas
se . Tiče me se mnogo , <naprotiv .> Ja sam veran podanik Nje
 neće naneti neko zlo , <naprotiv .> . . Na trkama će , gosp
 isključuje moralnost , <naprotiv .> Revolveri su bez sumnje
ih vas tešio , nego baš <naprotiv .> - Šta hoćete da kažet
```

Slika 3.5: Konkordance u formatu KWIC (čist tekst)

podešavanje dužine levog i desnog konteksta, pri čemu se dužina može zadati bilo brojem karaktera, bilo brojem tokena, odnosno korpusnih reči. Mogućnost promene dužine konteksta rešava probleme koji se javljaju prilikom prikaza konkordanci, poput nepreglednosti (ukoliko je reč o preširokom kontekstu) ili nedovoljne količine informacija za razumevanje upotrebe ključnih reči (ukoliko je kontekst prekratak).

Osim uobičajenog formata KWIC, konkordance mogu biti kreirane u formatima koji ključne reči prikazuju kao deo rečenice ili pasusa, ali to zahteva da granice rečenica, odnosno pasusa, budu anotirane u korpusu kao elementi logičke strukture teksta. Jednostavna pretraga Britanskog nacionalnog korpusa daje konkordance kao niz rečenica koje sadrže ključne reči (Slika 3.6).



Your query was

```
the contrary _ .
```

Only 13 solutions found for this query

APE 505 But if these commentators believed that schools were failing to hone their pupils' political critical faculties to a sufficient sharpness, by the 1980s, as we have seen, right-wing commentators were expressing the contrary view.

CRT 81 Far from leading to improved conditions, building new prisons may often have the contrary result.

ED6 1727 Mravinsky's attitude to dynamics seems to be on the lines of 'no gradual inflexion unless explicit instructions are given to the contrary'.

EDJ 282 And with some appellations , the contrary applies.

Slika 3.6: Konkordance u formatu KWIC preuzete iz Britanskog nacionalnog korpusa na osnovu upita `contrary _ .` (tačka nema specijalno značenje, dok `_` označava proizvoljan token).

Istorijat konkordanci

Preelektronske konkordance

Kao što je već navedeno u odeljku 1.1 (pododeljak o preelektronskim korpusima, str. 2) prve konkordance sa kompletnim referencama su izrađivane u XIII veku,

dok se za samu ideju konkordanci smatra da potiče iz X veka[14]. Prve konkordance su kreirane ručno, kao pomoćno sredstvo prilikom istraživanja, najpre u teologiji (biblijske konkordance), a potom i u književnosti (konkordance radova Šekspira, Tenisona, Šelija, Blejka) i leksikografiji (konkordance književnih dela za potrebe izrade Oksfordskog rečnika engleskog jezika u drugoj polovini XIX veka). Ručna izrada konkordanci je iziskivala ogroman napor, ljudske resurse i vreme[15].

Računarski generisane konkordance

Neposredno po pojavi prvih elektronskih računara javlja se ideja da se konkordance automatski generišu i pretražuju pomoću računara. Sveštenik Roberto Busa je 1949. godine uspeo da ubedi Tomasa Votsona, osnivača kompanije IBM, da IBM bude pokrovitelj izrade indeksa sabranih dela Svetog Tome Akvinskog — Index Thomisticum, projekta koji je trajao tridesetak godina i koji je već 1951. godine generisao prve automatske konkordance. Index Thomisticum sadrži pedeset šest tomova čije su konkordance danas dostupne na vebu ([Bernot & Alarcón, 2005]).

Pojava elektronskih korpusa je takođe uticala na razvoj računarskih programa za generisanje konkordanci, tj. konkordancera.

Pregled konkordancera kroz generacije

Prema [McEnery & Hardie, 2012]37–48, od prvih elektronskih korpusa do danas se pojavilo četiri generacije konkordancera.

Konkordanceri I generacije

Konkordanceri I generacije su obeležili razdoblje elektronskih korpusa I generacije, tj. dvadesetak godina od pojave prvog elektronskog korpusa početkom šezdesetih godina XX veka. U tom periodu još uvek nije bilo personalnih računara niti globalne

[14]Ideja se pripisuje Masoretima, jevrejskim srednjevekovnima prepisivačima, poznatim po tome što su svaki tekst prepisivali bez ikakave izmene, a na marginama bi ostavljali svoje ispravke. Masoreti su na osnovu Hebrejske Biblije izradili listu reči, pri čemu je za svaku reč navedena i fraza u okviru koje se reč pojavila u Bibliji ([Krstev & Gucul, 2007]).

[15]Navodno je oko 500 dominikanskih fratara učestvovalo u izradi konkordanci na osnovu latinske Biblije iz V veka, dok je Aleksander Kruden sam izradio biblijske konkordance na engleskom jeziku u periodu od 1736. do 1738. godine.

računarske mreže. Svaka istraživačka grupa je razvijala svoj konkordancer koji se izvršavao na centrali (eng. mainframe computer), a podacima se pristupalo isključivo u ustanovi u kojoj se nalazila centrala, tj. lokalno. Prvi konkordanceri su neretko bili neprenosivi sa jedne centrale na drugu. Zvanični standardi o kodiranju karaktera (izvan skupa slova engleskog alfabeta, cifara i nekih znakova interpunkcije) su bili u začetku, standardi (strukturne, morfološke, itd.) anotacije nisu ni postojali već su grupe istraživača improvizovale, nezavisno jedne od drugih. Rad konkordancera se svodio na puki prikaz konkordanci u formatu KWIC, a zadatke poput generisanja liste reči korpusa, sortiranje konkordanci, generisanje listi učestanosti, itd., realizovali su odvojeni programi, najčešće napisani *ad hoc*.

Konkordanceri II generacije

Početkom osamdesetih godina XX veka pojava personalnih računara i spoljašnjih memorija (poput disketa i kompakt diskova) omogućila je da se isti konkordanceri prošire na više računara i koriste za pretragu različitih korpusa, tj. prestala je potreba da svako programira svoj sopstveni konkordancer. Takođe, korpusni lingvisti su postali manje zavisni od programera, svaki lingvista je mogao da prebaci korpus i konkordancer na svoj personalni računar i da pretražuje svoj primerak korpusa, što je jedan od faktora koji su uticali na nagli razvitak korpusne lingvistike osamdesetih godina XX veka ([McEnery & Hardie, 2012: 39]).

Međutim, konkordanceri na prvim personalnim računarima su po svojim mogućnostima stagnirali ili čak zaostajali za konkordancerima I generacije ([McEnery & Hardie, 2012: 39–40]):

- konkordanceri I generacije na centralama su bili u stanju da pretražuju korpuse reda veličine milion reči, dok konkordanceri novije generacije, zbog ograničenosti performansi personalnih računara, nisu mogli da pretraže korpuse čija je veličina prelazila nekoliko desetina hiljada reči;
- tek krajem osamdesetih godina XX veka se pojavljuju prvi standardi za osmobitno kodiranje elektronskog teksta (ISO i Microsoft, v. odeljak 2.2), tako da većina konkordancera II generacije podržava isključivo standard ASCII;

- za kraj osamdesetih godina XX veka se vezuje i prva verzija standarda anotacije TEI (v. odeljak 2.4, str. 129 ali kao i u slučaju osmobitnih kodnih rasporeda, većina konkordancera ne podržava TEI. Štaviše, za razliku od konkordancera I generacije, novi konkordanceri ne razlikuju anotaciju od samog teksta i nisu u stanju da pretražuju korpuse sa ugrađenom anotacijom ako se u upitu uz ključne reči ne navedu i pridružene informacije na isti način na koji su ugrađene u korpus[16].

- iako se sistem za prikaz konkordanci i dalje uglavnom zasniva na formatu KWIC, njegova funkcionalnost je proširena, tj. sistem omogućava niz opcija za koje je konkordancerima I generacije bila neophodna pomoć posebnih programa (sortiranje konkordanci po levom i desnom kontekstu, generisanje liste korpusnih reči, osnovna statistička analiza korpusa, itd.).

Konkordanceri III generacije

Za konkordancere III generacije je karakteristično sledeće ([McEnery & Hardie, 2012: 40–43]):

- u stanju su da obrađuju elektronske korpuse II generacije veličine od nekoliko desetina do nekoliko stotina miliona reči;

- njihove implementacije podržavaju standard Unicode čime se omogućava razmena teksta na globalnom nivou, izbegava potreba da se za svaki konkordancer posebno razvija podrška za pojedina pisma, kao i da se od korisnika zahteva detaljno poznavanje osmobitnih kodnih rasporeda i liste jezika na koje se primenjuju;

- implementiraju standarde SGML i XML čime se otvara podrška i za korpuse koji su anotirani standardima zasnovanim na SGML-u, odnosno XML-u (TEI, CES/XCES, MULTEXT);

[16] Tipičan primer je pretraga korpusa anotiranih u horizontalnom formatu (v. odeljak 2.4, str. 127), kod kojih je konkordancer II generacije tretirao sekvencu `token_anotacija` kao jedan token i pretraživao na nivou tokena, te za korisnikov upit `token` nije proizvodio nikakav rezultat. Jedini način da korisnik dobije rezultat u takvom slučaju je da upit zada u formatu `token_anotacija`, pod uslovom da zna kako je određeni token anotiran.

- integrišu pomoćne alate za statističku analizu korpusa (generisanje listi učestanosti, n-grama, izračunavanje ključnih reči, kolokaciona analiza, v. odeljak 3.3) koji se mogu uspešno primeniti ne samo na korpusne reči, već i na anotaciju (na primer učestanost lema i određenih vrsta reči, kolokacijske veze određenih vrsta reči i određene korpusne reči, itd.).

Konkordanceri IV generacije

Između konkordancera IV i III generacije nema mnogo razlike u pogledu sredstava korpusne analize koji su na raspolaganju (konkordance, liste učestanosti, kolokacije, ključne reči) već pre svega u načinu na koji korisnik komunicira sa konkordancerom. Razvojem interneta i njegovih servisa, posebno veba, ogroman broj korpusa je, preko **veb-sučelja** (eng. **web interface**), istovremeno postao dostupan ogromnom broju korisnika, nezavisno od platforme (PowerPC, Intel) ili operativnog sistema (Windows, Unix) koji koriste. Pretraga korpusa preko veb-sučelja počiva na klijent-server arhitekturi i vebu kao servisu interneta: na moćnim računarima se nalaze serverski programi koji pretražuju korpus na zahtev **čitača veba** (eng. **web browser**) kao klijentskih programa. Čitač veba se koristi ne samo za prosleđivanje upita, već i za prijem i prikaz formatiranih rezultata pretrage (konkordance), uključujući i njihovu analizu (liste učestanosti, rezultati kolokacione analize, lista ključnih reči, itd.). Veb-sučelje na serverskoj strani je zasnovano ili na sistemu za upravljanje bazama podataka ili na sistemu za indeksiranje i pretraživanje korpusa ili na njihovoj kombinaciji. Sistem za upravljanje bazama podataka koristi standardni upitni jezik SQL za pretraživanje baze podataka i generisanje konkordanci, dok sistem za indeksiranje i pretraživanje korpusa mora imati svoj sopstveni upitni jezik za pretraživanje prethodno indeksiranog korpusa.

Detaljni pregled aktuelnih konkordancera IV generacije je dat u poglavlju 4.

3.3 Statistička analiza korpusa

Liste učestanosti

Jedno od najkorisnijih preliminarnih istraživanja korpusa je kreiranje i analiza njegove liste učestanosti. **Lista učestanosti** ili **frekvencijska lista** (eng. **frequency list**) korpusa je, u opštem slučaju, niz parova koje čine različiti tokeni korpusa (tipovi, v. definiciju 2.12, str. 95) i njihove učestanosti (frekvencije), tj. brojevi koji predstavljaju koliko puta se odgovarajući token pojavljuje u korpusu (Tabela 3.4). Frekvencija tokena može biti predstavljena apsolutno (tačan broj pojavljivanja tokena u korpusu) ili relativno, tj. količnikom apsolutne frekvencije tokena u korpusu i ukupnog broja tokena u korpusu. Apsolutne frekvencije su od interesa jedino prilikom analize korpusa na osnovu kog su generisane, dok su relativne frekvencije uobičajeno sredstvo prilikom poređenja različitih korpusa ili različitih delova korpusa (na primer, pisani i govorni deo korpusa, delovi korpusa koji predstavljaju pojedinačne žanrove, itd.). Pošto su relativne frekvencije brojevi iz intervala $[0, 1]$, obično se izražavaju u procentima, promilima ili čak **milionitim delovima celine** (eng. **parts per million**, skr. **ppm**).

Liste učestanosti se najčešće sortiraju po nekom od kriterijuma koji su bitni za željenu analizu korpusa, pa tako razlikujemo:

1. alfabetski sortirane liste učestanosti,
2. sufiksno sortirane liste učestanosti,
3. liste učestanosti sortirane po opadajućoj frekvenciji (frekvencijski sortirane liste učestanosti),
4. liste učestanosti sortirane po redosledu prvih pojavljivanja tokena u korpusu;
5. liste učestanosti sortirane po dužini tokena;

Alfabetski sortirane liste učestanosti su najpogodnije za konstatovanje frekvencije određenog tokena u korpusu, kao i za izradu rečnika korpusa. Ako korpus nije lematizovan, a jezik korpusa, kao što je slučaj sa srpskim jezikom, ima osobinu da svi

Tabela 3.4: Prvih 40 redova liste učestanosti Korpusa savremenog srpskog jezika (verzija SrpKor 2013, 122.255.064) korpusnih reči)

rang (r)	**token**	**apsolutna frekvencija** (f)	**relativna frekvencija**	**kumulativna frekvencija**	**proizvod** $r \cdot f$ (**Zipfov zakon**)
1	i	4.330.865	3,54%	3,54%	4.330.865
2	je	4.103.542	3,36%	6,90%	8.207.084
3	u	3.513.009	2,87%	9,77%	10.539.027
4	da	3.261.285	2,67%	12,44%	13.045.140
5	se	2.107.336	1,72%	14,16%	10.536.680
6	na	1.751.270	1,43%	15,60%	10.507.620
7	za	1.381.402	1,13%	16,73%	9.669.814
8	su	1.258.361	1,03%	17,76%	10.066.888
9	od	919.922	0,75%	18,51%	8.279.298
10	sa	779.469	0,64%	19,15%	7.794.690
11	a	740.476	0,61%	19,75%	8.145.236
12	koji	650.144	0,53%	20,28%	7.801.728
13	ne	612.218	0,50%	20,78%	7.958.834
14	o	517.105	0,42%	21,21%	7.239.470
15	će	505.643	0,41%	21,62%	7.584.645
16	iz	501.473	0,41%	22,03%	8.023.568
17	to	474.120	0,39%	22,42%	8.060.040
18	kao	460.343	0,38%	22,79%	8.286.174
19	što	445.434	0,36%	23,16%	8.463.246
20	bi	417.727	0,34%	23,50%	8.354.540
21	nije	388.057	0,32%	23,82%	8.149.197
22	do	337.396	0,28%	24,09%	7.422.712
23	U	329.179	0,27%	24,36%	7.571.117
24	ali	327.402	0,27%	24,63%	7.857.648
25	ili	322.658	0,26%	24,90%	8.066.450
26	koje	307.541	0,25%	25,15%	7.996.066
27	po	287.841	0,24%	25,38%	7.771.707
28	godine	280.101	0,23%	25,61%	7.842.828
29	sam	263.985	0,22%	25,83%	7.655.565
30	koja	259.146	0,21%	26,04%	7.774.380
31	samo	231.137	0,19%	26,23%	7.165.247
32	sve	227.913	0,19%	26,41%	7.293.216
33	kako	220.886	0,18%	26,60%	7.289.238
34	Srbije	215.324	0,18%	26,77%	7.321.016
35	više	204.516	0,17%	26,94%	7.158.060
36	biti	197.386	0,16%	27,10%	7.105.896
37	bio	194.118	0,16%	27,26%	7.182.366
38	još	183.095	0,15%	27,41%	6.957.610
39	već	181.356	0,15%	27,56%	7.072.884
40	s	180.243	0,15%	27,70%	7.209.720

ili velika većina flektivnih oblika određene leme dele zajednički levi faktor (prefiks u smislu teorije formalnih jezika), tada se u alfabetski sortiranoj listi učestanosti ti oblici nalaze jedni pored drugih, što olakšava i analizu frekvencije same leme. Iz istog razloga je alfabetski sortirana lista učestanosti pogodna za statističku analizu pojedinih tipova derivacije (prefiksacija, izvođenje, slaganje ili kompozicija).

Sufiksno sortirane liste učestanosti takođe koriste alfabetsko sortiranje niski, ali zdesna nalevo (inverzno alfabetsko sortiranje), tako da će se u listi naći jedne do drugih niske koje imaju zajednički desni faktor (sufiks). Ovako sortirane liste su takođe pogodne za statističku analizu pojedinih tipova derivacije, kao i za izradu obratnog rečnika.

Liste učestanosti sortirane po opadajućoj frekvenciji su zanimljive za analizu najučestalijih tokena u korpusu. Ukoliko je korpus referentan, poređenjem frekvencijski sortirane liste nekog teksta sa frekvencijski sortiranom listom korpusa može se utvrditi da li neki tokeni teksta imaju veću relativnu učestanost nego što referentni korpus sugeriše, na osnovu čega se mogu izvoditi pretpostavke o ključnim rečima teksta, domenu teksta, autorstvu teksta, itd. Takođe, ako je korpus semantički anotiran, tj. ako je svakom tokenu pridružena informacija o njegovom značenju, moguće je generisati listu učestanosti sortiranu po frekvencijama upotrebljenih značenja. Informaciju o učestalosti upotrebe značenja neke lekseme koriste leksikografi prilikom obrade pojedinačne lekseme u rečniku tako što najpre navode najčešće korišćena značenja lekseme u jeziku, a potom i ostala značenja po opadajućem redosledu njihovih frekvencija.

Liste učestanosti sortirane po redosledu prvih pojavljivanja tokena u korpusu se koriste u slučaju kada su neophodne informacije o organizaciji teksta. Jedna moguća primena je detekcija pozicija u tekstu gde je došlo do promene teme ([Barnbrook, 1996: 49]), pri čemu se uporedo koristi i frekvencijski sortirana lista učestanosti. Ako se tokeni „slične" visoke učestanosti prvi put pojavljuju na međusobno udaljenim pozicijama u tekstu, tada su njihove pozicije kandidati za mesta gde je došlo do promene teme u tekstu.

Ako je korpus anotiran morfološki ili sintaksički, pridružene informacije se mogu iskoristiti prilikom generisanja svih navedenih tipova listi. Na primer, ako je korpus

lematizovan, pored generisanja liste učestanosti tokena, moguće je proizvesti i listu učestanosti lema u korpusu, što je za pojedine primene svakako značajniji rezultat. Na sličan način se mogu odrediti učestanosti drugih lingvističkih jedinica u tekstu (sintagme, fraze, višečlane lekseme, itd.) ukoliko su eksplicitno anotirane u tekstu.

Generisanje liste učestanosti jedinica teksta (tokena, lema, sintagmi, rečenica, itd.) zahteva da se te jedinice prethodno identifikuju u tekstu, što u praksi podrazumeva, pre svega, tokenizaciju (v. odeljak 2.3, str. 92), a po potrebi i lematizaciju, parsiranje, određivanje granice između rečenica, itd. Prilikom prebrojavanja pojavljivanja pojedinih jedinica teksta, neophodno je precizirati koji se tokeni tretiraju kao istovetne jedinice, pri čemu izrazite teškoće stvaraju homografija, (ne)razlikovanje velikih i malih slova, kao i varijacije pravopisa[17]. Na primer, prilikom računanja učestanosti veznika `ako` deluje kao prirodna odluka ne razlikovati velika i mala slova, a time ni tokene `Ako` na početku rečenice i `ako` u sredini rečenice. Međutim, ako se ne razlikuju velika i mala slova, što se obično postiže konverzijom svih slova teksta u mala slova, neće se praviti razlika između predloga `u` i imena `U (Tant)`, odnosno veznika `ali` i imena `(Muhamed) Ali`.

Na osnovu podataka iz liste učestanosti moguće je izračunati:

- ukupan broj tokena, korpusnih reči, tipova, korpusnih tipova (Tabela 3.5);
- prosečnu frekvenciju tipova korišćenih u tekstu kao **količnik ukupnog broja tokena i ukupnog broja tipova** (eng. **token type ratio**).

Tabela 3.5: Ukupan broj tokena, tipova, korpusnih reči i korpusnih tipova i njihov odnos u Korpusu savremenog srpskog jezika (verzija SrpKor 2013)

tokeni	**tipovi**	**korpusne reči**	**korpusni tipovi**	**tokeni/ tipovi**	**korpusni tokeni/ korpusni tipovi**
152.540.721	1.424.899	122.255.064	1.402.664	107,05	87,16

Liste učestanosti predstavljaju empirijski materijal za proučavanje statističkih raspodela u jeziku. Jedna od zanimljivih zakonitosti koju ilustruju liste učestanosti

[17]Varijacije pravopisa su posebno izražene kod dijahronijskih korpusa koji pokrivaju duže vremenske periode. Međutim, i periode poput jednog veka mogu obeležiti česte promene pravopisa, što najbolje ilustruju dnevne novine *Politika* koje, uz prekide tokom okupacije, izlaze od 1904. godine. U njima se početkom XX veka pronalaze tokeni `austriski`, `pretsednik`, itd. kojima danas redom odgovaraju tokeni `austrijski`, `predsednik`.

jeste da u prirodnim jezicima postoji malo uobičajenih, visoko frekventnih reči[18] i mnogo reči niske učestalosti, a da se ostale reči po broju i frekvenciji nalaze između ova dva ekstrema. Pri tome, najveći broj najučestalijih korpusnih reči predstavljaju funkcijske reči iz zatvorenih klasa reči (predlozi, veznici, itd.), a slični rezultati važe i za pojedinačne tekstove. U slučaju kada se porede liste učestanosti pojedinačnih tekstova sa listom učestanosti referentnog korpusa, obično se primenjuju filteri za uklanjanje funkcijskih reči, tj. analiziraju se najfrekventniji tokeni koji jesu tekstuelne nefunkcijske reči.

Još 1916. godine ([Estoup, 1916]) je primećena veza između frekvencije tokena i njegovog **ranga** (eng. **rank**), tj. pozicije tokena u listi učestanosti sortiranoj opadajuće po frekvencijama, ali je široj javnosti postala poznata zahvaljujući Zipfovim radovima ([Zipf, 1929, 1935, 1949]), po kome se i danas naziva **Zipfovim zakonom** ([Manning & Schütze, 1999: 25–29]). Ako u listi učestanosti sa r označimo rang tokena čija je frekvencija jednaka f, tada Zipfov zakon tvrdi da su te dve veličine obrnuto proporcionalne, odnosno da za tu listu učestanosti postoji konstanta k takva da za sve tokene liste važi

$$f \cdot r = k. \tag{3.1}$$

U praksi se Zipfov zakon pokazuje kao grubo predviđanje statističke raspodele tokena u korpusu, tj. grafikoni lista učestanosti na kojima je prikazana zavisnost frekvencije od ranga tokena pokazuju odstupanja, posebno za tokene sa početka i kraja liste, tj. sa najvišom i najnižom frekvencijom. U pokušaju da se Zipfov zakon preciznije formuliše kako bi se smanjila odstupanja od empirijskih podataka, ponuđena je opštija formula ([Mandelbrot, 1954]):

$$f = P \cdot (r + \rho)^{-B}, \tag{3.2}$$

pri čemu su P, ρ, B parametri teksta koji mere korišćenje reči u tekstu. Za $\rho = 0$ i $B = 1$ formula se svodi na Zipfov zakon ($P = k$).

Glavna posledica formula 3.1 i 3.2 je predviđanje da će čak i u velikim korpusima biti mnogo reči sa niskom učestanošću, što otežava izučavanje njihove upotrebe u

[18] Tabela 3.4 ilustruje da pojavljivanja 40 najučestalijih korpusnih reči u Korpusu savremenog srpskog jezika (verzija SrpKor 2013) predstavljaju 27,70% svih pojavljivanja korpusnih reči.

jeziku ([Manning & Schütze, 1999: 23]). Zipfov zakon se može iskoristiti da bi se procenila veličina korpusa (s) u kome će se token određenog ranga (r) pojaviti sa željenom frekvencijom (f), pod pretpostavkom da je poznata frekvencija (f_1) najučestalijeg tokena, tj. tokena ranga 1 ([Koehn, 2010: 38]):

$$s = \frac{f \cdot r}{f_1}. \tag{3.3}$$

Kolokacije

U lingvistici se pod **kolokacijom** (eng. **collocation**) podrazumevaju „uobičajene leksičke veze (...), ali bez formiranja čvrstih frazeoloških jedinica", koje mogu da prerastu „u kliše neprekidnim ponavljanjem ili pak srastanjem i okoštavanjem njihovih elemenata" ([Bugarski, 1995: 132]). U korpusnoj lingvistici kolokacijom se smatra fenomen koji se manifestuje time što je „verovatnije da se u određenim kontekstima pojedine reči pojavljuju u kombinacijama sa nekim drugim rečima" ([Baker et al., 2006: 36]), odnosno time što „pojedine lekseme imaju tendenciju da se prilikom upotrebe prirodnog jezika češće zajedno pojavljuju nego što bi to diktirale sintaksa i semantika" ([Brown, 2005], odrednica *Collocations*). Pri tome, lekseme ne moraju da se nađu jedna pored druge u tekstu da bi predstavljale kolokaciju, već na nekom „malom rastojanju (...) od najviše četiri reči između" ([Sinclair, 1991: 170]). U tom slučaju se definiše **okolina** lekseme u tekstu kao skup koji sadrži određen broj tekstuelnih reči ispred i iza te lekseme, sami elementi te okoline se nazivaju **kolokatima** (eng. **collocate**) lekseme, a broj kolokata u okviru okoline naziva se **širina okoline**.

Kolokacije se u tekstu realizuju kao posledica:

- leksičkih relacija: višečlane lekseme (eng. compounds, na primer `beli luk`), idiomi (`preko noći`), ili pak izrazi posebnog značenja koji su u nekom periodu izuzetno učestali i skoro da postaju idiomi (`borba protiv korupcije i organizovanog kriminala`, `nepravedne i ničim (ne)izazvane sankcije`[19]);

[19] Češće se koristi nepravilni izraz `ničim izazvane` sa izostavljenom negacijom, što je istaknuto zagradama.

- sintaksičkih relacija: atribut-imenica (`prošle godine`), glagol-objekat (`voditi pregovore`, `potpisati sporazum`, `izraziti zahvalnost`), odredba-glagol (`teško ostvariti`);
- relacija koje uopšte nisu lingvističke prirode.

Identifikacija kolokacija u tekstu, odnosno korpusu, naziva se kolokaciona analiza. Ona je od posebnog interesa za leksikologiju, leksikografiju, nastavu jezika kao stranog, prevođenje sa jednog jezika na drugi.

Za razliku od analize konkordanci koja je, pre svega, kvalitativna i manuelna analiza, kolokaciona analiza je kvantitativna, zasnovana na statističkim testovima značajnosti i može se automatizovati. Testovi značajnosti se koriste kako bi se utvrdilo da li je razlika između očekivanog i realizovanog obrasca ponašanja kolokata u tekstu značajna, tj. da li oni zaista predstavljaju kolokaciju ili ne.

Dobar deo današnjih konkordancera sadrži alate koji kolokacionu analizu obavljaju automatski (v. poglavlje 4). Ukoliko se ispituju kolokacije u kojima se pojavljuje određena ključna reč, analiza se obavlja u nekoliko koraka ([Barnbrook, 1996: 91–94]):

(i) Polaznu tačku predstavljaju konkordance zadate ključne reči. Redukcijom konkordanci na okoline ključne reči, obično po pet korpusnih reči sa leve i desne strane, formira se **prozor** (eng. **window**) ili **opseg** (eng. **span**) koji se dalje analizira. Stoga je u ovom koraku neophodno izabrati širinu okoline, odnosno prozora, a shodno tome i dovoljnu širinu levog i desnog konteksta u originalnim konkordancama kako bi one mogle da se redukuju na prozor željene širine.

(ii) Generiše se lista učestanosti kolokata u prozoru, tj. određuje se sa kojom apsolutnom frekvencijom se korpusne reči pojavljuju u okolini ključne reči. Dobijene učestanosti se još nazivaju uočenim ili realizovanim (apsolutnim) učestanostima.

S obzirom da funkcijske reči i korpusne reči sa niskom apsolutnom učestanošću u okviru prozora „iskrivljuju" rezultate statističkih testova značajnosti, generisana lista učestanosti se filtrira. U slučaju funkcijskih reči filtriranje se postiže

ili pomoću njihove liste ekstrahovane iz elektronskog rečnika ili, ako je korpus anotiran, pomoću liste vrsta reči kojima pripadaju (predlozi, veznici, rečce, itd.). Korpusne reči, čija je apsolutna učestanost u prozoru niska, eliminišu se definisanjem donje granice za učestanost, tako da se u obzir uzimaju samo kolokati čija je apsolutna frekvencija u prozoru dovoljno velika.

Ako se istražuju sintaksičke kolokacije u okviru morfološki anotiranog korpusa, moguće je izvršiti filtriranje pomoću regularnih izraza tako da se kao rezultat dobiju samo kolokati koji pripadaju određenoj vrsti reči, na primer parovi pridev-imenica ili glagol-imenica.

(iii) Sledeći korak je određivanje očekivanih apsolutnih učestanosti kolokata u prozoru za šta je potrebno izabrati:

(a) jezički model kojim se pretpostavlja raspodela kolokata u jeziku;

(b) uzorak jezika na osnovu kog se izračunavaju očekivane relativne učestanosti kolokata u jeziku.

Jezički model se namerno bira tako da se ne pretpostavlja nikakva veza između kolokata i ključne reči, kako bi prilikom merenja odstupanja realizovanih od očekivanih apsolutnih učestanosti mogao da se donese zaključak da li kolokat i ključna reč zaista formiraju kolokaciju ili ne. U praksi se najčešće koristi **slučajna raspodela** (eng. **random distribution**) koja za očekivanu relativnu učestanost kolokata u jeziku uzima njegovu relativnu frekvenciju u izabranom uzorku jezika. U najjednostavnijem slučaju, kada se vrši kolokaciona analiza jednog teksta, ponekad se za uzorak jezika uzima taj tekst u celini. Pošto je jedan tekst daleko od reprezentativnog uzorka jezika, kao izvor podataka se obično koristi referentni korpus, poželjno što veći, a očekivana apsolutna učestanost posmatranog kolokata u prozoru se izračunava kao proizvod veličine prozora (ukupan broj svih kolokata u prozoru) i relativne frekvencije posmatranog kolokata u izabranom referentnom korpusu.

(iv) Poslednji korak je primena nekog od statističkih testova, odnosno mera, značajnosti kako bi se ispitala razlika između realizovanih (uočenih) i očekivanih apso-

lutnih učestanosti kolokata u prozoru. Rezultat se smatra statistički značajnim ukoliko je verovatnoća njegove slučajne realizacije dovoljno mala. Za merenje „jačine" kolokacije istraživači obično koriste z-test, Studentov t-test, Pirsonov χ^2-test, **zajedničku informaciju** (eng. **mutual information**, skr. **MI**), **test logaritamske verodostojnosti** (eng. **log likelihood**) kao statističku meru, odnosno test značajnosti ([Baker et al., 2006; Barnbrook, 1996; Manning & Schütze, 1999; Oakes, 1998]). t-test i test logaritamske verodostojnosti ocenjuju kolokaciju „jačom" ako su i parovi kolokata i sami kolokati visoke učestanosti, dok je za MI dovoljno da su parovi kolokata dovoljno učestali, dok pojedinačni kolokati mogu biti sa niskom učestanošću.

Tabela 3.6: Formule za računanje statističkih mera z-test, Studentov t-test i MI ([Barnbrook, 1996: 93-99]). O i E redom označavaju uočene (realizovane) i očekivane apsolutne učestanosti pojedinačnog kolokata ključne reči, dok je σ standardna devijacija kolokata. Ako N predstavlja veličinu prozora (ukupan broj kolokata u prozoru), a p — relativnu učestanost kolokata u referentnom korpusu, tada je $E = N \cdot p$ i $\sigma = \sqrt{N \cdot p \cdot (1-p)}$.

z-test	Studentov t-test	MI
$z = \frac{O-E}{\sigma}$	$t = \frac{O-E}{\sqrt{O}}$	$MI = \log \frac{O}{E}$

N-grami

N-grami (eng. **n-grams**) predstavljaju uopštenje listi učestanosti u smislu da se njima ne predstavljaju pojedinačni tokeni i njihove frekvencije već učestanosti sekvenci n uzastopnih tokena u korpusu, gde je n prirodan broj i $n \geq 2$. U slučaju kada je $n = 2$, odnosno $n = 3$, n-grami se nazivaju **bigramima** (eng. **bigrams**), odnosno **trigramima** (eng. **trigrams**). Filtriranjem n-grama visoke učestanosti, tako da se eliminišu parovi sa funkcijskim rečima, dobijaju se kandidati za razne tipove kolokacija (višečlane lekseme, idiomi, itd.). Sem za detekciju kolokacija, n-grami se koriste u raznim statističkim alatima za obradu prirodnog jezika (programi za automatsku anotaciju teksta, mašinsko prevođenje, utvrđivanje autorstva, itd.)

4

Pregled postojećih alata za rad sa korpusom

Alati za rad sa korpusom ili **korpusni alati** (eng. **corpus tools**) se prema svojoj nameni grubo mogu podeliti na sledeće klase softvera:

- alati za pripremu korpusnih tekstova,
- alati za anotaciju korpusnih tekstova,
- alati za pripremu paralelnih tekstova,
- alati za kompilaciju korpusa,
- alati za pretraživanje korpusa (konkordanceri u užem smislu reči),
- alati za statističku analizu korpusa.

Između predstavnika prvobitnih generacija navedenih klasa korpusnih alata su postojale jasne razlike s obzirom da su ti programi pisani sa namenom da obavljaju najčešće jednu funkciju. Alati pisani za operativni sistem Unix i njegove derivate i danas primenjuju pravila poznata kao „Unix-filozofija", među kojima su upravo pravila o tome da programi treba da bude jednostavni, da svaki kvalitetno obavlja samo jednu funkciju, pri čemu programi mogu lako da komuniciraju jedni sa drugima, a složeni zadatak se može obaviti kombinovanjem (kompozicijom) više takvih alata.

S druge strane, razvijeni su i programski sistemi za rad sa korpusom koji se trude da budu „sveobuhvatni" u pogledu svojih funkcija. Iako krajnjem korisniku izgledaju kao jedan program, takvi sistemi su ili dobijeni integrisanjem programskih modula od kojih svaki obavlja određenu funkciju (anotaciju, paralelizaciju, kompilaciju, itd.) u jedan programski sistem, ili se i fizički realizuju kao skup više alata (u duhu „Unix-filozofije") sa jednim „glavnim" programom, sa kojim korisnik komunicira i preko koga koristi pojedinačne alate. U potonjem slučaju, pojedinačni alati se mogu koristiti i nezavisno od programskog sistema (odnosno „glavnog" programa), ali po pravilu isključivo iz komandne linije, dok „glavni" program omogućava udobnije korišćenje, zahvaljujući svom **grafičkom korisničkom sučelju** (eng. **Graphical User Interface**, skr. **GUI**). Upravo zbog težnje ka integraciji različitih korpusnih alata kako bi se olakšao rad korisniku, karakteristične za novije generacije korpusnih alata, teško je povući jasnu granicu između pojedinih programa koji se ubrajaju u korpusne alate, odnosno sistema korpusnih alata.

Pojedine grupe alata, poput alata za pripremu korpusnih tekstova, obuhvataju i programe sa opštijom namenom kao što su aplikacije za konverziju iz jednog tekstuelnog formata u drugi, aplikacije za uređivanje teksta (uključujući i programe za uređivanje, validaciju i transformisanje XML-dokumenata), aplikacije za skaniranje i optičko prepoznavanje karaktera, itd. Stoga je, s obzirom na broj svih postojećih korpusnih alata, nemoguće dati njihov kompletan prikaz[1]. Stoga će u nastavku poglavlja biti opisani aktuelni integrisani sistemi korpusnih alata i specifični korpusni alati od značaja (konkretno, korpusni alati za veb).

Pregled svih alata za anotaciju korpusnih tekstova takođe izlazi iz okvira ovog rada. Svaki pokušaj upoređivanja raznih pristupa koje koriste alati za anotaciju bi neminovno doveo do predstavljanja različitih formalizama zasnovanih na pravilima, mašinskom učenju ili njihovoj kombinaciji (v. odeljak 2, str. 142). U odeljku 6.3 dat je pregled alata koji su razmatrani kao kandidati za morfološku anotaciju tekstova Korpusa savremenog srpskog jezika, dok se detaljan uporedni pregled alata za morfološku anotaciju tekstova na srpskom jeziku može naći u [Popović, 2008, 2010].

[1]Jedan pokušaj je stranica Dejvida Lija (David Lee) na adresi `http://www.uow.edu.au/~dlee/software.htm`, što je samo deo njegove bogate kolekcije referenci posvećenih korpusnoj lingvistici (`http://www.uow.edu.au/~dlee/CBLLinks.htm`).

4.1 Parametri korpusnih alata

Prilikom analize i izbora konkretnih korpusnih alata potrebno je uzeti u obzir sledeće parametre:

Licenca Softverskom licencom se reguliše korišćenje, menjanje i distribucija softvera. U zavisnosti od prateće licence postoje različite vrste softvera:

- **slobodan softver** (eng. **free software**),
- **softver kao javno dobro** (eng. **public domain software**).
- **besplatan softver** (eng. **freeware**),
- **softver otvorenog koda** (eng. **open source software**),
- **vlasnički softver** (eng. **proprietary software**),
- **komercijalni softver** (eng. **commercial software**).

Iako se neke od navedenih vrsta softvera međusobno preklapaju (na primer komercijalni i vlasnički softver, odnosno softver kao javno dobro, slobodni i besplatni softver), između njih postoje i bitne razlike ([Krstev, 2010]). Slobodni softver je uvek besplatan, dok obrnuto ne važi. Besplatan softver ne mora biti ni javno dobro, niti softver otvorenog koda, tj. može biti i vlasnički softver. Po nekim definicijama vlasnički i komercijalni softver su istovetni, dok druge definicije smatraju da je komercijalni softver isključivo softver koji je na prodaju, dok vlasnički softver može da bude i besplatan.

Slobodni softver otvorenog koda ima višestruke prednosti ([Krstev, 2010]):

- besplatan je;
- moguće je modifikovati izvorni kôd što dozvoljava permanentno unapređivanje softvera i produžava mu životni vek u odnosu na vlasnički i komercijalni softver;
- minimalizovana je zavisnost korisnika od autora softvera;
- poštuje smernice otvorenih standarda;

- korisnici slobodnog softvera su organizovani u mreže preko kojih razmenjuju iskustva vezana za prevođenje izvornog koda, instalaciju i korišćenje softvera, otklanjanje uočenih grešaka u programu;
- otvoren kôd doprinosi većoj pouzdanosti softverskih sistema i sigurnosti podataka koje sistem koristi, a takođe i olakšava komunikaciju dva ili više softverskih sistema.

Za razliku od licenci vlasničkog softvera, koje se uglavnom bave zaštitom autorskih prava, licence slobodnog softvera (Tabela 4.1) su ili posvećene zaštiti slobode softvera (eng. **copyleft licence**) ili pak prenosom svih prava (korišćenje, modifikacija, distribuiranje) na korisnika što je karakteristika **nerestriktivnih licenci** (eng. **permissive licence**)). Da bi se postigla veća fleksibilnost, neprofitna organizacija Creative Commons kreirala je više licenci slobodnog softvera koje kombinuju prenos i restrikciju različitih prava (kopiranje, modifikacija, uslovi distribuiranja, upotreba u komercijalne svrhe, itd.).

Platforma Pod platformom se podrazumevaju hardverski i softverski preduslovi za korišćenje programa, preciznije arhitektura računara, operativni sistem i programske biblioteke kao okruženje u kom program može da radi. Ukoliko je program otvorenog koda, postoji teorijska mogućnost da se izvorni kôd prevede u izvršni oblik ili interpretira na svakoj platformi. Međutim, u praksi se dešava da je program pisan za jednu određenu platformu, tako da u vreme izvršavanja zavisi od biblioteka programa specifičnih za određeni operativni sistem, što otežava prenosivost programa sa jedne platforme na drugu. Takođe, autori programa ponekad nisu ni zainteresovani da učine programe dostupnim za više različitih platformi. Jedno od novijih rešenja problema prenosivosti je korišćenje programskih jezika poput Jave, čiji se izvorni kôd ne prevodi u mašinski (zavisan od arhitekture računara), već u međukod, nezavisan od platforme, koji se intepretira pomoću virtuelne mašine.

Klijent-server arhitektura i veb Klijent-server arhitektura je jedno od osnovnih svojstava interneta koje se sastoji u tome da dve aplikacije, klijent i server, na umreženim računarima međusobno komuniciraju tako što klijent šalje

Tabela 4.1: Primeri softverskih licenci

naziv licence	**logo**	**tip licence**
BSD		na korisnika se prenose sva prava (nerestriktivna licenca)
GNU GPL		distribucija derivata pod istim uslovima kao i original (*copyleft*-licenca)
LGPL		distribucija softvera nastalog povezivanjem LGPL-softvera i ma kakvog drugog softvera se može odvijati po proizvoljnim uslovima, osim u slučaju direktnog derivata koji se distribuira pod istim uslovima kao i original (LGPL se obično koristi u slučaju kada je potrebno da se slobodni softver iskoristi kao dinamička biblioteka u okviru vlasničkog softvera)
CC BY		obavezno navođenja autora originala
CC BY-SA		obavezno navođenja autora originala, distribucija derivata pod istim uslovima kao i original
CC BY-ND		obavezno navođenja autora originala, zabranjeno kreiranje derivata
CC BY-NC		obavezno navođenja autora originala, dozvoljena isključivo nekomercijalna upotreba
CC BY-NC-ND		obavezno navođenja autora originala, dozvoljena isključivo nekomercijalna upotreba, zabranjeno kreiranje derivata
CC BY-NC-SA		obavezno navođenja autora originala, dozvoljena isključivo nekomercijalna upotreba, distribucija derivata pod istim uslovima kao i original

određeni zahtev serveru, a server odgovara klijentu tako što zahtev ispunjava ili odbija. Tipičan primer te komunikacije je popularno „surfovanje na vebu" gde se programi Internet Explorer, Mozzila Firefox, Google Chrome, Opera, itd., **čitači veba** (eng. **browser**) koriste kao klijenti pomoću kojih se šalje zahtev za određenom stranicom ili resursom na vebu ukucavanjem njegove adrese (URL).

Najveći broj korpusnih alata su aplikacije koje korisnik izvršava na svom računaru. Međutim, ogromne veličine savremenih elektronskih korpusa i pojava interneta su eliminisale potrebu da korisnik ima kopiju korpusa na svom računaru, kao i dovoljno moćan hardver koji bi sproveo pretragu velikog korpusa. Umesto toga, veliki korpusi su danas smešteni na moćnim računarima sa serverskim aplikacijama zaduženim za pretragu, dok korisnik na svom računaru koristi klijentsku aplikaciju preko koje zadaje upit za pretragu korpusa. U najvećem broju slučajeva se koristi klijent-server arhitektura veba, tj. klijentske aplikacije za pretragu korpusa su upravo čitači veba, a korisnik zadaje upit preko stranice na vebu, odnosno **veb-sučelja** (eng. **web interface**).

Ažurnost i podrška Pod ažurnošću se podrazumeva da li i koliko često autor programa ispravlja uočene i prijavljene greške, kao i da li nove verzije sadrže nove funkcionalnosti. Podrška se odnosi na mogućnost da korisnik programa uz program dobije i neophodnu dokumentaciju za korišćenje, kao i da li može da računa na dodatnu pomoć pri korišćenju programa, obično preko odgovora na listu najčešće postavljanih pitanja, bilo od samog autora, bilo od organizovane zajednice korisnika programa koji razmenjuju svoja iskustva.

Proširivost Ovaj parametar opisuje stepen integrisanosti pojedinačnih modula u programski sistem, tj. odgovara na pitanje da li korisnik može prilagoditi sistem dodavanjem novih eksternih alata i da li se to može izvesti u okviru postojećeg sučelja ili je neophodno modifikovati izvorni kôd programa.

Kada su u pitanju korpusni alati kao sistemi nastali integrisanjem više alata, poželjno je razmotriti i sledeće parametre koji se, pre svega, odnose na prisustvo-/odsustvo nekog alata/modula:

Prethodna obrada Prethodna obrada može da obuhvati širok spektar aktivnosti, od uređivanja teksta, optičkog prepoznavanja karaktera, preko konverzije teksta iz jednog formata u drugi, uključujući i konverziju kodnih rasporeda, zaključno sa indeksiranjem teksta radi efikasnije pretrage.

Jezički resursi Pod jezičkim resursima se podrazumeva „skup jezičkih podataka i opisa u mašinski čitljivom obliku koji se koriste za razvijanje, unapređivanje ili evaluaciju algoritama ili sistema za obradu prirodnog jezika" ([Stanković, 2009]). Korpusni alati koriste razne vrste jezičkih resursa:

- korpuse koje pretražuju i analiziraju;
- referentne korpuse koje konsultuju za potrebe statističke analize;
- leksičke resurse poput elektronskih rečnika, semantičkih mreža, višejezičnih baza podataka, ontologija;
- gramatičke resurse (banke sintaksičkih stabala, itd).

Tipovi pretrage Sistem integrisanih korpusnih alata obavezno sadrži i modul za pretragu (konkordancer u užem smislu reči). Ovaj modul omogućava zadavanje upita korišćenjem odgovarajućeg upitnog jezika, a rezultate najčešće prezentuje u vidu konkordanci. Većina upitnih jezika zasnovana je na upotrebi regularnih izraza ili nekog formalizma koji im je ekvivalentan (konačni automati, desno linearne gramatike). Pojedini upitni jezici omogućavaju pretragu anotiranih korpusa po pridruženim informacijama, kako lingvističkim, tako i metapodacima koji se odnose na tekstove korpusa ili sam korpus. Upitni jezici takođe mogu raspolagati opcijom za pretraživanje paralel(izova)nog teksta, pri čemu se kao rezultat dobijaju paralel(izova)ne konkordance.

Za potrebe prikaza konkordanci, modul za pretragu može raspolagati sledećim opcijama:

- sortiranje konkordanci po ključnim rečima i levom i desnom kontekstu;
- pseudoslučajan redosled prikazanih konkordanci;
- zadavanja maksimalnog broja konkordanci ili broja konkordanci po stranici ispisa;

- zadavanje širine levog, odnosno desnog konteksta;
- prikaz ili odsustvo pratećih metapodataka, odnosno lingvističke anotacije;
- izmena formata u kom se prikazuju konkordance (KWIC, konkordance kao niz rečenica, pasusa, itd.);
- izbor željenih konkordanci i njihovo preuzimanje u formi datoteke;
- navigacija kojom se omogućava direktan pristup početnoj, krajnjoj ili proizvoljnoj stranici ispisa konkordanci.

Raspoložive statističke funkcije Već iz naziva parametra je jasno da se on odnosi na prisustvo modula za statističku analizu korpusa čije funkcionalnosti mogu obuhvatiti generisanje i sortiranje listi učestanosti (tipova, lema, vrsta reči, itd.), generisanje n-grama, kolokacionu analizu, određivanje ključnih reči teksta, itd.

4.2 Korpusni alati za veb

Postoje dva glavna pristupa eksploataciji podataka sa veba ([Bergh & Zanchetta, 2008: 315]):

- **veb kao korpus** (eng. **Web as Corpus**, skr. **WaC**) i
- **veb za korpus** (eng. **Web for Corpus**, skr. **WfC**).

Pristup WaC direktno koristi veb kao korpus, dok se WfC bavi kompilacijom korpusa na osnovu tekstova sa veba. Međutim, u literaturi se često istim imenom (WaC) označavaju oba pristupa, bez obzira na navedene razlike.

Tipičan primer pristupa WaC je zadavanje obrasca upotrebe određene lekseme u formi upita za neki od pretraživača interneta (Google, Yahoo, Bing, itd.). Međutim, pretraživači su pravljeni za potrebe pronalaženja informacija, tako da sa stanovišta korpusne lingvistike imaju više nedostataka:

- nedovoljno izražajan upitni jezik;

- diskutabilno je da li dobijeni rezultati potiču iz tekstova na odgovarajućem jeziku, tj. koliko precizno pretraživači „pogađaju" jezik teksta na vebu;
- u opštem slučaju nije moguće suziti pretragu po određenom funkcionalnom stilu ili domenu teksta;
- anotacija i statističke funkcije se ne mogu primeniti dok se tekstovi ne prebace na lokalni računar;
- pretraživači ne omogućavaju pretragu celokupnog veba već samo onog dela koji su indeksirali;
- rezultat upita varira u zavisnosti od korisnikove istorije prethodnih upita, kao i od vremenskog trenutka kad je upit postavljen (neke stranice na vebu „nestaju", umesto njih se pojavljuju nove, a vremenom pretraživač menja i ocene relevantnosti koje dodeljuje stranicama);
- rangiranje pojedinačnih rezultata upita zavisi od faktora koji su najčešće lingvistički nerelevantni;
- format rezultata ne odgovara uobičajenom formatu konkordanci (prikazuje se samo jedan pogodak po stranici sa kratkim kontekstom i vezom na stranicu).

Iz tog razloga su razvijeni i posebni alati, **veb-konkordanceri** (eng. **web concordancer**), za pretragu veba kao korpusa. Veb-konkordanceri su zasnovani na klijent-server arhitekturi, pri čemu se kao klijentska aplikacija koristi čitač veba. Primeri veb-konkordancera su WebCorp[2], WebAsCorpus.org[3], WebCONC[4], GlossaNet[5]. Detaljan pregled veb konkordancera izlazi izvan okvira ovog rada.

U slučaju pristupa WfC takođe se mogu razlikovati dve vrste alata. Prvu grupu čine alati namenjeni preuzimanju proizvoljnih resursa sa veba (eng. download manager), uključujući kompletne veb sajtove, aplikacije, multimedijalne sadržaje, itd.

[2] `http://www.webcorp.org.uk/live`. Više detalja se može naći u [Gatto, 2009; Renouf, 2003; Renouf et al., 2006: 79–100]

[3] `http://webascorpus.org`

[4] `http://gandalf.uib.no/lingkurs/templates_c/%25%256E%5E6ED%5E6EDF58DD%25%25labor.html.php`

[5] `http://glossa.fltr.ucl.ac.be`. Videti, na primer, [Fairon, 2000]

Umesto navođenja svih mogućih rešenja, kao ilustraciju je dovoljno pomenuti wget[6], jedan od standardnih slobodnih softverskih paketa koji se isporučuje sa distribucijama operativnog sistema Linux.

Potonju grupu WfC-alata čine programi sa specifičnom namenom da, na osnovu tekstova sa veba, kompiluju korpus spreman za pretragu i analizu. Dva najpoznatija WfC-alata su KWiCFinder ([Fletcher, 2004a,b, 2006, 2012; Fletcher et al., 2001]) i BootCat/WebBootCat ([Baroni & Bernardini, 2004; Baroni et al., 2006a,b]). KWiCFinder se više ne održava[7], tako da će u nastavku biti reči samo o programu BootCat i veb-servisu WebBootCat[8].

BootCat

Licenca BootCat ([Baroni & Bernardini, 2004]) predstavlja skup programskih skripti napisanih u programskom jeziku Perl, te za njihovo kopiranje i distribuciju važe istim uslovi kao i za Perl, tj. u pitanju je besplatan softver otvorenog koda. Pored korišćenja Perl-skripti iz komandne linije, postoji i odgovarajuće grafičko sučelje (BootCat front-end), dostupno kao slobodan softver pod uslovima GNU GPL (verzija 3 i eventualne nove verzije). Autori originalnih skripti su Marko Baroni (Marco Baroni) i Silvija Bernardini (Silvia Bernardini), dok su značajan doprinos konačnoj verziji dali Eros Zanchetta, Nikola Ljubešić i Cyrus Shaoul.

Platforma BootCat se može koristiti na svakoj platformi koja poseduje interpretator Perl-skripti (Windows, Linux, itd.).

Klijent-server arhitektura i veb BootCat se ponaša kao veb klijent, tj. korisnik preko njega upućuje zahtev za određenim brojem stranica na vebu koje sadrže listu zadatih ključnih reči ili njihovih kombinacija (eng. *seeds*). Na osnovu liste formiraju se sekvence (n-torke) reči[9], u formi upita se prosleđuju specificiranom

[6] `http://www.gnu.org/software/wget`

[7] Zvaničan sajt `http://kwicfinder.com/KWiCFinder.html` je još uvek aktivan.

[8] Navedeni podaci o programu BootCat su prikupljeni tokom juna 2013. godine.

[9] S obzirom da se na osnovu liste od m elemenata može generisati m^n sekvenci dužine n, korisniku se omogućava da zada maksimalnu dužinu sekvence (n), maksimalan broj generisanih sekvenci, kao i da neposredno ukloni neželjene generisane sekvence.

pretraživaču na vebu (Bing, Google Search, Yahoo, itd.) i na osnovu dobijenih referenci koje proizvodi pretraživač[10] BootCat preuzima sa veba odgovarajuće stranice, obrađuje ih (uklanja HTML-anotaciju i netekstuelne elemente) i kreira od njih lokalni veb-korpus.

Ažurnost i podrška Aktuelna verzija programa BootCat je 0.61 i objavljena je jula 2012. godine. Dokumentacija je dostupna na adresi `http://docs.sslmit.unibo.it/doku.php?id=bootcat:start`.

Proširivost S obzirom da je dostupan izvorni kod programa na programskom jeziku Perl, korisnicima je omogućeno da prošire funkcionalnost softvera pomoću svojih Perl-skripti.

Prethodna obrada BootCat omogućava obradu preuzetih stranica sa veba u smislu eliminisanja HTML-anotacije i netekstuelnih HTML-elemenata (korisnik može da specificira listu HTML-elemenata čiji se tekstuelni sadržaj uključuje u rezultujući korpus).

Jezički resursi BootCat koristi isključivo veb kao jezički resurs.

Tipovi pretrage i raspoložive statističke funkcije BootCat ne poseduje sopstvene module za pretragu i analizu korpusa, već se u tu svrhu moraju koristiti drugi alati.

4.3 Sistemi integrisanih korpusnih alata

U nastavku su (po abecednom redu) obrađeni sledeći sistemi integrisanih korpusnih alata:[11]

- AntConc,
- IMS (O)CWB,

[10] Korisnik može da ograniči broj rezultata po svakoj sekvenci reči koja se u formi upita šalje pretraživaču, kao i da ograniči pretragu na određene internet domene bilo eksplicitnim navođenjem željenog domena, bilo navođenjem neželjenih domena.

[11] Podaci prikupljeni o navedenim korpusnim alatima su poslednji put ažurirani krajem jula i početkom avgusta 2013. godine.

- Monoconc, Paraconc i Collocate,
- NooJ i Unitex,
- SketchEngine i NoSketchEngine (Manatee i Bonito),
- Xaira i
- WordSmith Tools.

AntConc

Zvaničan sajt `http://www.antlab.sci.waseda.ac.jp/antconc_index.html`

Licenca AntConc ([Anthony, 2005]) je besplatan vlasnički softver i nije otvorenog koda. Nosilac autorskih prava je Lorens Entoni (Laurence Anthony).

Platforma AntConc podržava platforme: Windows, Macintosh OS X i Linux.

Klijent-server arhitektura i veb AntConc je **stona aplikacija** (eng. **desktop application**) i nema nikakvu podršku za prikupljanje tekstova sa veba, odnosno za kreiranje veb-korpusa.

Ažurnost i podrška Aktuelna stabilna verzija programa je 3.2.4, a autor trenutno radi na beta verziji 3.3.5. Na zvaničnom sajtu softvera[12] se može naći prateća dokumentacija, kao i video-priručnici pomoću kojih autor demonstrira i objašnjava raspoložive mogućnosti programa.

Proširivost Korisnici imaju mogućnost da promene podešavanja programa koja se odnose na:

- tokenizaciju — da li će se pod tokenom podrazumevati samo alfabetske niske, samo brojevi, samo interpunkcijski znaci ili neka kombinacija navedenih klasa tokena;

[12] `http://www.antlab.sci.waseda.ac.jp/antconc_index.html`

- pretragu — pored uobičajenih parametara pretrage (širina konteksta, broj rezultata, itd.), moguće je definisati sopstvene metakaraktere umesto ponuđenih (`*` kao oznaka za nula ili više karaktera, `?` kao oznaka proizvoljan karakter, `#` kao oznaka za proizvoljan token, itd.)

Prethodna obrada AntConc ne raspolaže nijednim alatom za prethodnu obradu teksta, a sve operacije obavlja neposredno nad originalnim tekstom, tj. tekst korpusa se ne indeksira. Ovakav pristup je pogodan ukoliko korisnik tokom pretrage korpusa često vrši izmene u tekstu, jer nema potrebe da se tekst ponovo indeksira ili posebno obradi da bi mogao da se pretražuje. S druge strane, zbog odsustva indeksiranja, AntConc bolje rezultate postiže na korpusima malih dimenzija.

Jezički resursi AntConc ne koristi nikakve dodatne jezičke resurse za većinu operacija koje podražava. Izuzetak su statističke funkcionalnosti koje zahtevaju dodatni referentni korpus za upoređivanje relativnih učestanosti korpusnih reči u analiziranom i referentnom korpusu.

Tipovi pretrage AntConc podržava jednostavnu i naprednu pretragu korpusa. Jednostavna pretraga, pored uobičajenih mogućnosti najjednostavnijih programa za uređivanje teksta, koristi i „džoker-znakove" kojima se prepoznaju proizvoljni karakteri ili niske karaktera, kao i tokeni. Napredna pretraga se oslanja na korišćenje regularnih izraza u POSIX-notaciji.

U slučaju da korpus koristi ugrađenu anotaciju, AntConc omogućava da se određeni tipovi anotacije ignorišu prilikom pretrage. To se posebno odnosi na XML-anotaciju i horizontalnu anotaciju (v. odeljak 2.4). U slučaju XML-anotacije, osim samih etiketa, moguće je eliminisati i kompletan XML-element (dakle, i sadržaj između etiketa), što je pogodno za ignorisanje metapodataka prilikom pretrage. U tom smislu, AntConc nudi, kao posebnu pogodnost, da se ignoriše XML-element `teiHeader` koji se u TEI-anotiranim tekstovima koristi za navođenje odgovarajućih metapodataka.

Raspoložive statističke funkcije AntConc podržava generisanje listi učestanosti, n-grama, liste ključnih reči i kolokacionu analizu. U slučaju generisanja liste ključnih reči neophodan je dodatni referentni korpus kako bi se uporedile relativne frekvencije korpusnih reči u analiziranom i referentnom korpusu.

IMS Open Corpus Workbench (IMS OCWB)

Zvaničan sajt `http://cwb.sourceforge.net`

Licenca IMS Open Corpus Workbench ili skraćeno IMS OCWB ([Evert & Hardie, 2011]) je besplatan softver otvorenog koda, dostupan u skladu sa uslovima GNU GPL, verzija 3.0. Prvobitna verzija programa je nastala na **Institutu za (automatsku) obradu prirodnih jezika** (nem. **Institut für Maschinelle Sprachverarbeitung**, skr. **IMS**) u Štutgartu početkom devedesetih godina XX veka ([Christ, 1994a]) po kome je program dobio naziv IMS CWB. Prve verzije programa su distribuirane besplatno, ali njihov izvorni kôd nije bio dostupan. Prelaskom na licencu GNU GPL program menja naziv u sadašnji — IMS Open Corpus Workbench ili IMS OCWB. Softver trenutno održavaju Štefan Evert (Stefan Evert) i Endru Hardi(Andrew Hardie).

Platforma IMS CWB je prvobitno napisan za platforme koje koriste derivate operativnog sistema Unix, pre svega Linux. Kada je u pitanju IMS OCWB, dostupne su verzije programa za Linux, Windows[13], Mac OS X i Solaris.

Klijent-server arhitektura i veb IMS OCWB ne poseduje alate za preuzimanje i obradu tekstova sa veba. S druge strane, kad je u pitanju pretraga ogromnih korpusa na vebu, IMS OCWB se koristi kao konkordancer u pozadini, tj. korisnici preko veb sučelja prosleđuju konkordanceru svoje upite. Pri tome je većina administratora korpusa razvijala različita veb sučelja prilagođena arhitekturi korpusa. Počev od 2011. godine autori softvera (posebno Endru Hardi) razvijaju sopstveno veb sučelje — CQPWeb, koje treba da omogući rad sa proizvoljnim korpusom kreiranim pomoću IMS OCWB-alata ([Hardie,

[13]Do jula 2013. godine, verzija programa IMS OCWB za platformu Windows je još uvek u fazi testiranja, tj. još uvek nije proizvedena stabilna verzija programa.

n.d.]). CQPWeb je implementiran pod uticajem veb sučelja BNCweb ([Hoffmann et al., 2008]) koje koriste korisnici Britanskog nacionalnog korpusa i koje takođe u pozadini koristi IMS OCWB-alate.

Ažurnost i podrška Trenutna stabilna verzija programa je 3.0.0. S obzirom da je u pitanju program otvorenog koda, korisnicima je u svakom trenutku dostupna i razvojna verzija softvera (beta verzija).

Dokumentacija je još uvek nedovršena[14]. Trenutno su na zvaničnoj prezentaciji dokumentacije softvera[15] dostupna dva nekompletna priručnika:

- administratorski priručnik ([Evert & The OCWB Development Team, 2010a]) koji opisuje kreiranje korpusa u formatu koji zahteva IMS OCWB;
- korisnički priručnik ([Evert & The OCWB Development Team, 2010b]) koji opisuje upitni jezik za pretragu i analizu korpusa.

Endru Hardi je pripremio i video uputstvo o korišćenju veb sučelja CQPWeb[16].

Kao zamena za nekompletnu dokumentaciju, postoji dopisna lista[17] preko koje članovi zajednice korisnika komuniciraju sa autorima programa i međusobno, razmenjuju iskustva u korišćenju programa, prijavljuju softverske greške i predlažu nove funkcionalnosti programa.

Proširivost Sav izvorni kod programa (pre svega, na programskom jeziku C/C++ i Perl) i pratećih alata (C/C++, Perl, PHP, Python, Java) je dostupan, tako da korisnici mogu da prošire funkcionalnost programa u saradnji sa autorima.

Primer jednog značajnog proširenja je grafičko okruženje TXM ([Heiden, 2010]), razvijeno u okviru projekta Textométrie[18], koje u sebi sadrži programski kôd IMS OCWB-a i predstavlja GUI za pretragu i analizu korpusa kreiranih pomoću IMS OCWB-alata. TXM, kao i IMS OCWB, je licenciran u skladu sa GNU

[14] U trenutku pisanja (jun 2013. godine) još uvek nije kompletiran deo dokumentacije koji se odnosi na pripremu i pretragu paralel(izova)nih korpusa, kao ni preciznije objašnjenje sintakse pojedinih komandi za pretragu jednojezičnih korpusa.

[15] `http://cwb.sourceforge.net/documentation.php`

[16] `http://www.youtube.com/watch?v=Yf1KxLOI8z8`

[17] `http://devel.sslmit.unibo.it/mailman/listinfo/cwb`

[18] `http://textometrie.ens-lyon.fr/?lang=en.`

GPL, verzija 3.0. Upravo je saradnja između autora TXM-a i IMS OCWB-a omogućila razvoj tekuće verzije IMS OCWB-a za platformu Windows.

Prethodna obrada IMS OCWB zahteva da ulazni tekstovi korpusa budu u vertikalnom formatu (v. odeljak 2.4), ali ne obezbeđuje alat za konverziju teksta u taj format. Ulazni tekstovi u vertikalnom formatu se indeksiraju na način opisan u odeljku 2.5, tako da svaka izmena u ulaznim tekstovima zahteva brisanje poslednje verzije korpusa, ponovljenu konverziju ulaznih tekstova u vertikalni format i njihovo indeksiranje.

Jezički resursi IMS CWB koristi isključivo korpuse kreirane pomoću njegovih alata. Informacije iz drugih jezičkih resursa moraju da se integrišu u tekstove korpusa kao skup pozicionih atributa i indeksiraju (v. odeljak 2.5).

Tipovi pretrage Pretraga korpusa se zasniva na korišćenju upitnog jezika CQL (detaljno obrađen u poglavlju 7), zasnovanog na regularnim izrazima u POSIX-notaciji. CQL je postao *de facto* standard kad su u pitanju upitni jezici savremenih konkordancera, tj. usvojen je od strane drugih konkordancera ili bez izmena ([Kilgarriff et al., 2004; Rychlý, 2007]), ili sa proširenjima ([Jakubiček et al., 2010], [Przepiórkowski, 2004]).

Raspoložive statističke funkcije IMS OCWB omogućava generisanje listi učestanosti i n-grama. CQPweb omogućava kolokacionu analizu konkordanci dobijenih kao rezultat korisnikovog upita.

MonoConc, ParaConc i Collocate

Zvaničan sajt Monoconc: `http://www.monoconc.com`

Paraconc: `http://www.paraconc.com`

Collocate: `http://www.collocationary.com/`

Licenca MonoConc ([Barlow, 2012]), ParaConc ([Barlow, 2008]) i Collocate ([Barlow, 2004]) predstavljaju vlasnički i komercijalni softver. Autor Majkl Barlou (Michael Barlow) je kreirao dva konkordancera, MonoConc — za rad sa

jednojezičnim korpusima — i ParaConc, namenjen radu sa paralel(izova)nim korpusima. Iako oba programa MonoConc i ParaConc sadrže opcije za statistička izračunavanja, razvijen je i treći alat, Collocate, koji proširuje njihove mogućnosti. MonoConc i ParaConc se mogu testirati pre kupovine, ali su na raspolaganju samo demonstrativne verzije starih[19] verzija programa koje, za razliku od aktuelnih, ne koriste Unicode već osmobitne kodne rasporede[20].

Platforma MonoConc i ParaConc su napisani isključivo za platformu Windows, ali se pod određenim uslovima mogu koristiti i na drugim platformama (v. odeljak Platforma za alat WordSmith Tools, str. 226).

Klijent-server arhitektura i veb Nijedan od programa MonoConc, ParaConc i Collocate ne koristi klijent-server arhitekturu, nema veb-sučelje, niti mogućnost za preuzimanje tekstova sa veba i kreiranje veb korpusa.

Ažurnost i podrška Iako se MonoConc i ParaConc redovno održavaju, to nije slučaj i sa njihovim demonstrativnim verzijama i besplatnom zvaničnom dokumentacijom. Kompletne komercijalne verzije programa se isporučuju sa dokumentacijom koja tokom pisanja nije bila dostupna, pa se o njoj ništa ne može reći. Što se tiče besplatnih priručnika, videti, na primer, [Barlow, 2012] za MonoConc, odnosno [Barlow, 2003] za ParaConc.

Proširivost S obzirom da softver nije otvorenog koda i ne omogućava dodavanje modula koje definišu korisnici, jedino autor ima privilegiju da proširi softver novim funkcionalnostima.

Prethodna obrada MonoConc omogućava kreiranje korpusa tako što korisnik iterativno učitava niz tekstualnih datoteka, pri čemu se tokom jedne iteracije može učitati više datoteka. Pre učitavanja ulaznih datoteka korisik može da izabere jezik korpusa iz ponuđene liste jezika, i na taj način zapravo izabere kodni raspored koji će program koristiti za prikaz teksta korpusa, konkordanci, listi učestanosti i kolokacija.

[19] Demonstrativna verzija ParaConc-a je stara skoro deset godina.

[20] Srpski jezik nije zastupljen u listi dostupnih jezika demonstrativne verzije programa MonoConc, ali se umesto njega može koristiti hrvatski koji zahteva da tekstovi koriste kodni raspored CP-1250.

Jezički resursi Nijedan od programa MonoConc, ParaConc i Collocate ne koristi dodatne jezičke resurse sem korpusa koje obrađuju.

Tipovi pretrage U okviru programa MonoConc postoje tri tipa pretrage: jednostavna pretraga teksta pomoću „džoker-znakova", pretraga pomoću POSIX-regularnih izraza i pretraga sa korišćenjem anotacije. Ukoliko tekstovi korpusa koriste ugrađenu anotaciju, rezultati pretraga pomoću „džoker-znakova" i POSIX-regularnih izraza neće praviti razliku između samog teksta korpusa i anotacije. Pretraga sa korišćenjem anotacije je dozvoljena ukoliko korisnik prethodno specifikuje u odgovarajućim opcijama programa (Tag Settings) na koji način se prepoznaju elementi strukturne anotacije (pasusi, rečenice, itd.) i elementi morfološke anotacije (vrsta reči, lema, itd.)[21] kako bi se tokom pretrage razlikovali od teksta. Pretraga sa korišćenjem anotacije koristi isključivo „džoker-znakove", tj. POSIX-regularni izrazi nisu na raspolaganju.

Tokom pretrage sa korišćenjem anotacije, kako bi se u upitu napravila razlika između teksta i anotacije, koristi se poseban karakter (podrazumevana vrednost je `&`). Značenje simbola za razlikovanje teksta i anotacije se menja u zavisnosti od njegove pozicije, tj. ako se simbol nalazi iza sekvence karaktera u upitu, onda ta sekvenca predstavlja korpusnu reč, odnosno, ako se nalazi ispred sekvence karaktera u upitu, onda ta sekvenca predstavlja element anotacije[22].

Podrazumevani metakarakteri koje MonoConc koristi kao „džoker-znake" su:

- `*` — ukoliko stoji samostalno, tj. belinama je odvojen od ostatka upita, označava korpusnu reč; u protivnom predstavlja nula ili više karaktera;
- `%` — predstavlja najviše jedan karakter;
- `?` — predstavlja tačno jedan karakter;
- `@` — ukoliko stoji samostalno, tj. belinama je odvojen od ostatka upita, označava interval korpusnih reči (podrazumevana vrednost je od dve do

[21] Na primer, korisnik može da specifikuje da se elementi morfološke anotacije pojavljuju: a) ispred reči kao etikete, na primer `<w N>reka` kao oznaka da je `reka` korpusna reč (`w`) i imenica (`N`), ili b) iza reči, odvojeni podvlakom (`reka_N`).

[22] Na primer, upit `reka& &N` traži korpusnu reč `reka` za kojom sledi imenica (`&N`).

pet korpusnih reči), pri čemu najmanju i najveću dužinu intervala zadaje korisnik.

Pre svake pretrage korisnik može da promeni podrazumevane oznake „džoker-znakova", kao i najmanju i najveću dužinu intervala korpusnih reči (označenog sa `@`), koristeći dijalog Search Options. U istom dijalogu se mogu promeniti podrazumevani simbol za razlikovanje teksta i anotacije (`&`), separatori korpusnih reči, maksimalan broj rezultata, minimalna učestanost pojedinačnog rezultata, širina konteksta. Posebno zanimljive mogućnosti su ignorisanje pojedinih karaktera tokom pretrage (na primer crtice, kako bi se u rezultatu dobile i varijante poput `auto-put` i `autoput`), odnosno izjednačavanje pojedinih karaktera.

Opcije pretrage u ParaConc-u su istovetne kao u MonoConc-u, jedina razlika je što su na raspolaganju dva jednojezična korpusa za pretragu koji se mogu pretraživati i pojedinačno i paralelno. U oba slučaja se prilikom prikaza konkordanci iz teksta na jednom jeziku (izvornom ili ciljnom), prikazuju i odgovarajući segmenti iz teksta na drugom jeziku (ciljnom ili izvornom).

Raspoložive statističke funkcije MonoConc na zahtev generiše broj korpusnih reči, korpusnih tipova, listu učestanosti za učitane tekstove, pri čemu se može podesiti maksimalan broj prikazanih elemenata liste (sortiran po opadajućoj frekvenciji ili alfabetski), minimalna i maksimalna frekvencija koju korpusna reč mora imati da bi bila prikazana u listi, da li se prilikom generisanja liste učestanosti razlikuju velika i mala slova u zapisu korpusne reči, da li se ignoriše anotacija korpusa, itd. Korisnik ima mogućnost da zada i neobaveznu listu stop-reči, tj. reči koje neće biti uzete u obzir prilikom generisanja liste učestanosti, niti tokom kolokacione analize konkordanci.

Kad je u pitanju kolokaciona analiza, korisniku se omogućava da zada veličinu kolokacionog opsega, ali ne i statističku meru značajnosti. Rezultati kolokacione analize se ispisuju sa odgovarajućim konkordancama, odnosno crvenom bojom se označava određen broj (podrazumevana vrednost je pet) najučestalijih kolokata. Prilikom izračunavanja se ne koriste dodatni resursi

u vidu relativnih učestanosti korpusnih reči nekog referentnog korpusa, već se upoređuju apsolutne učestanosti korpusnih reči iz kolokacionog opsega sa njihovom očekivanom apsolutnom učestanošću, izračunatom na osnovu veličine kolokacionog opsega i relativne učestanosti korpusnih reči u celom korpusu.

MonoConc nema mogućnost za generisanje liste n-grama.

ParaConc ima iste mogućnosti u pogledu statističkih izračunavanja kao i MonoConc, pri čemu se one mogu primeniti kako na pojedinačne jednojezične korpuse koji su komponente paralel(izovanog) korpusa, tako i na ceo bitekst.

Kako bi se unapredile mogućnosti statističkih izračunavanja koje imaju MonoConc i ParaConc, razvijen je poseban program Collocate koji ima tri glavne komponente[23].

Prva komponenta Collocate-a na osnovu zadate veličine kolokacionog opsega, statističkog testa (t-test, test zajedničke informacije, test logaritamske verodostojnosti) i tražene ključne reči proizvodi listu odgovarajućih kolokacija sa njihovom učestanošću i vrednošću primenjene statističke mere. Ključne reči mogu sadržati „džoker-znakove". Postoji i opcija da se traže kolokacije za listu ključnih reči.

Druga komponenta Collocate-a omogućava generisanje n-grama.

Trećom komponentom Collocate-a se ekstrahuju kolokacije iz celog korpusa na osnovu zadate veličine kolokacionog opsega.

NooJ i Unitex

Zvaničan sajt NooJ: `http://www.nooj4nlp.net`

Unitex: `http://www-igm.univ-mlv.fr/~unitex`

Licenca Kao što je već rečeno u odeljku 2, pododeljak LADL/DELA, Laboratorija za automatsku dokumentaciju i lingvistiku (LADL) je razvila programski alat INTEX ([Silberztein, 1999]) namenjen obradi korpusa pomoću mor-

[23]Pošto nije dostupna demonstrativna verzija programa Collocate, sve navedene informacije su prenete sa njegovog zvaničnog sajta `http://www.collocationary.com`

fooloških elektronskih rečnika u formatima DELA. Posle razlaza LADL-a i autora INTEX-a, Maksa Silberštajna, LADL je razvio alat Unitex ([Paumier, 2011]) kao kompatibilnu zamenu za INTEX sa podrškom za Unicode, dok je Silberštajn odustao od daljeg razvoja INTEX-a i napravio NooJ ([Silberztein, 2003]), novi alat sa novim formatima morfoloških elektronskih rečnika[24], takođe sa podrškom za Unicode.

NooJ je od početka razvijan kao besplatni softver, ali ne i kao softver otvorenog koda. Tokom projekta CESAR ([Váradi, 2011]) razvijena je verzija NooJ-a u programskom jeziku Java kao softver otvorenog koda licenciran pomoću AGPL[25]. Glavni autor softvera je Maks Silberštajn (Max Silberztein).

Unitex se sve vreme razvija kao besplatan softver otvorenog koda koji se distribuira u skladu sa licencom LGPL, dok se prateći lingvistički resursi mogu koristiti uz poštovanje uslova licence LGPLLR[26]. Vodeći autor Unitex-a je Sebastijan Pomije (Sébastien Paumier).

Platforma NooJ je .NET-aplikacija, najpre implementirana u programskom jeziku C# i isključivo za platformu Windows. Prilagođavanjem izvornog koda napravljena je verzija NooJ-a za okruženje MONO koje omogućava izvršavanje .NET-aplikacija na drugim platformama (Linux, FreeBSD, Mac OS X, Solaris). Tokom projekta CESAR razvijena je verzija NooJ-a u programskom jeziku Java, nezavisna od platforme.

Unitex je implementiran korišćenjem programskih jezika C++ i Java, pa je samo delimično nezavisan od platforme. Međutim, naporom autora, izvorni kôd programa se može kompilirati na svakoj od sledećih platformi: Linux, Windows, Mac OS X.

[24] Formati rečnika koje koristi NooJ su nekompatibilni sa formatom DELA.

[25] AGPL ili Affero GPL je licenca kompatibilna sa GNU GPL 3.0, ali ne i sa GNU GPL 2.0. Kreirana je sa ciljem da se spreči propust koji nastaje kada se softver otvorenog koda licenciran pomoću GNU GPL koristi u okviru veb servisa. S obzirom da se veb servisi ne distribuiraju već samo koriste, GNU GPL omogućava autoru veb servisa da svoj softver ne licencira pomoću GNU GPL iako sadrži tuđi softver licenciran pomoću GNU GPL. AGPL se razlikuje od GNU GPL samo po tome što primorava autora koji u svojoj aplikaciji koristi tuđi softver licenciran kao AGPL, da i svoj softver licencira na isti način bez obzira da li ga distribuira ili ne.

[26] `http://infolingu.univ-mlv.fr/DonneesLinguistiques/Lexiques-Grammaires/lgpllr.html`

Klijent-server arhitektura i veb Ni NooJ ni Unitex ne koriste klijent-server arhitekturu, niti omogućavaju pristup tekstovima na vebu, već su u pitanju stone aplikacije.

Ažurnost i podrška I NooJ i Unitex se redovno održavaju. Aktuelna verzija Unitex-a je 3.0, ali korisnici mogu da preuzmu i razvojnu (beta) verziju 3.1. Na zvaničnom sajtu je dostupan i detaljan priručnik ([Paumier, 2011]) koji pokriva sve aspekte korišćenja programa. U slučaju originalne implementacije NooJ-a (C#) se ne može govoriti o stabilnoj verziji jer se program ažurira svakodnevno i sa zvaničnog sajta se uvek može preuzeti isključivo poslednja verzija programa. Na zvaničnom sajtu NooJ-a je dostupan i priručnik ([Silberztein, 2003]).

I NooJ i Unitex imaju svoje organizovane zajednice korisnika. Korisnici NooJ-a komuniciraju preko foruma na adresi `http://groups.yahoo.com/group/nooj-info`, dok su korisnici Unitex-a u kontaktu sa autorom softvera preko e-adrese `unitex@univ-mlv.fr`.

Proširivost S obzirom da su u pitanju programi otvorenog koda, korisnici mogu da dodaju nove funkcionalnosti[27].

Jezički resursi NooJ i Unitex koriste leksičke resurse u obradi korpusa, pre svega elektronske morfološke rečnike (v. str. 145 za Unitex, odnosno [Utvić, 2008]67–75 za NooJ), kao i lokalne gramatike (v. str. 112) u formi grafova. Iako se formati koje Unitex i NooJ koriste za predstavljanje leksičkih resursa međusobno razlikuju[28], interno se svi leksički resursi kompiliraju u konačne automate i

[27]Kad su u pitanju proširenja funkcionalnosti za Unitex, vredi spomenuti doprinose Sedrika Ferona (Cédrick Fairon), Aljoše Obuljena i Saše Petalinkara. Feron je razvio GlossaNet ([Fairon, 2000]), pretraživač i konkordancer koji koristi Unitex i leksičke resurse za pretragu dnevnih novina na vebu. Aljoša Obuljen je proširio funkcionalnost Unitex-a dodavanjem programa Stats koji ispisuje rezultate kolokacione analize na osnovu indeksne datoteke konkordanci, koristeći z-test ([Paumier, 2011: 274]). Saša Petalinkar je omogućio da se Unitex-u zada više ulaznih datoteka i da prikaz konkordanci uzme u obzir i logičku strukturu teksta, tj. da se uz pojedinačne rezultate prikazuju i podnaslovi odgovarajućih logičkih jedinica teksta, odnosno izvornih datoteka, kao i da se uz liste učestanosti tokena generišu i liste učestanosti lema [Petalinkar, 2011].

[28]Kada su u pitanju rečnici jednočlanih leksema (eng. simple words), njihovi formati su praktično izomorfni. Međutim, pristup i format koji NooJ, odnosno Unitex koriste za polusloženice i višečlane lekseme (eng. compounds) se bitno razlikuje. NooJ odvojeno klasifikuje fleksije jednočlanih i višečlanih leksema, tj. opis flektivnih klasa višečlanih leksema ne koristi postojeće opise flektivnih

konačne transduktore koji se koriste prilikom prethodne obrade i pretrage teksta. Resursi su dostupni za 23 jezika (NooJ), odnosno 15 jezika (Unitex).

Prethodna obrada I NooJ i Unitex sadrže svoj modul za prethodnu obradu teksta, pri čemu originalni ulazni tekst ostaje neizmenjen, već se sva obrada odnosi na njegovu kopiju. U oba slučaja prethodna obrada se sastoji iz tokenizacije i segmentacije teksta, pri čemu Unitex segmentira tekst isključivo na rečenice (v. str. 171), dok NooJ nudi korisniku da definiše separator segmenata. U fazi prethodne obrade teksta omogućena je i primena leksičkih resursa (elektronskih morfoloških rečnika i rečničkih transduktora) na tekst, pri čemu se kao rezultat generiše morfološki rečnik teksta kao presek svih raspoloživih leksičkih resursa i samog teksta. Morfološki rečnik teksta se potom može koristiti u pretrazi i modifikaciji (kopije) teksta.

NooJ omogućava kreiranje, obradu i pretragu korpusa učitavanjem više ulaznih datoteka, dok je za Unitex korpus — jedna datoteka[29].

Tipovi pretrage NooJ i Unitex koriste lokalne gramatike (v. str. 112) kao formu zadavanja upita. NooJ i Unitex-lokalne gramatike su predstavljene ili u formi pravila (NooJ-regularni izrazi i Unitex-regularni izrazi) ili u formi grafova (NooJ-gramatike i Unitex-grafovi).

Regularni izrazi koje koriste NooJ i Unitex za pretraživanje su zasnovani na osnovnim regularnim operacijama (unija, dopisivanje, Klinijevo zatvorenje) pri čemu se prvobitno podrazumevalo da se te operacije ne primenjuju na karaktere već na tokene, odnosno korpusne reči, ali su oba programa u međuvremenu omogućili morfološke filtere[30] u kojima se regularne operacije primenjuju na

klasa jednočlanih leksema. Unitex, s druge strane, u slučaju da se višečlana leksema sastoji iz dve ili više jednočlanih leksema čija je fleksija već opisana, opisuje samo kako se fleksije tih članova međusobno slažu, koristeći formalizam MULTIFLEX ([Savary, 2005]), zasnovan na unifikaciji, tj. opisuje vrednosti kojih morfoloških kategorija članova moraju biti istovetne i koje su, prema tome, vrednosti morfoloških kategorija same višečlane lekseme.

[29]Pomenuto proširenje Saše Petalinkara ([Petalinkar, 2011]) još uvek nije uvršćeno u zvaničnu verziju Unitex-a.

[30]Za morfološke filtere NooJ koristi regularne izraze programskog jezika Perl, a Unitex POSIX-regularne izraze.

karaktere tokena (Tabela 4.2[31]). Ono što je specifično za NooJ i Unitex u odnosu na ostale implementacije regularnih izraza jeste korišćenje leksičkih informacija prilikom zadavanja upita (lema, vrsta reči, semantički markeri, morfološke kategorije).

Tabela 4.2: Uporedni prikaz primera regularnih izraza koje u pretrazi koriste NooJ i Unitex.

NooJ	**Unitex**	**značenje**
|	+	unija (alternacija)
␣	␣ ili . ili prazna niska	dopisivanje
*	*	Klinijevo zatvorenje
<E>	<E>	prazna niska
<WF>	<MOT>	proizvoljna korpusna reč
<hteti>	<hteti>	svi flektivni oblici leme hteti
<A+Col>	<A+Col>	svi flektivni oblici prideva (A) koji su označeni semantičkim markerom +Col (*boja*)
<N+Hum+NProp>	<N+Hum+NProp>	svi flektivni oblici imenica (N) koje su označene semantičkim markerima +Hum (*ljudsko biće*) i +NProp (*vlastita imenica*)
<N+Hum-NProp>	<N+Hum~NProp>	svi flektivni oblici imenica (N) koje su označene semantičkom markerom +Hum (*ljudsko biće*) i nisu vlastite imenice (+NProp)
<N+m+4+s>	<N:m4s>	svi flektivni oblici imenica (N) muškog roda (m) u akuzativu (4) jednine (s)
<N+NProp+m+4+s>	<N+NProp:m4s>	svi flektivni oblici vlastitih imenica (N+NProp) u akuzativu (4) jednine (s)
	<kosi,kositi.V>	sva pojavljivanja korpusne reči kosi ukoliko je anotirana u tekstu kao oblik glagola (V) kositi
<kosi,V>		svi flektivni oblici leme čiji je jedan flektivni oblik kosi i koja predstavlja glagol
<A+MP=''ski$''>	<A><<ski$>>	korpusne reči koje predstavljaju oblike prideva (A) i završavaju se niskom karaktera ski (primer morfološkog filtera)

Unitex-grafovi su već detaljno objašnjeni (v. pododeljak 3.1, str. 171), a sličan koncept koristi i NooJ. Oba programa koriste više vrsta grafova: za pret-

[31]Oznake vrsta reči, semantičkih markera i flektivnih kategorija su preuzeti iz elektronskih morfoloških rečnika za srpski jezik u formatu NooJ ([Gucul-Milojević et al., 2008; Stanković et al., 2011]) i DELA ([Krstev, 2008; Krstev & Vitas, 2005; Vitas et al., 2003]).

hodnu obradu ulaznog teksta (samo Unitex), automatsko generisanje rečnika flektivnih i derivacionih oblika na osnovu rečnika lema, pretragu, anotaciju teksta (samo Unitex), razrešavanje višeznačnosti, itd[32]. Grafovi za pretragu i anotaciju se u terminologiji oba programa nazivaju **sintaksičkim grafovima** (eng. **syntactic graphs**). Svaki sintaksički graf se čuva u zasebnoj datoteci određenog tipa (`.grf` u slučaju Unitex-a, odnosno `.nog` za NooJ-sintaksičke grafove). Ime datoteke u kojoj se čuva graf (bez tipa `.grf`, odnosno `.nog`) predstavlja naziv grafa.

Sintaksički grafovi u svojim čvorovima koriste regularne izraze odgovarajućeg programa. Prednost sintaksičkih grafova u odnosu na ekvivalentne regularne izraze je mogućnost da se u čvoru grafa A nalazi referenca na neki postojeći graf B, što je ekvivalentno uključivanju grafa B kao podgrafa grafa A. Time se omogućava da se graf koji se često koristi kao podgraf drugih, složenijih grafova, definiše i ažurira samo na jednom mestu, a koristi po potrebi jednostavnim referisanjem na njegov naziv u čvoru složenijeg grafa.

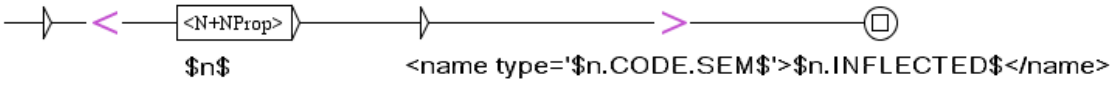
U sintaksičkim grafovima se mogu definisati promenljive koje „pamte" delove prepoznatog niza tokena (ulazne promenljive), kao i delove niski koje graf prilikom prepoznavanja generiše (izlazne promenljive). Sintaksički graf koristi vrednosti promenljivih i pridružene leksičke informacije (lemu, vrstu reči, semantičke markere, vrednosti morfoloških kategorija) za:

- anotaciju teksta (Slika 4.1),
- za konstruisanje uslova ograničenja koji se nameću tokom dalje pretrage u okviru istog grafa ili njegovih podgrafova (Slika 4.2).

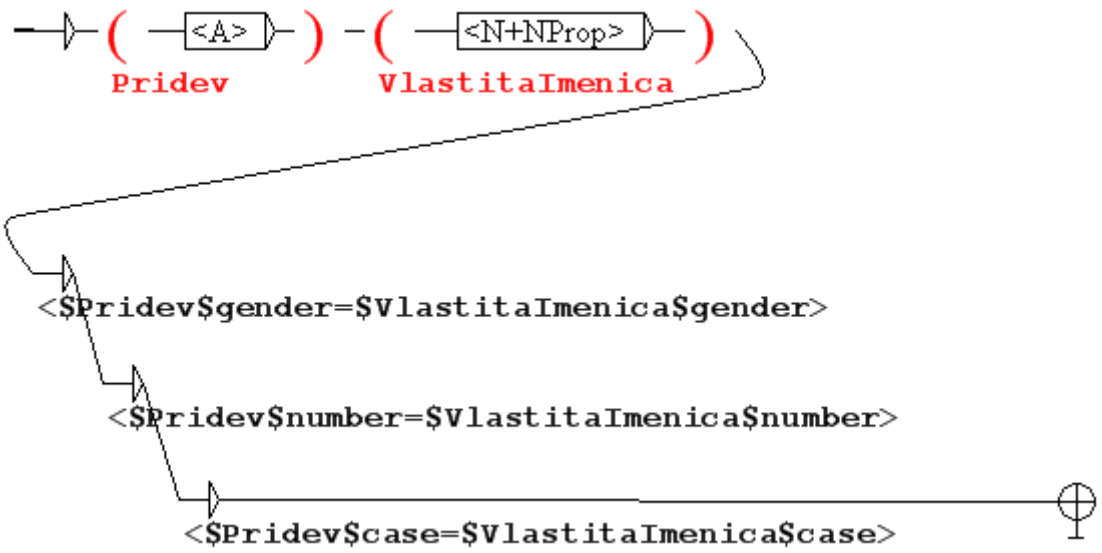
Raspoložive statističke funkcije NooJ omogućava kreiranje liste učestanosti za tokene i bigrame, a za generisane konkordance računa standardnu devijaciju broja konkordanci za svaki tekst korpusa.

Unitex omogućava generisanje liste učestanosti, kao i kolokacionu analizu zasnovanu na z-testu.

[32]NooJ ne podržava modifikaciju ulaznog teksta primenom lokalnih gramatika, već samo generisanih konkordanci.



Slika 4.1: Primer sintaksičkog grafa kojim Unitex anotira vlastite imenice (`N+NProp`) u tekstu, koristeći svoj rečnik vlastitih imena.



Slika 4.2: Primer sintaksičkog grafa kojim NooJ pronalazi sve parove korpusnih reči takve da je prva korpusna reč (promenljiva `$pridev`) pridev (`A`), a druga (promenljiva `$VlastitaImenica`) vlastita imenica (`N+NProp`), pri čemu se te dve korpusne reči slažu u rodu (promenljiva `$gender`), broju (promenljiva `$number`) i padežu (promenljiva `$gender`).

SketchEngine i NoSketchEngine (Manatee i Bonito)

Zvaničan sajt SketchEngine:`http://www.sketchengine.co.uk`

NoSketchEngine[33]: `http://nlp.fi.muni.cz/trac/noske`

Licenca Tokom rada na svojoj doktorskoj disertaciji ([Rychlý, 2000]), Pavel Rihli (Pavel Rychlý) je razvio dva besplatna alata otvorenog koda, Manatee i Bonito ([Rychlý, 2007]). Manatee je **upravljač korpusima** (eng. **corpus manager**), tj. indeksira tekstove u vertikalnom formatu, kreira korpuse i omogućuje rad

[33] Adresa starog sajta koji je još uvek dostupan je `http://www.textforge.cz/products`

sa njima u svojstvu serverske aplikacije. Bonito je klijentska aplikacija, namenjena korisnicima korpusa za potrebe pretrage i analize, koja korisničke upite prosleđuje serveru (Manatee). Manatee i Bonito su razvijeni pod snažnim uticajem srodnog programa IMS (O)CWB (v. str. 209).

U saradnji sa Adam Kilgerifom (Adam Kilgarriff), Rihli je proširio funkcionalnosti klijent-server aplikacije Manatee/Bonito i razvio komercijalni vlasnički softver — SketchEngine ([Kilgarriff et al., 2004]). SketchEngine je dobio naziv po **skici reči** (eng. **word sketch**), „automatski generisanoj stranici koja na osnovu korpusa rezimira gramatičko i kolokacijsko ponašanje reči" ([Kilgarriff et al., 2004: 1]). SketchEngine je u vlasništvu privatne kompanije Lexical Computing Ltd. čiji je osnivač Adam Kilgerif.

Manatee i Bonito su i dalje besplatni alati otvorenog koda koji su sada dostupni pod nazivom NoSketchEngine i pod uslovima licence GPL (verzija 2). NoSketchEngine je u vlasništvu Centra za obradu prirodnog jezika Fakulteta za informatiku Masarikovog univerziteta u Brnu[34].

Klijent-server arhitektura i veb Kao što je već spomenuto u pododeljku Licenca, i SketchEngine i NoSketchEngine poseduju klijent-server arhitekturu, pri čemu Manatee i Bonito predstavljaju odgovarajući serverski i klijentski program tim redom. SketchEngine i NoSketchEngine koriste Bonito na dva načina: (1) kao samostalnu klijentsku aplikaciju na programskom jeziku Tcl/Tk[35] i (2) kao veb sučelje generisano na programskom jeziku Python. Aktuelne verzije koriste upravo veb sučelje koje se i dalje razvija.

SketchEngine podržava i kreiranje korpusa preuzimanjem tekstova sa veba, za šta koristi alat WebBootCat ([Baroni et al., 2006a,b]).

Ažurnost i podrška Zvaničan sajt programa SketchEngine sadrži detaljnu dokumentaciju[36] koja se može dobrim delom koristiti i za NoSketchEngine. Takođe

[34] `http://www.muni.cz`
[35] U pitanju je verzija Bonito1 koja se više ne razvija niti održava.
[36] http://trac.sketchengine.co.uk/wiki/WikiStart

postoji poseban video kanal sa uputstvima za korišćenje pojedinih mogućnosti programa SketchEngine[37].

Manatee i Bonito se održavaju nezavisno kao delovi programa SketchEngine i NoSketchEngine. Aktuelna verzija serverskog programa Manatee je 2.59.1, a klijentskog programa Bonito2 je 2.91.13. Aktuelna stabilna verzija NoSketchEngine-a je objavljena krajem oktobra 2012. godine. NoSketchEngine na svom zvaničnom sajtu poseduje vrlo šturu dokumentaciju koja se pre svega odnosi na proces instalacije serverske i klijentske aplikacije, dok se za ostale informacije korisnici upućuju na dokumentaciju SketchEngine-a.

Proširivost S obzirom da su Manatee i Bonito besplatni programi otvorenog koda, korisnici mogu da proširuju njihovu funkcionalnost, dok za SketchEngine takvu mogućnost imaju samo autori.

Prethodna obrada Kao što je već spomenutu u pododeljku Licenca, IMS (O)CWB (v. str. 209) je snažno uticao na razvoj programa Manatee i Bonito, što se između ostalog ogleda u istovetnom formatu (format vertikalnog teksta) koji se zahteva od ulaznih tekstova korpusa. Ulazni tekstovi korpusa se indeksiraju pomoću biblioteke FINLIB ([Rychlý, 2000]) da bi se kreirao korpus kome pristupa klijent Bonito.

SketchEngine sadrži alat Corpus Architect koji omogućava kompilaciju korpusa na osnovu tekstuelnih dokumenata u različitim formatima (TXT, PDF, PS, DOC, HTML, VERT).

Jezički resursi Od dodatnih jezičkih resursa SketchEngine koristi automatski generisane kolokacijske rečnike i to za engleski, mađarski, češki, bugarski, hrvatski, francuski, poljski, srpski, slovački, španski i malteški jezik.

Tipovi pretrage Iako postoji nekoliko tipova pretrage korpusa koju omogućava Bonito, interno se one svode na korišćenje upitnog jezika CQL (v. poglavlje 7), preuzetog bez izmena od programa IMS (O)CWB (v. str. 209). Korisnicima su na raspolaganju:

[37] `http://www.youtube.com/user/TheSketchEngine?feature=watch`

- jednostavna pretraga (ne koristi regularne izraze, svaka korpusna reč upita se tretira kao lema i rezultat sadrži sve oblike te leme);
- pretraga po lemama, ako je korpus anotiran tom morfološkom informacijom (koristi regularne izraze, a u slučaju SketchEngine-a postoji dodatna mogućnost filtriranja po vrsti reči);
- pretraga po frazama (ne uzima u obzir eventualnu informaciju o lemi, koristi regularne izraze);
- pretraga po oblicima rečima (ne koristi informaciju o lemi, koriste regularne izraze, u slučaju SketchEngine-a postoji dodatna mogućnost filtriranja po vrsti reči);
- pretraga po karakterima (na osnovu zadate niske karaktera generišu se konkordance čije ključne reči sadrže tu nisku karaktera kao svoju podnisku);
- CQL-pretraga omogućava zadavanje upita korišćenjem upitnog jezika CQL (v. poglavlje 7) i svi prethodno navedeni tipovi pretrage se mogu realizovati pomoću ovog tipa pretrage.

Filtriranje konkordanci omogućavaju dodatne opcije Context[38], Text Types[39], dok SketchEngine omogućava i filtriranje po vrsti reči (ako je odgovarajući korpus anotiran tom informacijom).

Raspoložive statističke funkcije I SketchEngine i NoSketchEngine poseduju opciju Word List koja omogućava formiranje listi učestanosti. Korisnik može u posebnim datotekama da zada tzv. „bele" i „crne" liste reči, tj. liste korpusnih reči koje se moraju, odnosno ne smeju pojaviti u listi učestanosti. Pre generisanja liste učestanosti korisnik zadaje adrese datoteka sa listama reči u formatu „jedna reč u jednoj liniji". Liste učestanosti se mogu filtrirati i pomoću regularnog izraza, kao i preciziranjem minimalne frekvencije koju korpusna reč iz konteksta mora imati da bi se našla u listi. Word List omogućava kreiranje

[38]Na primer, na kom rastojanju od ključne reči moraju da se pojave oblici određenih lema.

[39]Na osnovu metapodataka o pojedinačnim tekstovima korpusa se sužava pretraga na tekstove čiji metapodaci imaju zadate vrednosti, na primer pripadaju određenom funkcionalnom stilu.

listi učestanosti kako za korpusne reči i tokene, tako i za leme (ako korpus sadrži tu informaciju).

Još jedna zajednička opcija SketchEngine-a i NoSketchEngine-a je kolokaciona analiza konkordanci pri čemu korisnik može da izabere jednu ili više statističkih mera ([LCL, 2012]): t-test, z-test, test zajedničke informacije, test logaritamske verodostojnosti, itd.

SketchEngine poseduje dodatne opcije u vidu generisanja skica reči (Word Sketches) i njihovog upoređivanja (Sketch-Diff). Ovaj proces se zapravo svodi na izračunavanje učestanosti uređenih trojki (w_1, R, w_2), gde su w_1 i w_2 korpusne reči, a R određena gramatička relacija koja ih povezuje.

WordSmith Tools

Zvaničan sajt `http://www.lexically.net/wordsmith/index.html`

Licenca WordSmith Tools ([Scott, 2012b]), skr. WordSmith, je vlasnički i komercijalni softver. Njegov nastanak se vezuje za program MicroConcord koji je objavio Oxford University Press 1993. godine[40]. Autori MicroConcord-a, Tim Džons (Tim Johns) i Majk Skot (Mike Scott), su nameravali da naprave jednostavan konkordancer, prevashodno namenjen kao pomoćno sredstvo u nastavi jezika. Uz MicroConcord, korisnicima su, za 50 funti, stavljena na raspolaganje dva korpusa na engleskom jeziku od po milion reči, jedan sastavljen od novinskih, a drugi od akademskih tekstova.

Mike Scott je 1996. godine započeo sa razvojem kolekcije alata za korpusnu lingvistiku, WordSmith Tools, sa idejom da se u svakoj narednoj verziji kolekcija dopunjava novim alatima. Od 1996. do 2012. godine Skot je objavio ukupno šest verzija kolekcije alata (1996., 1997., 1999., 2004., 2007. i 2012. godine). Trenutno WordSmith obuhvata tri glavna alata (Concord, KeyWords, WordList) i deset pomoćnih modula (WebGetter, Text Converter, Aligner, Character Profiler, File Utilities, Minimal Pairs, Data Converter, Corpus Cor-

[40] `http://www.lexically.net/personal_pages/memories%20of%20Tim%20Johns.html`

ruption Finder, File Viewer, WSConcgram). Prvi i jedan od glavnih alata kolekcije je Concord, čija je preteča MicroConcord.

Pored licenci za pojedinačne korisnike, postoje i grupne licence koje omogućavaju da se program instalira kao mrežna aplikacija, tako da svi korisnici iz grupe mogu da rade sa istim korpusima.

Platforma MicroConcord je napisan u programskom jeziku Pascal, a delovi sa kritičnim vremenom izvršavanja u asembleru. WordSmith takođe koristi kombinaciju asemblera i višeg programskog jezika, pri čemu je Pascal zamenjen objektnom verzijom tog jezika, poznatijom kao Delphi. Sve verzije WordSmith-a su napisane isključivo za Windows. Za ostale platforme (Mac OS X, Linux) autor preporučuje korišćenje softvera za virtuelizaciju[41] ili softvera koji omogućava da se na drugom operativnom sistemu (Linux, Mac OS X, FreeBSD i Solaris) izvršavaju Windows-aplikacije bez kopije samog Microsoft Windows-a (na primer Wine[42]).

Klijent-server arhitektura i veb U okviru svoje kolekcije alata WordSmith Tools raspolaže i modulom WebGetter čija je namena da na osnovu korisnikovog upita pretraži veb i formira odgovarajući korpus. Upit se zadaje na način opisan u pododeljku Tipovi pretrage (str. 231). Uz sam upit, korisnik mora da specifikuje:

- kom pretraživaču veba se prosleđuje upit;
- lokaciju na lokalnom računaru gde će biti sačuvani dokumenti preuzeti[43] sa veba (rezultat upita);

[41] Softver za virtuelizaciju omogućava da se na računaru pokrene više programa, virtuelnih mašina, pri čemu svaka virtuelna mašina predstavlja softversku simulaciju jednog računara sa određenom hardverskom konfiguracijom i svojim operativnim sistemom. Na taj način na računaru koji koristi operativni sistem koji nije Microsoft Windows, korisnik može da pokrene virtuelnu mašinu na kojoj je instaliran Microsoft Windows i da na njoj koristi WordSmith. Nedostaci ovog rešenja su višestruki: neophodna je kopija operativnog sistema Microsoft Winodows, a izvršavanje operativnog sistema i aplikacija na virtuelnoj mašini je značajno sporije u odnosu na pravi računar.

[42] http://wiki.winehq.org/

[43] Adrese preuzetih tekstova predstavljaju prvih 1000 rezultata dobijenih od izabranog pretraživača.

- vreme (izraženo u sekundama) posle kog WebGetter odustaje od preuzimanja veb-stranice ukoliko do tada nije bilo odgovora od odgovarajućeg veb-servera;
- minimalnu veličinu svake datoteke koja se preuzima sa veba (izražena u kilobajtima, podrazumevana vrednost 20 kilobajta);
- minimalni broj korpusnih reči[44] koje datoteka preuzeta sa veba mora imati da bi bila zadržana u rezultujućoj kolekciji.

Korisnik eventualno može da zada i listu korpusnih reči od kojih ili bar jedna ili sve (zavisno od korisnikove želja) moraju da se pojave u preuzetim dokumentima sa veba. Takođe, korisnik može da zahteva i uklanjanje HTML-anotacije (etiketa).

Ažurnost i podrška Aktuelna verzija softvera je WordSmith Tools 6.0. Korisnicima koji žele da testiraju program pre eventualne kupovine na raspolaganju je besplatna verzija za demonstraciju (skr. demo) koja može da se koristi bez vremenskih ograničenja. Iako su u demo verziji dostupne sve funkcionalnosti programa, prikaz rezultata svih modula aplikacije je ograničen[45].

Na zvaničnom sajtu programa je dostupna detaljna dokumentacija ([Scott, 2010]). S obzirom da program obiluje alatima i opcijama, za početnika je najbolje da sledi „Korak po korak vodič kroz WordSmith" ([Scott, 2012a]).

Proširivost WordSmith Tools je zamišljen kao skup alata koji će se neprekidno proširivati, ali s obzirom da je u pitanju vlasnički i komercijalni softver čiji izvorni kôd programa nije dostupan, nove alate dodaje isključivo autor Majk Skot. Sledi kratak opis pomoćnih modula WordSmith Tools-a koji nisu detaljnije objašnjeni u pregledu ostalih parametara programa:

[44] Podrazumevani skup separatora, pomoću kog se definišu korpusne reči u programu WordSmith Tools, predstavljaju nealfabetski karakteri koji zavise od izabranog jezika, odnosno njegovog alfabeta. Korisnik može da promeni podešavanja navođenjem znakova interpunkcije ili slova iz drugih jezika za koje dozvoljava da budu deo korpusnih reči.

[45] Na primer, za svaki upit WordSmith Tools prikazuje najviše 25 konkordanci. Međutim, kompletan rezultat upita se može sačuvati u datoteci i pogledati pomoću nekog od programa za uređivanje teksta.

- Aligner je alat koji se koristi za kreiranje paralelnog teksta, preciznije biteksta, na nivou rečenica ili na nivou pasusa. Aligner prikazuje bitekst kao niz jedinica teksta (rečenice ili pasusi) koje se uparuju, pri čemu su u prikazu naizmenično raspoređene jedinice iz izvornog teksta i jedinice iz ciljnog teksta. Osim pregleda biteksta, korisnik ima mogućnost da menja granice jedinica teksta spajanjem susednih jedinica, odnosno podelom postojeće jedinice teksta na dve nove jedinice, pri čemu redosled jedinica određuje njihovo međusobno uparivanje.
- Minimal Pairs je alat za detekciju potencijalnih tipografskih grešaka. Naziv je dobio po tome što pronalazi parove reči u tekstu koje se „minimalno razlikuju". Pod „minimalnom razlikom" se podrazumeva mogućnost da se jedna tekstuelna reč transformiše u drugu primenom minimalnog broja operacija umetanja, brisanja ili zamene karaktera (na primer, `srpski` i `sprski`). Element para sa nižom frekvencijom je kandidat za tipografsku grešku.
- Data Converter vrši konverziju podataka proizvedenih u prethodnim verzijama programa WordSmith Tools u format koji koristi aktuelna verzija.
- Corpus Corruption Finder proverava da li jedan ili više tekstova od kojih se formira korpus odudara po vrednosti određenih parametara od ostalih tekstova korpusa, tj. „po svojoj prirodi ne pripada korpusu" (na primer, nije na istom jeziku kao ostali tekstovi korpusa ili nije tekst uopšte, već iskvarena datoteka koja sadrži slučajni niz karaktera). Alat radi tako što upoređuje „sumnjivi" tekst sa kolekcijom tekstovima koje je korisnik prethodno označio kao „dobre" uzimajući u obzir relativne učestanosti karaktera i (ako korisnik izabere takvu opciju) parova karaktera.

Prethodna obrada Po izboru odgovarajućeg alata, korisnik kreira svoj korpus tako što bira tekstualne datoteke čiji će sadržaj biti deo korpusa. Ulazne datoteke moraju biti u formatu čistog teksta (bez ikakvog formatiranja) koji koristi ili 8-bitni kodni raspored ANSI ili 16-bitni kodni raspored UTF-16 LE. PDF-dokumenti ili dokumenti kreirani pomoću programa Microsoft Word

(*.doc, *.docx) ne mogu se koristiti direktno već se prethodno moraju transformisati u format čistog teksta (*.txt). WordSmith raspolaže svojim alatom Text Converter koji podržava sledeće tipove konverzije:

- konverzije tekstuelnih dokumenata iz formata PDF, Microsoft Word i Excel u format čistog teksta;
- konverzije proizvoljnog kodnog rasporeda u Unicode, tj. UTF-16;
- konverziju znakova za novi red (kraj linije) iz formata Unix u format Windows;
- konverziju znakova za novi red (kraj linije) u razmake[46];
- zamenu znakova navoda (apostrofi i navodnici) i crtica koji nisu ASCII karakteri u odgovarajuće ASCII karaktere, kao i zamenu karaktera koji predstavlja tri tačke u tri karaktera koji predstavljaju po jednu tačku;
- konverziju jednog tipa anotacije u drugi i to:
  - konverziju horizontalnog formata `token_ETIKETA`[47] u format `<ETIKETA>token` ili format `token<ETIKETA>`;
  - međusobne konverzije između formata `<ETIKETA>token` i `token<ETIKETA>`;
  - konverziju vertikalnog formata u format `token<ETIKETA>`[48].
- uklanjanje XML-etiketa;
- konverziju karakterskih entiteta u karaktere na osnovu zadate datoteke sa listom zamena[49];
- zamena svih korpusnih reči njihovim lemama na osnovu zadate datoteke sa listom lema i odgovarajućih oblika (v. pododeljak Jezički resursi).

[46]Oznake kraja pasusa su izuzete iz konverzije, pri čemu se od korisnika očekuje da definiše sekvencu karaktera koja predstavlja oznaku kraja pasusa (podrazumevanu oznaku za kraj pasusa predstavljaju dva uzastopna znaka za novi red).

[47]Konverzija se vrši i ako je umesto podvlake (_) korišćena kosa crta (/) kao separator tokena i etikete, pri čemu korisnik mora to da potvrdi pre same konverzije.

[48]Ako je `<TAB>` oznaka za tabulator koji razdvaja kolone vertikalnog formata, tada se `token<TAB>vrsta_reči<TAB>lema` konvertuje u `token<vrsta_reči><lema>`.

[49]WordSmith se isporučuje sa listom zamena karakterskih entiteta iz kodnog rasporeda ISO 8859-1 u formatu `&quot; 34`, tj. karakterski entitet, razmak, kodna pozicija karaktera u ISO 8859-1.

Svaki od navedenih tipova konverzije se može primeniti na više ulaznih datoteka istovremeno po sistemu *nađi i zameni* (eng. *find and replace*), pri čemu se izmene čuvaju ili u originalnim datotekama ili u njihovim kopijama kreiranim u katalogu koji zadaje korisnik. Specifikacija ulaznih datoteka se svodi na zadavanje adrese kataloga koji ih neposredno sadrži (ulazni katalog), pri čemu korisnik može da filtrira ulazne datoteke po tipu (ekstenziji), kao i da zada konverziju i datoteka koje se nalaze u potkatalozima ulaznog kataloga.

Pored Text Converter-a, WordSmith Tools sadrži niz pomoćnih alata za manipulaciju tekstuelnim datotekama (pregled, pretraga, podela datoteke na delove, spajanje više datoteka u jednu datoteku, pronalaženje duplikata datoteke u kolekciji, poređenje sadržaja dve datoteke, itd.) koji su grupisani u pomoćni modul File Utilities.

Jezički resursi Kada su obavezni jezički resursi u pitanju, modul KeyWords zahteva listu učestanosti dodatnog referentnog korpusa, kako bi se upoređivanjem relativnih učestanosti korpusnih reči u analiziranom tekstu i referentnom korpusu pronašli kandidati za ključne reči teksta. Opciono, moduli Concord i WordList (v. pododeljke Tipovi pretrage i Raspoložive statističke funkcije) mogu koristiti rečnik lema u formatu[50]:

```
;lema ->flektivniOblik, ...flektivniOblik
jesam ->sam, si, je, smo, ste, su
jesam ->jesam, jesi, jeste, jesmo, jesu
kap ->kapi, kapima
kapa ->kape, kapi, kapu, kapo, kapom, kapama
```

Tipovi pretrage Za pretragu korpusa i prikaz konkordanci u okviru programa WordSmith Tools je zadužen alat Concord. Concord ne koristi POSIX-regularne izraze za specifikovanje upita, već kombinuje „džoker-znakove" (`*`, `?`, `^`, `#`)

[50] Simbol `;` predstavlja početak jednolinijskog komentara, zapeta razdvaja oblike, a znak za novi red odrednice rečnika. Ista lema se može pojaviti u više odrednica, i u tom slučaju njoj odgovara unija flektivnih oblika iz svih takvih odrednica.

sa metakarakterima (/) koji imaju sličnu ulogu kao metakarakteri POSIX-regularnih izraza (|), ali se drugačije označavaju (Tabela 4.3).

Izražajna moć WordSmith-regularnih izraza nije ekvivalentna moći POSIX-regularnih izraza jer u slučaju WordSmith-regularnih izraza ne postoji ekvivalent Klinijevom i pozitivnom zatvorenju (v. Definiciju 2.6, str. 89, i Primer 3.6, str. 163), tj. „džoker-znacima" se može opisati samo potklasa regularnih skupova koje se POSIX-regularnim izrazima opisuju isključivo primenom Klinijevog ili pozitivnog zatvorenja.

Prioritet regularnih operacija je isti kao kod POSIX-regularnih izraza, tj. unija (/) ima niži prioritet u odnosu na dopisivanje, pa time i na primenu „džoker-znakova". Za razliku od POSIX-regularnih izraza, zagrade nisu metakarakteri i ne koriste se za grupisanje, odnosno zaobilaženje ugrađenog prioriteta operacija. Ulogu zagrada su delimično preuzele liste reči kojima se izbegava učestalo korišćenje metakaraktera /[51]. Lista reči se čuva u datoteci, u svakoj liniji po jedna reč, a prilikom zadavanja upita, umesto kucanja regularnog izraza koji predstavlja uniju reči iz liste, upit se učitava iz datoteke sa listom reči.

Prilikom pretrage se podrazumeva nerazlikovanje velikih i malih slova u zapisu korpusnih reči, što korisnik može da promeni primenom specijalnog para metakaraktera == ispred i iza WordSmith-regularnog izraza koji razlikuje mala i velika slova.

Ukoliko korisnik obezbedi lematizovanu listu reči (v. pododeljak Jezički resursi), pretraga tekstova korpusa se može pojednostaviti time što se umesto regularnog izraza za opis flektivne paradigme koristi samo lema[52].

Raspoložive statističke funkcije Za statističku obradu teksta WordSmith Tools koristi pet modula:

- Character Profiler računa frekvencije pojedinačnih karaktera u tekstu.

[51] Time se takođe izbegava ograničenje dužine izraza koji predstavljaju uniju više regularnih izraza (izraz sastavljen od regularnih izraza razdvojenih metakarakterom / je ograničen na 80 karaktera).

[52] Prilikom testiranja demo verzije se ispostavilo da se u upitu može navesti samo jedna lema i ništa drugo, dok je u posebnom dijalogu dozvoljeno da se kontekst preciznije opiše navođenjem željenih ili nepoželjnih korpusnih reči, ali ne i lema.

- WordList je modul zadužen za kreiranje listi učestanosti i n-grama koje se u terminologiji dokumentacije za WordSmith nazivaju **listama reči** (eng. **word lists**).
- WSConcgram pronalazi uopštenja n-grama, tzv. **konkgrame** (eng. **concgram**). Za razliku od n-grama gde se uvek radi o sekvenci n uzastopnih reči, u slučaju konkgrama se radi o n suštinski povezanih reči, bez obzira da li su uzastopne ili ne.
- KeyWords je modul zadužen za pronalaženje ključnih (karakterističnih) reči teksta. Ulaz za modul Keywords su dve liste učestanosti. Prva lista učestanosti je kreirana pomoću alata WordList na osnovu korisnikovog korpusa, dok druga predstavlja listu učestanosti referentnog korpusa. Ključnim (karakterističnim) rečima teksta se smatraju tekstuelne reči sa učestanošću u tekstu daleko većom ili daleko manjom u poređenju sa njenom učestanošću u posmatranom referentnom korpusu (tzv. „pozitivne i negativne ključne reči").
- Concord ima mogućnost da uz generisanje konkordanci obavi i kolokacionu analizu (v. odeljak 3.3, str. 193).

Prilikom statističkih izračunavanja korisniku stoje na raspolaganju podešavanja širine opsega (prozora) za računanje kolokacija, izbor statističke mere (Pirsonov test, test logaritamske verodostojnosti), zadavanje liste reči koje ne treba uzeti u obzir pri računanju (lista stop-reči), zadavanje rečnika lema i njihovih oblika (v. odeljak Jezički resursi) tako da se mogu izračunati i učestanosti lema, a ne samo korpusnih reči kao njihovih oblika, itd.

Xaira

Zvaničan sajt `http://xaira.sourceforge.net/`

Licenca *XML-orijentisana arhitektura za indeksiranje i pronalaženje* (eng. *XML Aware Indexing and Retrieval Architecture*, skr. *XAIRA, Xaira*) je bespla-

Tabela 4.3: Primeri upita koje podržava modul Concord

Upit	**Regularni skup**	**Objašnjenje**
`autor`	$\{autor, Autor, AUTOR, AuToR, \ldots\}$	32 varijante u zapisivanju velikih i malih slova korpusne reči `autor`
`==autor==`	$\{autor\}$	Korpusna reč `autor` (upit koji pravi razliku između velikih i malih slova)
`autor*`	$\{autor, Autorstvo, autoritet, \ldots\}$	Sve korpusne reči koje počinju sa `autor`
`*autor`	$\{autor, Koautor, kantautor, \ldots\}$	Sve korpusne reči koje se završavaju sa `autor`
`*autor*`	$\{autor, Koautor, Autorski, koautorski, kantautorov, autorov, kantautor, \ldots\}$	Sve korpusne reči koje sadrže `autor` kao podnisku
`k*`	$\{ko, kad, kada, kuda, Ko, Kad, \ldots\}$	Sve korpusne reči koje počinju karakterom `k` ili `K`
`*ski`	$\{srpski, Autorski, niski, \ldots\}$	Sve korpusne reči koje se završavaju niskom `ski`
`ko * autor*`	$\{ko\ je\ autor, KO\ GRDI\ AUTORE, ko\ nudi\ autorima, ko\ su\ autori, \ldots\}$	Svi nizovi od po tri korpusne reči, pri čemu je prva `ko`, druga je proizvoljna, a treća počinje niskom `autor`
`autor?`	$\{autora, autoru, autor., autor!, \ldots\}$	`?` predstavlja jedan proizvoljan karakter
`autor^`	$\{autora, autoru, autore, autori, \ldots\}$	`?` predstavlja jedno proizvoljno slovo alfabeta izabranog jezika
`autor/pisac`	$\{autor, pisac, Autor, Pisac, \ldots\}$	sve varijante u zapisivanju velikih i malih slova korpusnih reči `autor` i `pisac`
`19##`	$\{x \mid x$ je ceo broj, $1900 \leq x \leq 1999\}$	Svi četvorocifreni brojevi koji počinju sa 19

tan i slobodan softver otvorenog koda (Burnard [2006])[53]. Xaira je naslednik *SGML-orijentisane aplikacije za pronalaženje* (eng. *SGML Aware Retrieval Application*, skr. *SARA*), promovisane juna 1994. godine na Oksfordskom univerzitetu. SARA je kreirana u okviru projekta izgradnje Britanskog nacionalnog korpusa (BNC) sa ciljem da omogući akademskoj zajednici što jednostavniji pristup i efikasnu analizu korpusa koji su anotirani primenom SGML-a, pre svega BNC-a ([Aston & Burnard, 1998]). SARA je bila dostupna kao besplatan softver, ali ne i kao slobodan softver otvorenog koda, i, uprkos početnim ciljevima, njena uloga se svela na pretragu i analizu jednog korpusa, BNC-a.

Pojavom XML-a i potiskivanjem SGML-a kao standarda za anotaciju, autori programa, Lu Bernard (Lou Burnard) i Toni Dod (Tony Dodd), su rešili da izmene implementaciju, obezbede podršku za standarde XML, TEI, XCES i Unicode, odvoje program od BNC-a, odnosno da omoguće ravnopravan tretman svih korpusa sastavljenih od dobro formiranih XML-dokumenata kao pojedinačnih tekstova. Navedene izmene u implementaciji su iziskivale i promenu naziva programa u XARA (akronim od XML Aware Retrieval Application). S obzirom da se ispostavilo da je naziv XARA već zaštićen[54], autori su na kraju izabrali akronim Xaira[55]. U međuvremenu je objavljena i XML-verzija BNC-a ([Burnard, 2007]), tako da je Xaira preuzela i poslednju ulogu koju je nekada imala SARA.

Prve verzije programa Xaira nisu bile otvorenog koda. Međutim, počev od verzije 1.12 iz aprila 2005. godine Xaira je dostupna u skladu sa uslovima licence GNU GPL 2.0 kao besplatan i slobodan softver otvorenog koda. Autorska prava pripadaju Univerzitetu u Oksfordu[56].

Platforma SARA i prve verzije programa Xaira su bile dostupne isključivo za platformu Windows. Objavljivanjem otvorenog koda, napisanog u programskom

[53] Prema autorima, ime programa se izgovara kao englesko ime *Sarah* (*Sara*), mada neki koriste i *Zara* kako bi napravili razliku u odnosu na prethodnu verziju aplikacije (`http://projects.oucs.ox.ac.uk/xaira/index.xml?ID=name`).

[54] `http://www.xara.com/us/`

[55] Detaljniji opis kako je izabrano konačno ime Xaira se može naći na adresi `http://projects.oucs.ox.ac.uk/xaira/index.xml?ID=name`

[56] U originalu „© Chancellor, Masters and Scholars of Oxford University".

jeziku C++, Xaira postaje dostupna i za druge platforme, posebno Unix i njegove derivate (Linux, FreeBSD, Mac OS X).

Klijent-server arhitektura i veb Xaira je osmišljena kao objektno-orijentisana aplikacija sa klijent-server arhitekturom, ali je u potpunosti implementirana samo na platformi Windows kao skup sledećih alata:

- *xaira_ daemon.exe* (serverski deo aplikacije),
- *xaira-indexer.exe* (alat za indeksiranje, koristi se iz komandne linije),
- *xaira-tools.exe* (grafičko sučelje za administriranje korpusa, tj. indeksiranje tekstova i obradu metapodataka),
- *xaira.exe* (klijentski deo aplikacije, namenjen za pretragu i analizu korpusa).

Datoteke proizvedene tokom indeksiranja su nezavisne od platforme, tj. moguće je obaviti migraciju korpusa sa jedne platforme na drugu bez potrebe da se korpus ponovo kreira na osnovu ulaznih tekstova.

Međutim, ako se izuzme Windows, još uvek ne postoji klijentski deo aplikacije za ostale platforme, već samo nedovoljno dokumentovano **sučelje za programiranje aplikacija** (eng. **application programming interface**, skr. **API**)[57]. Na zvaničnom sajtu se, pored klijenta za Windows, može preuzeti otvoreni kôd jednostavnih demonstracionih verzija klijenata nezavisnih od platforme, implementiranih u programskim jezicima Java i PHP.

Xaira ne omogućava preuzimanje tekstova sa veba i direktno kreiranje veb korpusa, ali tekstovi drugim putem preuzeti sa veba, prethodno transformisani u neki od zahtevanih ulaznih formata (čisti tekst, SGML, XML), mogu da se iskoriste za kreiranje korpusa.

Jedan od razloga za izbor klijent-server arhitekture je opcija da Xaira-klijent bude na računaru korpusnika, a da korpus i Xaira-server budu na udaljenom

[57]U opštem slučaju, API programa X (na primer, API programa Xaira) predstavlja opis klasa, struktura podataka i funkcija tog programa kojim se specifikuje kako bi nečija namenska aplikacija komunicirala sa programom X. Drugim rečima, API opisuje na koji način se odvija interakcija različitih softverskih komponenti.

računaru. Nažalost, ni ova opcija nije u potpunosti dostupna jer su poslednje verzije programa Xaira kompatibilne samo sa starim verzijama veb servera Apache i interpretatora za programski jezik PHP, dok sa aktuelnim verzijama ne funkcionišu.

Ažurnost i podrška Aktuelna verzija programa je 1.26, objavljena avgusta 2010. godine. Program se isporučuje sa dokumentacijom koja detaljno objašnjava pretragu, dok deo koji se odnosi na indeksiranje korpusa još uvek ima prazna poglavlja, tj. samo njihove naslove. Slično je i sa dokumentacijom sa zvaničnog sajta[58] koja je poslednji put ažurirana 2009. godine. Korisnici mogu naći dodatne informacije na forumu[59] posvećenom programu Xaira. Utisak je da se program i dokumentacija neredovno ažuriraju u poređenju sa softverom sa kojim bi Xaira morala da bude kompatibilna (v. napomenu za Apache i PHP u pododeljku Klijent-server arhitektura i veb, str. 236).

Proširivost S obzirom da je Xaira modularno organizovana, programski jezik korišćen za implementaciju (C++) je objektno-orijentisan i otvoreni kôd je javni dostupan, nema prepreka da se Xaira dalje proširuje modulima sa dodatnim funkcionalnostima.

Prethodna obrada Da bi se Xaira koristila za pretragu korpusa, korpus mora najpre da se indeksira primenom aplikacije Xaira-tools. Xaira-tools od korisnika očekuje informacije o nazivu i opisu korpusa, lokacijama ciljnog korpusa i izvornih tekstova za korpus, kao i formatu izvornih tekstova. Svi ulazni tekstovi moraju biti u istom formatu (čist tekst, SGML ili dobro formirani XML) koji takođe mora da se specifikuje pre indeksiranja. Xaira je posebno prilagođena formatu TEI, tako da se korisniku nudi kao opcija da istakne da je korpus sastavljen od TEI-dokumenata, kao i da navede kojim TEI-elementima su anotirane granice pojedinačnog teksta, odnosno granice jedinica teksta (pasusi, rečenice) koje se prikazuju kao kontekst ključnih reči u konkordancama. Takođe, bez obzira na format ulaznih tekstova (TEI ili ne), na osnovu po-

[58] `http://www.oucs.ox.ac.uk/rts/xaira/`
[59] `http://sourceforge.net/p/xaira/discussion/442267`

dataka koje korisnik specificira pre indeksiranja generiše se zaglavlje korpusa usklađeno sa TEI, koje kontroliše sam proces indeksiranja.

Korisnik takođe može da utiče na tokenizaciju, izborom ili podrazumevanih pravila standarda Unicode za tokenizaciju, ili, u slučaju da su u TEI-dokumentu već anotirani pojedinačni tokeni, navođenjem TEI-elemenata koji su upotrebljeni za razdvajanje tokena u tekstu.

Posebna pogodnost koju omogućava Xaira tokom indeksiranja je kreiranje datoteke sa bibliografskim podacima o tekstovima korpusa na osnovu metapodataka navedenih u zaglavlju pojedinačnih TEI-dokumenata koji sačinjavaju korpus.

Jezički resursi Xaira ne koristi dodatne jezičke resurse sem sopstvenih korpusa.

Tipovi pretrage Postoji više načina da se posredstvom Xaira-klijenta zada upit nad korpusom. Xaira interno koristi **XML-upitni jezik** (eng. **XML Query Language**, skr. **XQL**) koji se još označava i kao **korpusni upitni jezik** (eng. **Corpus Query Language**, skr. **CQL**)[60], tako da se svi tipovi upita interno prevode u XQL. Takođe, postoji opcija da korisnik direktno zada upit koristeći XQL. Međutim, u zvaničnoj pratećoj dokumentaciji programa sintaksa XQL-a nije precizno objašnjena, već je navedeno svega par konkretnih primera XQL-upita.

Ostali načini zadavanja upita nad korpusom u programu Xaira su:

- upit u formi POSIX-regularnog izraza koji opisuje jednu korpusnu reč (opcija Pattern);
- upit nad leksikonom korpusa, tj. skupom svih različitih korpusnih reči (opcija Word). U ovom slučaju upit se tretira ili kao POSIX-regularni izraz ili kao prefiks korpusnih reči leksikona. Rezultat upita je tabelarni prikaz svih korpusnih reči leksikona koje zadovoljavaju zadati regularni upit ili počinju zadatim prefiksom, zajedno sa odgovarajućim korpusnim

[60] CQL koji koristi Xaira nema nikakve veze sa istoimenim jezikom koji koriste IMS OCWB i SketchEngine/NoSketchEngine.

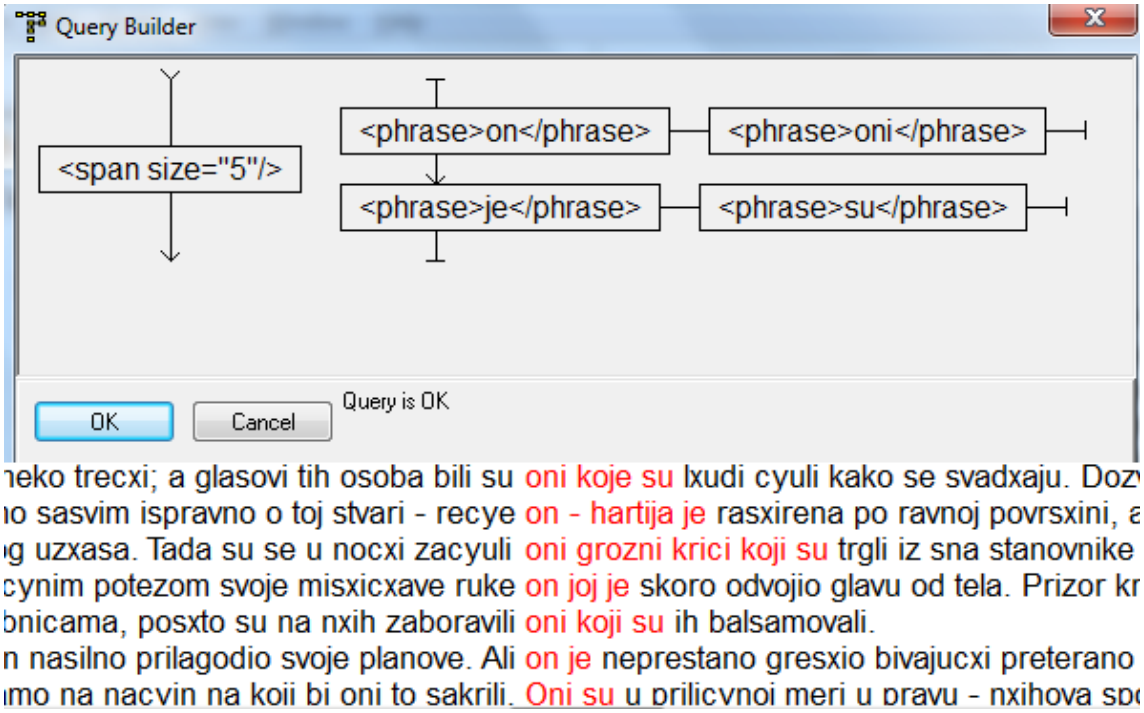
frekvencijama i brojem različitih oblika korpusne reči koji se pojavljuju u korpusu[61]. Izborom skupa lema (redova tabele rezultata) se generišu odgovarajuće konkordance, pri čemu korisnik ima na raspolaganju da vidi konkordance svih lema, ili konkordance najučestalijih lema (korisnik određuje njihov broj) ili konkordance lema čija je učestanost u zadatom intervalu. Takođe, ako je korpus anotiran informacijom o lemi, rezultati pretrage leksikona za svaku lemu (red u tabeli) sadrže podtabelu sa onim oblicima leme koji se pojavljuju u korpusu, tako da se umesto generisanja svih konkordanci leme mogu generisati samo konkordance željenih oblika.

- upit u formi jedne ili više korpusnih reči (opcija Phrase), pri čemu može da se koristi „džoker-znak" _ kao zamena za proizvoljnu korpusnu reč. Međutim, nije dozvoljeno opisivanje pojedinačnih korpusnih reči POSIX-regularnim izrazima. Ovaj način pretrage je pogodan za pronalaženje višečlanih leksema, sintagmi i fraza.
- upit u formi otvorene ili zatvorene etikete XML-elementa (opcija XML). U dijalogu se korisniku nudi lista svih XML-elemenata u korpusu iz koje on bira željeni element i tip etikete (otvorena/zatvorena). U slučaju da korisnik izabere početnu etiketu elementa koji ima atribute, nudi mu se i lista atributa iz koje može da odabere atribut i specifikuje željene vrednosti[62].

Korisnik može da zada upit i u formi grafa (opcija Query Builder) koji se interno takođe prevodi u XQL (Slika 4.3). Čvor u Xaira-grafu je ili čvor konteksta (eng. scope node) ili čvor sadržaja (eng. content node). Čvor sadržaja predstavlja upit u nekoj dozvoljenoj formi (POSIX-regularni izraz, prefiks korpusne reči, fraza, XML-element, proizvoljna korpusna reč). Svaki Xaira-graf sadrži tačno jedan čvor konteksta koji definiše kontekst pretrage.

[61] Podrazumevana podešavanja programa svaku korpusnu reč tretiraju kao jednu lemu koja ima samo jedan oblik. Međutim, ako je svaki token korpusa anotiran istoimenim XML-elementom (najčešće `w`) i pridružen mu je određeni XML-atribut (najčešće `lemma`) čija vrednost određuje lemu tokena, tada korisnik može da promeni podešavanja programa tako da pretraga uzme u obzir pridružene informacije o lemi tokena.

[62] Ukoliko atribut ima manje od 250 mogućih vrednosti, korisniku se prezentuje njihova lista, pri čemu korisnik može da izabere više željenih vrednosti.



Slika 4.3: Primer upita u formi Xaira-grafa (opcija Query Builder) i odgovarajuće konkordance. Prvi čvor sleva predstavlja kontekst pretrage (u ovom slučaju pet uzastopnih korpusnih reči), dok ostali predstavljaju čvorove sadržaja. U ovom primeru gornji čvorovi sadržaja (`on` ili `oni`) moraju prethoditi (ne nužno neposredno) donjim (`je` ili `su`) u okviru konteksta pretrage, tj. u nizu od pet uzastopnih korpusnih reči.

Xaira-graf može sadržati više čvorova sadržaja koji se povezuju horizontalno ili vertikalno. Horizontalno povezivanje uvek predstavlja alternaciju (‚ili'). Vertikalno povezivanje određuje redosled čvorova i njihovo međusobno rastojanje, kao i da li se čvorovi moraju istovremeno naći u okviru zadatog kontekstu pretrage[63].

Zahvaljujući opciji Query Text menija Query za svaki postavljeni upit se može videti ekvivalentni zapis u XQL-u, tako da se eksperimentisanjem sa različitim tipovima upita može naučiti sintaksa XQL-a uprkos nepotpunoj dokumentaciji (Slika 4.4).

[63]Postoje tačno četiri tipa vertikalnog povezivanja: a) donji čvor mora neposredno da sledi za gornjim; b) donji čvor sledi, ali ne neposredno, za gornjim i ne mora biti u okviru konteksta pretrage; c) donji čvor sledi za gornjim u okviru konteksta pretrage; d) donji i gornji čvor moraju biti u okviru kontekstu pretrage, ali u proizvoljnom redosledu.

```
1  <scope>
2    <prod>
3      <or>
4        <phrase>on</phrase> <phrase>oni</phrase>
5      </or>
6      <or>
7        <phrase>je</phrase> <phrase>su</phrase>
8    </or>
9    </prod>
10   <span size="5"/>
11 </scope>
```

Slika 4.4: XQL-ekvivalent upita sa Slike 4.3

Raspoložive statističke funkcije Xaira ima mogućnost generisanja liste učestanosti tokena, a u slučaju da je korpus anotiran informacijom o lemi, može se generisati i lista učestanosti lema. Kada su u pitanju n-grami, ne postoji opcija njihovog generisanja.

Za generisane konkordance postoji opcija kolokacione analize koja se zasniva na upoređivanju relativnih učestanosti tokena u prozoru (opsegu) zadate širine sa relativnim učestanostima istih tokena u celom korpusu. Xaira koristi dve statističke mere za kolokacionu analizu: z-test i test zajedničke informacije.

Prilikom pretrage korpusa u programu Xaira je moguće podeliti tekstove u particije i klase (na primer funkcionalni stilovi i registri) bilo ručno, bilo automatski na osnovu vrednosti izabranog XML-elementa ili rezultata prethodno zadatog upita. Kada je podela na particije i klase precizirana, pri narednom generisanju konkordanci za zadati upit, moguće je generisati i raspodelu rezultata po definisanim particijama i klasama (broj rezultata po klasi, procentni udeo klase u ukupnom rezultatu, koliko tekstova iz klase se pojavljuje u rezultatu, itd.).

Deo III

Korpus savremenog srpskog jezika

5

Korpus savremenog srpskog jezika (SrpKor)

5.1 Projekti

Pripremu i izgradnju Korpusa savremenog srpskog jezika su neposredno ili posredno pomogli sledeći domaći i međunarodni projekti:

- *Matematička i računarska lingvistika*, SR Srbija, 1981–1985. godine;
- *Automatska obrada teksta*, OZN Beograda, SR Srbija, 1983–1985;
- 1.51 *Računarstvo sa primenama*, RZN SRS, 1986–1990;
- *Jezičke industrije* (eng. *Language Industries*), Evropska unija, 1989–1991. godine;
- *Trans-evropska infrastruktura jezičkih resursa I–II* (eng. *Trans European Language Resources Infrastructure I–II*, skr. *TELRI I–II*), 1995–2001. godine;
- 1743 *Interakcija teksta i rečnika*, Ministarstvo za nauku Republike Srbije, 2002–2004. godine;
- 148021 *Teorijsko-metodološki okvir za modernizaciju opisa srpskog jezika*, Ministarstvo za nauku Republike Srbije i SANU, 2005–2010. godine;

- *Building Language Resources and Translation Models for Machine Translation focused on South Slavic and Balkan Languages*, SEE-ERA.NET (ICT 10503 RP), 2007–2008;
- III 178006 *Srpski jezik i njegovi resursi: teorija, opis i primene*, Ministarstvo za obrazovanje i nauku Republike Srbije, 2011–2014. godine;
- III 47003 *Infrastruktura za elektronski podržano učenje u Srbiji*, Ministarstvo za obrazovanje i nauku Republike Srbije 2011–2014. godine;
- *Resursi Srednje i Jugoistočne Evrope* (eng. *Central and South-East European Resources*, skr. *CESAR*)[1], 2011–2013. godine.

5.2 Istorijski pregled (2002–2013)

Prva verzija Korpusa savremenog srpskog jezika (u daljem tekstu: SrpKor) je nastala 2003. godine pod nazivom **Neetiketirani korpus srpskog jezika** (skr. **NETK**). Naziv *Neetiketirani korpus srpskog jezika* (skr. *NETK*) je potekao od odluke da prva zvanična verzija Korpusa savremenog srpskog jezika ne sadrži ni lingvističku anotaciju ni bibliografske podatke o tekstovima, već da se te informacije dodaju kasnije.

Razvoj NETK-a je finalizovan u okviru projekta 1743 *Interakcija teksta i rečnika* (2002-2004)[2] koji je za cilj imao izgradnju novih i unapređivanje postojećih jezičkih resursa za srpski jezik. Projekat je okupio istraživače sa Univerziteta u Beogradu (Matematički, Filološki i Rudarsko-geološki fakultet) i Univerziteta u Novom Sadu (Filozofski fakultet). Na prvom sastanku učesnika projekta, održanom 14. decembra 2002. godine na Filozofskom fakultetu u Novom Sadu, članovi Grupe za jezičke tehnologije su izložili trenutno stanje jezičkih resursa za srpski jezik koje su razvili, kao i planove za razvoj novih resursa:

- *Izgradnja korpusa savremenog srpskog jezika - prvi rezultati* (Duško Vitas),

[1] ICT Policy Support Programme, Grant agreement no.: 271022.
[2] Projekat je finansiralo Ministarstvo za nauku Republike Srbije.

- *Sistemi elektronskih rečnika u sistemu INTEX* (Cvetana Krstev),
- *Metode paralelizacije tekstova i njihove eksploatacije* (Ivan Obradović)
- *Semantičke mreže - WordNet i njegovi izvodi* (Gordana Pavlović Lažetić)
- *Napredne metode pretraživanja on-line sadržaja na prirodnom jeziku* (Nebojša Vasiljević),
- *Regularni izrazi u obradi prirodnih jezika* (Miloš Utvić).

Do septembra 2003. godine aktivnosti[3] na izgradnji NETK-a su obuhvatale:

- prikupljanje tekstova za korpus (Duško Vitas),
- testiranje različitih alata za kreiranje i pretragu korpusa (Goran Jovanović, Ivona Marić, Duško Višić, Miloš Utvić),
- izradu baze podataka i veb sučelja koji bi omogućili kontrolisan pristup korpusu (registraciju i autorizaciju korisnika), administraciju korpusa (upravljanje korpusima i korisničkim nalozima) i jednostavnu pretragu korpusa (Željko Pajkić).

Za fizičku lokaciju budućeg korpusa izabrana je Računarska laboratorija Matematičkog fakulteta Univerziteta u Beogradu (skr. RLAB-MATF). Učesnici projekta su od samog početka podržavali korišćenje slobodnog softvera otvorenog koda. Posle konsultacija sa tehničkim osobljem RLAB-MATF-a, posebno sa Milanom Vukosavljevićem, administratorom mreže MATF-a i učesnikom projekta 1743, odlučeno je da se na računaru predviđenom za NETK koristi sledeći softver:

- operativni sistem: Linux[4], distribucija Slackware[5];
- veb server: Apache[6];

[3] U pripremnim aktivnostima na izgradnji NETK-a, pored istraživača na projektu 1743, učestvovali su i studenti osnovnih studija na Matematičkom fakultetu u Beogradu: Ivona Marić, Goran Jovanović, Duško Višić i Željko Pajkić.

[4] `http://www.linux.org`

[5] `http://www.slackware.com`

[6] `http://www.apache.org`

- sistem za upravljanje bazama podataka: MySQL[7];
- programski jezik za izradu veb sučelja korpusa: PHP[8].

Od svih testiranih alata za kreiranje i pretragu korpusa, izabran je IMS Corpus Workbench (skr. IMS CWB), detaljno opisan u odeljku 4.3, str. 209.

Prikupljanje tekstova za korpus, razvoj metoda izgradnje i obrade korpusa su započeli još tokom ranijih domaćih i međunarodnih projekata na kojima je učestvovala i Grupa za jezičke tehnologije (v. odeljke 5.1 i 6.1). Duško Vitas je na osnovu prikupljenih tekstova odredio sadržinu i strukturu NETK-a, tj. tekstove NETK-a i odnos pojedinih tipova teksta.

Od sredine septembra do početka decembra 2003. godine se intenzivno radilo na obradi tekstova za korpus (v. odeljak 6.2) i kreiranju prvih probnih verzija korpusa. Prikupljeni tekstovi za NETK su već bili u nekom od formata digitalnog teksta: HTML, SGML ili format čistog teksta. Stoga se obrada korpusnih tekstova svela na prilagođavanje formatu vertikalnog teksta koji je IMS CWB očekivao od ulaznih datoteka, kao i na primenu kodne sheme aurora (odeljak 6.2, str. 277). Korpusne tekstove su obradili Duško Vitas (konverzija u kodnu shemu aurora), Cvetana Krstev (uklanjanje anotacije) i Miloš Utvić (konverzija u kodnu shemu aurora, uklanjanje anotacije i konverzija u format vertikalnog teksta). Koristeći IMS CWB, probne verzije NETK-a i prvu zvaničnu verziju NETK-a je kreirao Miloš Utvić koji je u međuvremenu izabran za administratora SrpKor-a.

Prva zvanična verzija Korpusa savremenog srpskog jezika (NETK) postala je dostupna korisnicima sredinom decembra 2003. godine[9]. Prve korisničke naloge su dobili istraživači sa projekta i donatori korpusa, a širenjem informacije o postojanju NETK-a i broj njegovih korisnika se postepeno povećavao[10].

Veličina NETK-a, odnosno korpusa SrpKor2003, je 22,2 miliona reči (Tabela 5.1). Detaljni uporedni pregled dva jezička resursa za srpski jezik, NETK-a i elektronskog morfološkog rečnika, kao i struktura NETK-a, su dati u [Krstev & Vitas, 2005].

[7] `http://www.mysql.com`

[8] `http://www.php.net`

[9] U tom trenutku adresa zvaničnog sajta je bila `http://www.korpus.matf.bg.ac.yu`.

[10] Do 2011. godine, tj. do pojave nove verzije korpusa (SrpKor2011), se registrovalo oko 300 korisnika korpusa NETK (SrpKor2003).

Tabela 5.1: Veličina korpusa NETK (SrpKor2003)

NETK/SrpKor2003			
tokeni	**tipovi**	**korpusne reči**	**korpusni tipovi**
27.572.229	607.910	22.203.417	603.286

Tokom 2004. godine administrator SrpKor-a je razvio podsistem za generisanje listi učestanosti korpusnih reči, bigrama i trigrama (v. odeljak 3.3) i primenio ga na NETK, čime su dobijeni prvi rezultati statističke analize Korpusa savremenog srpskog jezika ([Krstev & Vitas, 2005]).

Istovremeno je započeo rad na pridruživanju bibliografskih informacija tekstovima SrpKor-a. Bibliografske opise tekstova NETK-a je prikupio i uneo u bazu podataka Duško Vitas, a Miloš Utvić je modifikovao bazu podataka i veb sučelje SrpKor-a tako da se uz rezultate pretrage (konkordance) prikazuju i odgovarajući bibliografski opisi izvora. Modifikovana verzija NETK-a je promenila naziv u SrpKor2003.

Još tokom pripreme prve verzije NETK-a, Grupa za jezičke tehnologije je razmatrala moguće pristupe strukturnoj i lingvističkoj anotaciji buduće verzije Korpusa savremenog srpskog jezika. S obzirom na uticaj koji su krajem devedesetih godina XX veka imali standardi za anotaciju TEI (v. odeljak 2, str. 129) i SGML (v. odeljak 2.4, str. 121), još pre početka izgradnje NETK-a je jedan manji deo korpusnih tekstova bio anotiran u skladu sa smernicama TEI P3, odnosno realizovan u obliku SGML-dokumenata. Upravo su ti anotirani tekstovi korišćeni tokom prvih testiranja alata za paralelizaciju teksta ([Vitas & Krstev, 1998; Vitas et al., 1998]), tj. kao osnova budućih paralel(izova)nih korpusa: englesko-srpskog i francusko-srpskog.

U slučaju lingvističke anotacije, prvi koraci su načinjeni ka specifikaciji i izboru anotacije morfosintaksičkih opisa. Duško Vitas i Cvetana Krstev su još tokom devedesetih godina XX veka razvili sistem morfosintaksičkih opisa u formatu LADL/DELA, primenjen u elektronskom morfološkom rečniku srpskog jezika (v. odeljak 2, str. 145). Tokom izrade NETK-a, Grupa za jezičke tehnologije se pridružila projektu MULTEXT-East (v. odeljak 2, str. 146) i započela izradu alternativne morfosintaksičke anotacije ([Erjavec, Krstev, Petkevič, Simov, Tadić & Vitas, 2003]). Duško Vitas i Cvetana Krstev su, u saradnji sa studentkinjom Filološkog fa-

kulteta Katarinom Todorović, morfosintaksički anotirali elektronsku verziju romana *1984* Džordža Orvela u formatu LADL/DELA, a potom su istom tekstu pridružili i morfosintaksičku anotaciju u formatu MULTEXT-East za koji su prethodno razvili specifikaciju za srpski jezik u okviru istoimenog projekta ([Krstev et al., 2004]).

Tokom perioda od 2005. do 2010. godine Grupa za jezičke tehnologije učestvuje u radu projekta 148021 *Teorijsko-metodološki okvir za modernizaciju opisa srpskog jezika*[11], u okviru kojeg nastavlja sa razvojem novih i održavanjem postojećih jezičkih resursa koje je razvila tokom prethodnih projekata. Posebna pažnja je posvećena:

- konstrukciji paralel(izova)nih korpusa i razvoju odgovarajućih alata ([Vitas et al., 2006], [Vitas & Krstev, 2006], [Obradović et al., 2008]),
- proširivanju elektronskog morfološkog rečnika višečlanim leksemama (eng. compounds) [Krstev et al., 2006], [Krstev, 2008] i
- razvoju leksičkih resursa za prepoznavanje i klasifikaciju imenovanih entiteta ([Krstev, Vitas & Gucul, 2005], [Krstev, Vitas, Maurel & Tran, 2005], [Tran et al., 2005], [Vitas et al., 2007], [Krstev & Vitas, 2007], [Maurel et al., 2007], [Utvić, 2008]).

Kada su paralelni korpusi u pitanju, Grupa za jezičke tehnologije je započelu izgradnju paralelnog englesko-srpskog korpusa formiranjem dva potkorpusa: SELFEH i BALKANTIMES.

Potkorpus *SELFEH* (skr. od eng. *Serbian-English Law Finance Education and Health*) je paralelni englesko-srpski potkorpus sastavljen od tekstova iz domena finansija, zdravlja, zakona i obrazovanja, formiran u okviru projekta INTERA ([Gavrilidou et al., 2006]). SELFEH sadrži preko 150 paralelnih tekstova u formatu TMX (v. odeljak 6.5, str. 294) sa ukupno milion korpusnih reči. Korpusne reči SELFEH-a su anotirane odgovarajućom lemom i vrstom reči. U izgradnji SELFEH-a su učestvovali Cvetana Krstev, Duško Vitas, Gordana Pavlović Lažetić, Ivan Obradović i Sandra Gucul.

BALKANTIMES je potkorpus sastavljen od vesti sa sajta *Southeast European Times* (skr. *SETimes*) koje obuhvataju informacije o Jugoistočnoj Evropi („tekući

[11]Projekat su finansirali Ministarstvo za nauku Republike Srbije i SANU.

događaji, ekonomija, diplomatija, film, turizam, sportovi i nauka") na deset jezika: albanskom, bošnjačkom[12], hrvatskom, engleskom, makedonskom, srpskom i turskom[13]. Svaki tekst potkorpusa BALKANTIMES je pripremio po jedan student Filološkog fakulteta Univerziteta u Beogradu, sa Katedre za opštu lingvistiku ili sa Katedre za bibliotekarstvo i informatiku, kao seminarski rad u okviru predmeta Informatika IV i Primenjena lingvistika u periodu od školske 2003/2004. godine zaključno sa školskom 2009/2010. godinom. Predmetni nastavnik Cvetana Krstev i saradnik Miloš Utvić su organizovali i nadgledali izradu seminarskih radova, a potom pregledali i korigovali završene radove. U izgradnji BALKANTIMES-a su značajno pomogli i:

- Sandra Gucul, student postdiplomskih studija na Filološkom fakultetu Univerziteta u Beogradu (obuka u instalaciji i korišćenju alata za paralelizaciju tekstova XAlign i Concordancier, v. odeljak 6.5, str. 294);

- Ranka Stanković, asistent Rudarsko-geološkog fakulteta i član Grupe za jezičke tehnologije (izrada softverskog modula za konverziju paralelnog teksta iz formata TEI u formate TMX i HTML, v. odeljak 6.5, str. 294);

- Tanja Samardžić, asistent sa Katedre za opštu lingvistiku (u svojstvu saradnika u nastavi za predmet Primenjena lingvistika je organizovala i nadgledala izradu seminarskih radova studenata sa Katedre za opštu lingvistiku, a potom pregledala i korigovala završene radove školske 2005/2006. godine, školske 2007/2008. i školske 2008/2009. godine).

Detalji o pripremi paralelnih korpusa se mogu naći u odeljku 6.5.

Tokom trajanja projekta *Teorijsko-metodološki okvir za modernizaciju opisa srpskog jezika* nastavljeno je prikupljanje novih tekstova za SrpKor i unapređivanje veb sučelja (pretraga paralelnih korpusa, ekstrakcija i snimanje izabranih konkordanci),

[12]U originalnom tekstu preuzetom sa adrese `http://www.setimes.com/cocoon/setimes/xhtml/sr_Latn/document/setimes/footer/about/about` stoji „bosanskom".

[13]Na početku izgradnje potkorpusa BALKANTIMES, 2004. godine, sajt je koristio adresu `http://www.balkantimes.com`, po kojoj je potkorpus i dobio ime. Sadašnja adresa sajta (2013. godine) je `http://www.setimes.com`. Sponzor SETimes-a je Američka komanda u Evropi (EUCOM), združena vojna komanda odgovorna za operacije Sjedinjenih Država u 52 zemlje.

ali sam korpus nije ažuriran. Nove tekstove je prikupljao Duško Vitas, dok je Cvetana Krstev, koristeći elektronski morfološki rečnik srpskog jezika, korigovala tekstove verzije SrpKor2003, kako bi ispravljene verzije bile uključene u novu verziju SrpKor-a.

Manji deo novih tekstova je prikupljen kroz seminarske radove studenata Filološkog fakulteta Univerziteta u Beogradu, posebno sa Katedre za bibliotekarstvo i informatiku, kao i studenata Matematičkog fakulteta Univerziteta u Beogradu.

Intenzivnu obradu prikupljenih tekstova za novu verziju SrpKor-a je započeo Miloš Utvić tokom jula i avgusta 2009. godine. Tom prilikom je pripremljeno oko 130 novih tekstova (književnoumetnički, naučno-popularni i feljtoni iz dnevnih novina *Danas*) koji su delimično ili potpuno strukturno anotirani u skladu sa Smernicama TEI.

Tokom školske 2010/2011. godine Miloš Utvić je okupio grupu studenata master studija (Jelena Andonovski, Biljana Đorđević, Katarina Stanišić, Tijana Stojković) koji su prethodno završili osnovne studije na Katedri za bibliotekarstvo i informatiku Filološkog fakulteta Univerziteta u Beogradu i koji su tokom studija odslušali predmete Struktura informacija (kao obavezni predmet), Metode obrade prirodnih jezika i Elektronsko izdavaštvo i digitalne biblioteke (kao izborne predmete). U okviru navedenih predmeta studenti su savladali teorijska i praktična znanja koja se koriste prilikom konstrukcije korpusa (primena regularnih izraza, konstrukcija dobro formiranog i validnog XML-dokumenta, strukturna i bibliografska anotacija u skladu sa smernicama TEI), tako da su uzeli učešće u delimičnoj obradi novih tekstova za SrpKor i prikupljanju i unosu odgovarajućih bibliografskih opisa tih tekstova. Kontrolu prethodne obrade i završnu obradu tekstova je obavio Miloš Utvić.

Tokom 2011. godine su započela tri projekta u kojima je učestvovala Grupa za jezičke tehnologije, a koji su podržali dalju izgradnju SrpKor-a:

- III 178006 *Srpski jezik i njegovi resursi: teorija, opis i primene*[14] (2011-2014),
- III 47003 *Infrastruktura za elektronski podržano učenje u Srbiji*[15] (2011-2014) i

[14]Projekat finansira Ministarstvo za obrazovanje i nauku Republike Srbije.
[15]Projekat finansira Ministarstvo za obrazovanje i nauku Republike Srbije.

- *Resursi Srednje i Jugoistočne Evrope* (eng. *Central and South-East European Resources*, skr. *CESAR*)[16], 2011-2013. godine.

Glavni pokretač izgradnje SrpKor-a u periodu od 2011. do 2013. godine je projekat CESAR. CESAR je imao za cilj prikupljanje, povezivanje, nadgradnju i dokumentovanje digitalnih jezičkih resursa i alata na bugarskom, hrvatskom, mađarskom, poljskom, slovačkom i srpskom jeziku ([Váradi, 2011]). CESAR je deo šire mreže META-NET koju finansira Evropska unija sa ciljem da se izgradi višejezični evropski digitalni informacioni prostor koji bi omogućio pristup i razmenu kvalitetnih digitalnih jezičkih resursa i softverskih rešenja za eksploataciju takvih resursa ([Ogrodniczuk et al., 2012]). CESAR je omogućio angažovanje spoljnih saradnika, pre svega nastavak saradnje sa diplomiranim studentima master studija koji su već stekli iskustvo u obradi tekstova za SrpKor (Jelena Andonovski, Biljana Đorđević, Katarina Stanišić, Tijana Stojković), kao i angažovanje novih saradnika (Biljana Lazić). Na taj način su se stvorili uslovi da se administrator SrpKor-a, Miloš Utvić, posveti morfosintaksičkoj anotaciji SrpKor-a. Da bi se spoljni saradnici osposobili da samostalno mogu da obrade novi tekst za SrpKor od početka do kraja (konverzija iz drugih formata u format čistog teksta, strukturna anotacija, konverzija pisma i kodnog rasporeda, detekcija i ispravljanje grešaka u tekstu i kodnom rasporedu), Miloš Utvić je razvio programski alat CorpusPreprocessor (v. odeljak 6.2, str. 270) i obučio spoljne saradnike da koriste taj alat za potrebe automatske i poluautomatske obrade novih tekstova za SrpKor.

Nove tekstove za SrpKor u periodu od 2011. do 2013. godine su prikupljali Goran Rakić i Miloš Utvić.

Goran Rakić je napisao skriptu na programskom jeziku Pajton pomoću koje su preuzeti članci iz 1750 brojeva dnevnih novina *Politika* objavljenih u periodu od 2005. do 2011. godine. Obradu tih članaka i pripremu njihovih bibliografskih opisa je obavio Miloš Utvić.

Miloš Utvić je uz pomoć alata *wget* i skripti napisanih na programskom jeziku awk prikupio zakone koje je usvojila Narodna skupština Republike Srbije tokom perioda od 2001. do 2011. godine. Katarina Stanišić i Tijana Stojković su obradile

[16]ICT Policy Support Programme, Grant agreement no.: 271022.

te administrativne tekstove i pripremile odgovarajuće bibliografske opise. Miloš Utvić je takođe prikupio stotinak novih tekstova, pre svega književnoumetničkih, naučnih i naučno-popularnih, koje su potom obradili spoljni saradnici.

U periodu od 2011. do 2013. godine SrpKor je ažuriran tri puta, tj. Miloš Utvić je generisao tri verzije SrpKor-a:

- SrpKor2011 (jul 2011. godine, 113 miliona korpusnih reči),
- SrpKor2012 (jul 2012. godine, 118 miliona korpusnih reči) i
- SrpKor2013 (januar 2013. godine, 122 miliona korpusnih reči).

Sve tri verzije SrpKor-a (SrpKor2011, SrpKor2012 i SrpKor2013) su automatski morfološki anotirane (v. odeljak 6.3, str. 285), tekstovima SrpKor-a su pridružene odgovarajuće bibliografske informacije (videti odeljak 6.3, str. 280), uključujući i informacije o tome kom funkcionalnom stilu pripada tekst, kao i da li je tekst originalno nastao na srpskom jeziku ili je u pitanju prevod na srpski. Detalji o prikupljanju i obradi tekstova, uključujući i anotaciju, se mogu naći u poglavlju 6.

Iako je SrpKor nastao i uz finansijsku podršku Ministarstva nauke Republike Srbije, mora se konstatovati da SrpKor tokom više od deset godina svog postojanja nije imao podršku države kakvu su imali i imaju nacionalni korpusi drugih jezika, poput poljskog, ruskog, hrvatskog, a o engleskom i da ne govorimo. Dok su u izgradnji nacionalnih korpusa pomenutih jezika učestvovale pojedinačne institucije ili grupe institucija (v. odeljak 1.5), SrpKor je izgradilo nekoliko pojedinaca, istraživača i nastavnika sa Univerziteta u Beogradu, kojima su pomogli njihovi studenti, većinom izradom seminarskog rada kojim su obradili deo nekog korpusnog teksta, dok je nekolicina studenata zajedno sa svojim nastavnicima uložila ogroman trud u prikupljanju, pripremi i (bibliografskoj i strukturnoj) anotaciji tekstova SrpKor-a i paralel(izova)nih korpusa.

U periodu od 2011. godine do 2013. godine su formirani i paralelni englesko-srpski i francusko-srpski korpusi. Na izgradnji ovih korpusa su radili Cvetana Krstev, Ivan Obradović, Ranka Stanković, Jelena Andonovski i Miloš Utvić (englesko-srpski)[17], odnosno Duško Vitas, Cvetana Krstev i Miloš Utvić (francusko-srpski).

[17]Nekoliko kratkih priča Ernesta Hemingveja su paralelizovali Zoran Ristović i Bojana Đorđević.

U istom periodu je finalizovano i višejezično elektronsko izdanje romana Žila Verna *Put oko sveta za 80 dana* ([Vitas et al., 2008]). Ovaj jezički resurs obuhvata verzije romana na 18 jezika (original na francuskom i prevode na 17 jezika, uključujući i srpski). Svi prevodi su paralelizovani sa nekom od verzija na francuskom, engleskom ili srpskom jeziku, tako da resurs sadrži ukupno 32 biteksta u formatima TMX i HTML (v. odeljak 6.5, str. 294). Izradom ovog resursa je rukovodio Duško Vitas.

Srpsku verziju romana *Put oko sveta za 80 dana* je morfosintaksički anotirala Cvetana Krstev u formatu LADL/DELA tokom projekta *Building Language Resources and Translation Models for Machine Translation focused on South Slavic and Balkan Languages*, SEE-ERA.NET (ICT 10503 RP), 2007–2008. godine.

5.3 Parametri tekuće verzije Korpusa savremenog srpskog jezika (SrpKor2013)

Zvaničan sajt `http://www.korpus.matf.bg.ac.rs`

Obim (veličina) SrpKor2013 ima 122 miliona korpusnih reči (Tabela 5.2).

Tabela 5.2: Veličina korpusa SrpKor2013

SrpKor2013			
tokeni	**tipovi**	**korpusne reči**	**korpusni tipovi**
152.540.721	1.424.899	122.255.064	1.402.664

Struktura Za SrpKor je pripremljeno ukupno 5.058 tekstova, od čega je u korpus uključeno samo 4.890[18]. Za pripremljene tekstove je određena raspodela po funkcionalnim stilovima (Tabela 5.3) i raspodela po statusu teksta u odnosu na jezik na kome je tekst originalno nastao (Tabela 5.4).

Period Originalne verzije tekstova SrpKor-a su nastale u periodu od 1910. do 2012. godine (Tabela 5.5)[19]

[18]Tokom indeksiranja korpusnih tekstova je greškom izostavljeno 168 tekstova. Da su i oni uneti u SrpKor, veličina korpusa bi bila preko 123 miliona korpusnih reči.

[19]U slučaju papirnih publikacija je beležena godina kada je objavljeno odgovarajuće izdanje na osnovu kojeg je kreirana elektronska verzija teksta za korpus. Izneti su podaci za sve prikupljene

Tabela 5.3: Raspodela tekstova korpusa i korpusnih reči SrpKor2013 po funkcionalnim stilovima.

SrpKor2013	
Funkcionalni stil	**korpusni tekstovi**
književnoumetnički	348
naučni i naučno-popularni	188
novinski	3.245
administrativni	923
ostalo	354

Tabela 5.4: Raspodela tekstova korpusa i korpusnih reči SrpKor2013 po jeziku originalne verzije teksta.

SrpKor2013	
jezik originalne verzije teksta	**korpusni tekstovi**
napisan na srpskom jeziku	4545
prevod na srpski	513

Tabela 5.5: Raspodela tekstova korpusa i korpusnih reči SrpKor2013 po decenijama.

SrpKor2013	
Godina izdanja	**korpusni tekstovi**
1920–1929	3
1930–1939	5
1940–1949	0
1950–1959	13
1960–1969	32
1970–1979	31
1980–1989	36
1990–1999	157
2000–2009	2.718
2010–2013	793
neutvrđena	1.270

Izvor/medijum SrpKor se, pre svega, sastoji od pisanih tekstova[20].

Anotacija i mogućnosti pretrage SrpKor je anotiran:

- bibliografskim informacijama o korpusnim tekstovima (za detalje videti odeljak 6.3, str. 280) i

tekstove (njih 5.058).

[20]Manji deo tekstova, preuzet iz medijskih arhiva, predstavlja transkripte usmenih intervjua objavljenih u dnevnim novinama (*Politika*, `http://www.politika.rs`) ili emitovanih u okviru radijskih emisija (*Peščanik*, `http://pescanik.net`).

- morfosintaksičkim informacijama (vrsta reči i lema, za detalje videti odeljak 6.3, str. 285).

Iako su pojedini korpusni tekstovi strukturno anotirani, tj. sadrže informaciju o logičkoj strukturi teksta, sam SrpKor nije strukturno anotiran u smislu postojanja XML-etiketa, ali sadrži informaciju o krajevima segmenata (rečenica), pošto pojedini znaci interpunkcije imaju oznaku za kraj rečenice (SENT) kao vrednost „leme". O strukturnoj anotaciji korpusnih tekstova se može više naći u odeljku 6.3, str. 283.

SrpKor se može pretraživati kao kolekcija teksta, a takođe i po informacijama prisutnim u (bibliografskoj i morfosintaksičkoj) anotaciji (v. poglavlje 7).

6

Faze u kreiranju korpusa SrpKor

6.1 Prikupljanje tekstova za SrpKor

Prilikom prikupljanja tekstova za Korpus savremenog srpskog jezika (SrpKor), prednost su imali već postojeći elektronski tekstovi raznih formata (TXT, PDF, MS Word DOC i DOCX, HTML i XHTML, DJVU, EPUB, MOBI, TEI XML itd.), dok je daleko manje neelektronskih uzoraka za SrpKor digitalizovano na način opisan u odeljku Digitalizacija neelektronskih tekstova za SrpKor, str. 263. Takođe, zbog lakše dostupnosti, korpus uglavnom čine pisani tekstovi, mada ima i primera govornih tekstova (novinski intervjui, transkripti radijskih i televizijskih emisija). Originalnim tekstovima na srpskom jeziku je obuhvaćen vremenski period od 1910. godine (roman *Došljaci*) do 2012. godine, dok prevodi stranih knjiga, mahom književnih dela, obuhvataju duži period, od XVIII veka do današnjih dana.

Još tokom projekta TELRI I–II (1995–2001) su pripremljeni sledeći resursi za srpski jezik ([Erjavec, Lawson & Romary, 1998]):

- prevod Platonove *Republike* na srpski jezik u formatima čisti tekst, SGML i HTML, kao i paralelne verzije teksta na ukupno 17 jezika ([Vitas et al., 1998]);
- prevod Orvelovog romana *1984* na srpski jezik u formatima čisti tekst, SGML i HTML, kao i paralelne verzije teksta na ukupno 7 jezika ([Vitas & Krstev,

1998]);

- vesti agencije TANJUG (agencijski izveštaji u periodu od septembra do novembra 1995. godine i maj-jun 1996. godine), ukupno 1,2 miliona reči;
- književni tekstovi 13 autora (Ivo Andrić, Miroslav Josić-Višnjić, Laza Kostić, Momčilo Nastasijević, Milorad Pavić, Borislav Pekić, Veljko Petrović, Vasko Popa, Milisav Savić, Slobodan Selenić, Svetlana Velmar-Janković, itd.), ukupno 140 hiljada reči;
- prevodi na srpski, ukupno 322 hiljade reči;
- šesnaest udžbenika (razni predmeti i nivoi), ukupno 263 hiljade reči;
- jedan zakonodavni tekst, 6 hiljada reči;
- novinski tekstovi:
  - 9 hiljada reči kratkih vesti;
  - 90 hiljada reči vesti iz kulture (Vukova *Danica*).
- 6 hiljada srpskih poslovica koje je prikupio i objavio Vuk St. Karadžić.

Jedan deo tekstova je obezbeđen u saradnji sa drugim istraživačima koji razvijaju sopstvene korpuse.

Tokom izrade verzije SrpKor2003 su preuzeti neki tekstovi, pretežno književnoumetničkog stila, iz YU-korpusa srpskohrvatskog jezika Heninga Merka (Henning Mørk) sa Instituta za slavistiku Univerziteta u Arhusu, Danska. Originalnu verziju tekstova YU-korpusa Merk je proizveo 1992. godine kombinovanjem skaniranja i OCR-a. U originalnoj elektronskoj verziji tekstova primenjeno je interno kodiranje karaktera specifičnih za srpski jezik zasnovano na kodnom rasporedu ASCII. Borut Maričić je tokom 1996. godine proizveo dve nove elektronske verzije tekstova koje koriste kodni raspored MS CP-1250, odnosno ISO-8859-2, a 2000. godine i konačnu verziju koja koristi kodni raspored UTF-8. Paralelno uz konverziju u UTF-8, Višnja Puzić-Radman je čitala i kontrolisala tekstove. YU-korpus nije dostupan za pretragu, ali se pojedinačni tekstovi ovog korpusa mogu preuzeti sa adrese

veb stranica koje uređuje Borut Maričić: `http://www.yurope.com/books/yu/` i `http://www.borut.com/library/`[1]. Preuzeti tekstovi za SrpKor, ukupno oko dvadesetak, su, bez izuzetka, dela domaćih autora (Saša Božović, Milisav Savić, Dragoslav Mihailović, Miodrag Bulatović, Aleksandar Tišma, Momo Kapor, Filip David, Antonije Isaković, Vuk Drašković, i dr.).

Do još jedne grupe književno-umetničkih tekstova se došlo u saradnji sa Adrijem Barentsenom (Adrie Barentsen) sa Univerziteta u Amsterdamu. Barentsen rukovodi izradom **Amsterdamskog paralelizovanog korpusa slovenskih jezika** (eng. **Amsterdam Slavic Parallel Aligned Corpus**, skr. **ASPAC**) sa ciljem da se obezbedi materijal za kontrastivno izučavanje 14 slovenskih jezika: ruskog, beloruskog, ukrajinskog, poljskog, češkog, slovačkog, bugarskog, makedonskog, srpskog, hrvatskog, slovenačkog, gornjolužičkosrpskog, donjolužičkosrpskog i moliškohrvatskog. Svaki paralelni tekst korpusa ASPAC sadrži verziju na ruskom jeziku i na bar još jednom slovenskom jeziku, a obično i na bar još jednom neslovenskom evropskom jeziku (holandskom, engleskom, nemačkom, švedskom, italijanskom, francuskom, portugalskom, španskom, rumunskom, grčkom ili latinskom). Iako je cilj da svaki tekst bude zastupljen na što više jezika, trenutno su od slovenskih jezika najzastupljeniji ruski, poljski, češki i srpski/hrvatski, dok donjolužičkosrpski i moliškohrvatski uopšte nisu zastupljeni. Značajan deo tekstova u korpusu ASPAC čine klasična dela književnosti za decu, iz razloga što za njih postoji i po nekoliko dostupnih prevoda za većinu evropskih jezika. Zaključno sa oktobrom 2012. godine, od Barentsena je u dva navrata dobijeno oko četrdesetak tekstova na srpskom jeziku, pri čemu prednjače prevodi dela stranih pisaca (Umberto Eko, Džerom K. Džerom, Hemingvej, Iljf i Petrov, Puškin, Luis Kerol, A. A. Miln, Antoan de Sent-Egziperi, Mark Tven, Tolkin, Stanislav Lem, Nikolaj Ostrovski, i dr.), dok su od domaćih pisaca zastupljeni tekstovi Ive Andrića i Milorada Pavića.

Posebno treba istaći mnogobrojne donatore elektronskih verzija tekstova, pre svega autore monografskih publikacija (uglavnom književnih dela i udžbenika), izdavače i medijske arhive u Srbiji koji su priložili svoja dela, odnosno izdanja, u elektronskoj verziji, bilo kao „elektronski rukopis", bilo kao završnu verziju pripreme

[1]Ovim sajtovima je poslednji put pristupano 4. novembra 2012. godine.

za štampu. Detaljan pregled donatora tekstova za SrpKor izložen je u Dodatku A.

Dostupni elektronski tekstovi za SrpKor

Glavni izvor prikupljenih elektronskih tekstova je internet, pri čemu po količini teksta i njegovoj javnoj dostupnosti prednjače elektronske arhive serijskih publikacija (novine, časopisi, revije, magazini itd.) navedene u tabeli 6.1. Pri tome su dnevne novine *Politika* ubedljivo najviše zastupljene u odnosu na ostale slične serijske publikacije. To je posledica činjenice da najveći broj dnevnih novina u Srbiji prenosi identične vesti preuzete od novinskih domaćih (*Beta*, *FoNet*, *TANJUG*) i stranih novinskih agencija (*Srna*, *Hina*, *Asošijeted pres*, *Rojters*, *Frans Pres*, *A.N.S.A*, itd.), te bi stoga preuzimanje kompletnih elektronskih izdanja različitih dnevnih novina neminovno dovelo do ponavljanja istovetnih tekstova u korpusu (duplikata). Stoga je odlučeno da je najbolje preuzeti kompletna elektronska izdanja samo jednih, visoko tiražnih i uticajnih dnevnih novina, a tu se *Politika* nametnula sama po sebi kao „najstariji dnevni list na Balkanu".

S druge strane, postoje portali i forumi koji svojim članovima nude ogromne baze veza (linkova) prema dokumentima raznih formata (tekstuelnih, grafičkih, multimedijalnih, itd.), uključujući kako i one koji su slobodni i javno dostupni, tako i piratske verzije elektronskih knjiga. U nedostatku memorijskih resursa, kao i zbog zakona o autorskim pravima, portali i forumi se retko odlučuju da čuvaju same sadržaje, već obično sadrže samo njihove adrese na internetu, najčešće u vlasništvu firmi koje se bave čuvanjem ogromne količine elektronskih dokumenata (eng. file hosting) i deljenjem dokumenata (eng. document-sharing) poput `4shared.com`, Dropbox[2], `iFile.it`, MediaFire[3], RapidShare[4], SendSpace[5], Scribd[6] i mnogih drugih.

Kada su u pitanju projekti na internetu orijentisani ka tekstovima na srpskom jeziku, treba spomenuti[7]:

[2] https://www.dropbox.com
[3] `http://www.mediafire.com`
[4] `rapidshare.com`
[5] `http://www.sendspace.com`
[6] `www.scribd.com`
[7] Jedan od mogućih izvora tekstova za SrpKor je bio i sajt *Baneprevoz* (`http://baneprevoz.com`) koji je ugašen 2012. godine posle krivične prijave koju je Udruženje izdavača i knjižara Srbije predalo u decembru 2011. godine zbog distribucije piratskih izdanja domaćih i stranih pisaca.

- *Projekat Rastko: Elektronsku biblioteku srpske kulture*[8],
- *Antologiju srpske književnosti*, projekat Učiteljskog fakulteta Univerziteta u Beogradu[9],
- *Tvorac grada*[10] i
- Biblioteku `znaci.net`.

Digitalizacija neelektronskih tekstova za SrpKor

Manji broj neelektronskih tekstova, sačinjen najvećim delom od štampanih publikacija, prikupljen je za SrpKor i oni su transformisani u elektronski tekst ili prekucavanjem ili kombinovanjem skaniranja i optičkog prepoznavanja karaktera (OCR). Tekstovi za prekucavanje su deljeni na manje celine koje su obrađivali studenti Univerziteta u Beogradu, pre svega sa Matematičkog, Filološkog i Rudarsko-geološkog fakulteta, kao deo seminarskog rada. Tekstovi su prekucavani korišćenjem alfabeta aurora (v. odeljak 6.2, pododeljak Kodna shema aurora). S obzirom da i tokom prekucavanja i OCR-a dolazi do neizbežnih grešaka, nastavnici koji su pregledali seminarske radove su pokušali da detektuju što više grešaka primenom programskog sistema Unitex i elektronskog morfološkog rečnika srpskog jezika u formatu DELA. Detektovane greške su vraćane studentima koji su posle ispravke slali nastavnicima novu verziju seminarskog rada. Nastavnici su ponovo vršili kontrolu druge elektronske verzije teksta, ručno ispravljali preostale greške i tako formirali konačnu elektronsku verziju teksta.

Osim prekucavanja štampanog teksta, studenti su i anotirali logičku strukturu teksta, pre svega naslove, pasuse i fusnote, u pojedinim slučajevima stihove i epigrafe, koristeći elemente iz specifikacije TEI-Lite (v. odeljak 6.3, pododeljak Strukturna anotacija (primena specifikacije TEI-Lite)): `text`, `div`, `head`, `p`, `epigraph`, `note`, `lg`, `l`. S obzirom da se uglavnom radilo o studentima I godine koji prethodno nisu odslušali nijedan kurs koji objašnjava XML i strukturnu anotaciju teksta, u uputstvu za

[8] `http://www.rastko.rs`
[9] `http://www.ask.rs`
[10] `http://www.tvorac-grada.com`

Tabela 6.1: Elektronske arhive serijskih publikacija korišćene kao izvori za Korpus savremenog srpskog jezika

Naziv publikacije	**URL**
Danica	`http://www.vukova-zaduzbina.rs/arhiva.htm`
Ebit	`http:/www.ekonomist.co.yu/magazin/ebit`
Ekonomist	`http://www.ekonomist.co.yu`
Glasnik	`http://www.spc.rs`
Ilustrovana politika	`http://www.ilustrovana.com`
Kalibar	`www.kalibar.rs`
Moje srce	`http://www.color.rs/cmi/ljubavna.html`
Mostovi	`http://mostovi.net/`, `http://www.yurope.com/zines/mostovi`
NIN	`www.nin.co.rs`
Politika	`http://www.politika.rs`
Pravoslavlje	`www.pravoslavlje.rs`
Svet	`www.svet.rs`
TANJUG	`www.tanjug.rs`
Teološki pogledi	`teoloskipogledi.blogspot.com`
Trn	`http://www.novosti.rs`
Viva	`www.vivamagazin.info`

izradu seminarskog rada su opisani neki osnovni principi na kojima se zasnivaju **dobro formirani** (eng. **well-formed**) i **validni** (eng. **valid**) XML dokumenti. Da bi se osiguralo da će seminarski radovi koje studenti šalju biti validni XML dokumenti sa što manje grešaka u kodiranju teksta (poput ne-ASCII karaktera ili pogrešno primenjene kodne sheme aurora), pre samog slanja studenti su bili dužni da obave online-validaciju na strani `http://arhimed.matf.bg.ac.rs/~misko/flf/ip/semi.html`, a da pri tom nije bilo potrebno da studenti imaju bilo kakvo predznanje o korišćenju programa za validaciju, konceptu DTD-a, i sl. Online-validacija zahteva samo da se tekst kopira u tekstuelnu oblast (eng. textarea), a potom se aktiviranjem odgovarajuće dugmadi pozivaju funkcije napisane na jeziku JavaScript koje proveravaju da li je ulazni tekst validan XML dokument, kao i da li sadrži nedozvoljene ne-ASCII karaktere ili potencijalne greške u primeni kodne sheme aurora (v. odeljak 6.2, pododeljak Kodna shema aurora).

Na taj način je posao bolje raspodeljen jer studenti sami pronalaze i ispravljaju greške, te nastavnici-kontrolori dobijaju daleko preciznije urađene seminarske radove nego što je to bio slučaj kada su studenti slali radove bez ikakve provere, a nastavnici sami obavljali celokupnu kontrolu.

Naravno, sve greške nisu mogle biti ni detektovane, a time ni ispravljene, osim da se tekst pažljivo čita od početka do kraja. S obzirom na broj korpusnih tekstova koje treba obraditi, s jedne strane, i raspoloživih resursa (vreme, ljudski resursi, finansijski resursi za angažovanje dodatnih kontrolora), s druge strane, moralo se pribeći kompromisu i odustajanju od temeljnog ispravljanja grešaka.

6.2 Obrada elektronskih tekstova za SrpKor

Formati prikupljenih elektronskih tekstova za SrpKor mogu biti raznovrsni:

- **format čistog teksta** (eng. **plain text format**),
- **prenosivi format dokumenta** (eng. **Portable Document Format**, skr. **PDF**),
- formati dokumenta koje koristi Microsoft Word (DOC i DOCX),

- formati jezika za označavanje (HTML, XHTML, SGML, XML),
- formati elektronskih knjiga (EPUB, DJVU, MOBI), itd.

.

Da bi se kreirala finalna verzija korpusa, neophodno je transformisati polazni format prikupljenih elektronskih tekstova u format koji zahteva izabrani alat za indeksiranje teksta. Međutim, ulazni format izabranog alata za indeksiranje teksta ne mora biti pogodan za čuvanje samih korpusnih tekstova iz sledećih razloga:

(Č1) korpusni tekstovi treba da se čuvaju u formatu u kome lako mogu da se iskoriste za kreiranje bilo nove verzije istog korpusa, bilo nekog sasvim različitog korpusa;

(Č2) potrebno je da postoji mogućnost automatske provere da li se korpusni tekst čuva u propisanom formatu čuvanje korpusnih tekstova;

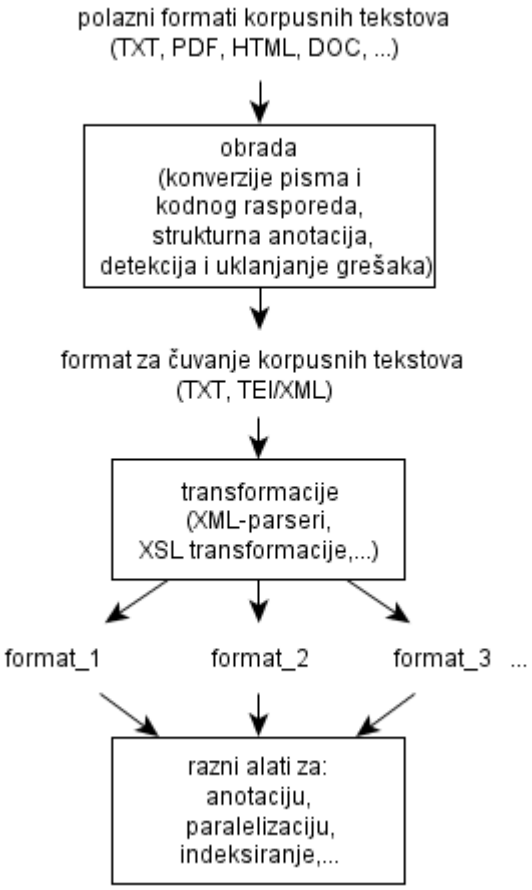
(Č3) korpusni tekstovi treba da se čuvaju u formatu iz kog mogu lako da se transformišu u formate koje zahtevaju različiti alati za anotaciju i indeksiranje teksta.

Prema tome, potrebno je precizirati ne samo ulazni format za indeksiranje teksta izborom odgovarajućeg alata, već i format za čuvanje korpusnih tekstova koji zadovoljava uslove (Č1)–(Č3).

Prva verzija SrpKor-a (NETK, odnosno SrpKor2003) je koristila isključivo format čistog teksta, kôd aurora i kodni raspored ISO 8859-1 za čuvanje korpusnih tekstova, bez elemenata strukturne anotacije. Kao posledica toga, prilikom konverzije polaznog formata korpusnih tekstova u format čistog teksta pojedini korpusni tekstovi su izgubili informacije o logičkoj strukturi teksta, bilo zbog eliminacije postojeće strukturne anotacije (na primer, eliminacijom HTML-etiketa), bilo zbog toga što pre konverzije nisu iskorišćene pogodnosti koje bi omogućile automatsku ili poluautomatsku strukturnu anotaciju korpusnog teksta (v. odeljak 6.3).

Iskustvo na izgradnji paralelizovanih korpusa (v. odeljak 6.5) je pokazalo da je bolje rešenje koristiti TEI/XML kao format za čuvanje korpusnih tekstova zato što TEI/XML zadovoljava sve uslove (Č1)–(Č3). Zato je odlučeno (Slika 6.1):

- da se prilikom izrade narednih verzija SrpKor-a (SrpKor2011 i kasnije verzije) kao format za čuvanje novih korpusnih tekstova koristi TEI/XML;
- da se stari korpusni tekstovi postepeno konvertuju iz formata čistog teksta u format TEI/XML;
- da se u slučajevima kada je na raspolaganju manje vremena nego što je potrebno za konverziju novog korpusnog teksta u TEI/XML (tj. kada tekst mora da se uključi u SrpKor pre nego što se završi konverzija u TEI/XML), korpusni tekst čuva u formatu čistog teksta dok se ne završi njegova konverzija u TEI/XML.



Slika 6.1: Formati za čuvanje i eksploataciju korpusnih tekstova.

Konverzija u format za čuvanje korpusnih tekstova

Tok konverzije u format za čuvanje tekstova SrpKor-a zavisi od toga da li rezultujuća datoteka sadrži strukturnu anotaciju ili ne, tj. da li je rezultat TEI/XML-dokument ili datoteka u formatu čistog teksta. Krajnji rezultat obrade je tekstuelna datoteka u kodnom rasporedu ISO 8859-1, koja sadrži isključivo ASCII-karaktere i kôd aurora, sa ili bez XML etiketa.

U slučaju konverzije korpusnog teksta u strukturno neanotiranu datoteku, obrada se odvija po sledećim fazama:

(I) Elektronski tekst se najpre prebacuje iz polaznog formata u **UTF8-tekst**, tj. u format čistog teksta sa kodnim rasporedom UTF-8. Kodni raspored UTF-8 obezbeđuje da karakteri specifični za srpsku ćirilicu i latinicu, kao i ostali karakteri koji bi se mogli izgubiti pri konverziji teksta u Latin-1, budu pravilno sačuvani. Način na koji će se obaviti konverzija umnogome zavisi od polaznog formata i biće detaljno razmotrena u pododeljku Konverzija u čist tekst iz pojedinih formata, str. 272.

(II) U zavisnosti od toga da li UTF8-tekst koristi ćirilicu, latinicu ili oba pisma, pristupa se odgovarajućoj konverziji alfabeta u kôd aurora (v. pododeljak Kodna shema aurora, str. 277). Izlazna datoteka, **aurora-UTF8-tekst**, i dalje koristi kodni raspored UTF-8 za slučaj da se u tekstu pojavljuju ne-ASCII karakteri (najčešće interpunkcijski znaci poput ne-ASCII navodnika, crtica, simbola specifičnih za azbuke stranih jezika, itd.).

(III) Detektuju se preostali ne-ASCII karakteri i, ako je to moguće, zamenjuju se svojim ASCII-ekvivalentima (Tabela 6.2). Karakteri koji se nalaze u gornjoj kodnoj strani rasporeda Latin-1, za koje ne postoje ASCII-ekvivalenti (npr. ©, ®, ℃, itd.), ostaju neizmenjeni. Ostali ne-ASCII karakteri koji se uglavnom svode na slova specifična za azbuke stranih jezika, obrađuju se od slučaja do slučaja:

a) Kad god je to moguće, vrši se transliteracija sa stranog jezika na srpski. U tu svrhu se mogu koristiti online-programi za transliteraciju, po-

put onih na adresama `http://translit.cc`, `http://translit.ru`, itd. Npr. ruska reč *литературный* se posle transliteracije zapisuje kao *literaturnyj*.

b) Ako su sporni karakteri deo reči, sintagme ili rečenice na stranom jeziku za koje je poznat prevod, one se zamenjuju svojim prevodom (najčešće je to slučaj sa epigramima na početku teksta).

c) U slučaju da su sve mogućnosti iscrpljene, karakter se briše, pri čemu, ako je to neophodno, briše se i veća jedinica koja ga sadrži (reč, sintagma, rečenica), tako da ostatak teksta ima smisla (npr. ¶).

(IV) Na kraju se rezultat prethodnog koraka snima kao datoteka sa čistim tekstom u kodnom rasporedu ISO-8859-1.

Tabela 6.2: Primeri ne-ASCII karaktera i njihovih ASCII-ekvivalenata

Ne-ASCII karakteri	**ASCII-ekvivalent**
” ” ″ “ ” ” ‘ ’ ´ ` ′ ′	" '
– — —	-
. . .	...
á à â ấ ā	a
ä	ae
ö	oe
ü	ue
ß	ss
½	1/2

U slučaju formata TEI/XML konverzija se odvija po istim fazama, ali se u prvoj fazi tekst postepeno pretvara u XML dokument validan u odnosu na DTD koji opisuje jedan podskup standarda TEI Lite (v. odeljak 6.3, str. 283).

Nezavisno od rezultujućeg tipa datoteke (`.txt` ili `.xml`), tj. od toga da li rezultujući korpusni tekst sadrži strukturnu anotaciju, po završetku konverzije se mogu obaviti dodatna ispitivanja sa ciljem da se pronađu i isprave eventualne greške. Obim ispitivanja je obično određen veličinom teksta, odnosno vremenom koje je na raspolaganju da se tekst dovede do finalne verzije. Ispitivanja uključuju otkrivanje i uklanjanje rastavljanja reči na kraju reda (hifenacije), otkrivanje i ispravljanje pogrešne primene kodne sheme aurora, kao i detekciju tipografskih grešaka. Poslednje

ispitivanje obuhvata prethodna dva i koristi elektronski morfološki rečnik srpskog jezika u formatu DELA, pri čemu se njegova primena ostvaruje kroz alate programskog sistema Unitex. Sva tri navedena ispitivanja najmanje vremena troše na detekciju potencijalnih grešaka, u slučaju primene prva dva testa na jednu prosečnu novinsku kolumnu detekcija je gotovo trenutna. Međutim, mnogo više vremena odlazi na odlučivanje da li je potencijalna greška zaista greška, kao i na njeno eventualno ispravljanje, pogotovo u slučaju primene rečnika, jer se preciznost od 100% postiže jedino ručnom obradom. To je i razlog za posebno izdvajanje prva dva testa, jer se, sem detekcije, i samo ispravljanje grešaka može donekle automatizovati, ali potpunu preciznost može proveriti isključivo čovek. Detaljni pregled tehnika za detekciju i ispravljanje grešaka dat je u odeljku 6.2, str. 270.

CorpusPreprocessor

Za potrebe obrade korpusnih tekstova napravljen je poseban program — CorpusPreprocessor. CorpusPreprocessor je napisan u programskom jeziku C# i prevashodno je namenjen obradi tekstova pod operativnim sistemom Windows. Ovaj program se sastoji iz tri modula: Viewer, Conversion i Database.

Modul Viewer je namenjen za obradu pojedinačnog dokumenta u formatu čistog teksta ili u formatu TEI/XML. Ovaj modul omogućava detekciju ne-ASCII karaktera u tekstu, potencijalnih grešaka u primeni kodne sheme aurora, kao i detekciju tipografskih grešaka primenom leksičkih resursa za srpski jezik programa Unitex (v. odeljak 4.3, str. 216). Takođe, u slučaju obrade dokumenta u formatu XML, postoje opcije provere da li je dokument validan u odnosu na DTD koji koristi, bilo primenom klase XmlValidatingReader[11] programskog jezika C#, bilo pozivanjem spoljašnjeg programa *xmllint.exe*[12]. Višestruka mogućnost validacije XML-dokumenta postoji zbog toga što u nekim slučajevima poruke o greškama koje pruža jedan validator

[11] Za detalje videti dokumentaciju klase XmlValidatingReader na zvaničnom sajtu `http://msdn.microsoft.com/en-us/library/system.xml.xmlvalidatingreader%28v=vs.110%29.aspx`.

[12] *xmllint.exe* je deo binarne verzije biblioteke *libxml2* za operativni sistem Windows. Adresa zvaničnog sajta biblioteke *libxml2* je `http://xmlsoft.org/`, dok verziju za Windows održava i distribuira Igor Zlatković (`http://www.zlatkovic.com/libxml.en.html`). Softver je besplatan i dostupan u skladu sa licencom MIT (`http://www.opensource.org/licenses/mit-license.html`).

nisu podjednako informativne kao kod drugog, tj. ponekad se greške lakše ispravljaju pomoću jednog validatora, a ponekad — pomoću drugog. Iz istog razloga postoji i opcija za utvrđivanje broja pojavljivanja različitih XML-elemenata u tekstu, odnosno njihovih otvorenih i zatvorenih etiketa, pošto nepoklapanje broja otvorenih i zatvorenih etiketa elementa koji se ređe pojavljuje olakšava pronalaženje i otklanjanje grešaka.

Za potrebe automatske paketne obrade datoteka razvijen je modul Conversion u okviru programa CorpusPreprocessor. Glavni deo modula omogućava korisniku da specifikuje ulazne datoteke koje treba da se obrade, bilo kao sve datoteke zadatog kataloga, bilo kao skup pojedinačnih datoteka, kao i katalog u kome će biti generisane odgovarajuće izlazne datoteke. Ostatak modula čine konkretne akcije koje treba izvršiti nad određenim skupom datoteka:

- konverzija pisma (alfabeta);
- konverzija kodnog rasporeda;
- konverzija TEI/XML u format čistog teksta bez XML etiketa;
- generisanje izveštaja o detektovanim potencijalnim greškama nastalim usled pogrešne primene koda aurora, prisustva ne-ASCII karaktera u tekstu ili kao posledica činjenice da XML-dokument nije dobro formiran, odnosno validan;
- primena XSL-transformacije, itd.

Dizajn modula Conversion je takav da omogućava lako proširivanje dodavanjem novih funkcionalnosti, tj. tipova akcija koje se mogu izvršiti nad pojedinačnom datotekom bez potrebe da se modifikuje glavni deo modula zadužen za specifikaciju ulaza i izlaza.

Modul Database predstavlja sučelje ka relacionoj bazi podataka koja sadrži evidenciju korpusnih tekstova i njihove metapodatke (identifikator teksta, identifikator autora, identifikator funkcionalnog stila, UDK-broj, godinu izdanja teksta, relativnu adresu teksta u skladištu tekstova, itd.). Ovaj modul se koristi za konverziju korpusnih tekstova iz formata za čuvanje u ulazni format alata za indeksiranje (v. odeljak 6.4).

Konverzija u čist tekst iz pojedinih formata

S obzirom da je za verziju korpusa SrpKor2013 potrebno pripremiti oko pet hiljada datoteka, jasno je da svaku fazu obrade tekstova za korpus treba maksimalno automatizovati. Nažalost, već u prvoj fazi obrade, konverzija polaznih elektronskih tekstova u format čistog teksta se ne može uvek i do kraja automatizovati.

Prva teškoća sa kojom se treba suočiti je mnoštvo različitih formata polaznih elektronskih tekstova, pri čemu su retki tzv. „univerzalni" konverteri, tj. programi koji mogu da konvertuju gotovo sve formate teksta u format čistog teksta, i uglavnom predstavljaju vlasnički softver. Zato je potrebno da se obezbede različiti programi, po mogućstvu besplatni, koji će ne samo omogućiti konverziju, već i **paketnu obradu** (eng. **batch processing**) tekstova. Program poseduje opciju paketne obrade ukoliko može da primeni istu procedure obrade na sve datoteke zadate liste, pri čemu paketna obrada, po otpočinjanju, teče automatski, tj. bez potrebe da proces nadgleda čovek.

U nastavku će biti razmotreni najčešći polazni formati tekstova za korpus, kao i specifičnosti koje prate njihovu konverziju u format čistog teksta.

TXT U najjednostavnijem slučaju tekst prikupljen za korpus je već u formatu čistog teksta. Pre dalje obrade je potrebno još proveriti da li je tekst u kodnom rasporedu UTF-8 i, ako nije, generisati datoteku sa kopijom teksta u kodnom rasporedu UTF-8. U slučaju kada se obrađuje pojedinačni tekst, za konverziju kodnog rasporeda u UTF-8 mogu se iskoristiti razni programi za uređivanje teksta, od onih jednostavnih koji se isporučuju sa operativnim sistemom (na primer Notepad pod operativnim sistemom Windows), do onih sa naprednijim mogućnostima (PSPad[13], Notepad++[14], itd.). Programi sa navedenim mogućnostima se međusobno razlikuju prema broju kodnih rasporeda koje podržavaju. Iako većina njih ima višejezičku podršku, pre svega kroz mogućnost lokalizacije softvera, podržane kodne rasporede uglavnom ograničava na kodne rasporede CP-1252, UTF-8, UTF-16 LE, UTF-16 BE[15].

[13] `http://www.pspad.com`

[14] `http://notepad-plus-plus.org`

[15] CP-1252, UTF-16 LE i UTF-16 BE se u programima za uređivanje teksta obično označavaju kao *ANSI*, *Unicode* i *Unicode big endian*, tim redom, tipičan primer je Notepad.

Izuzetak u pozitivnom smislu predstavlja Microsoft Word koji omogućava ulazno-izlaznu konverziju dokumenta u format čistog teksta sa proizvoljnim kodnim rasporedom (Slika 6.2).

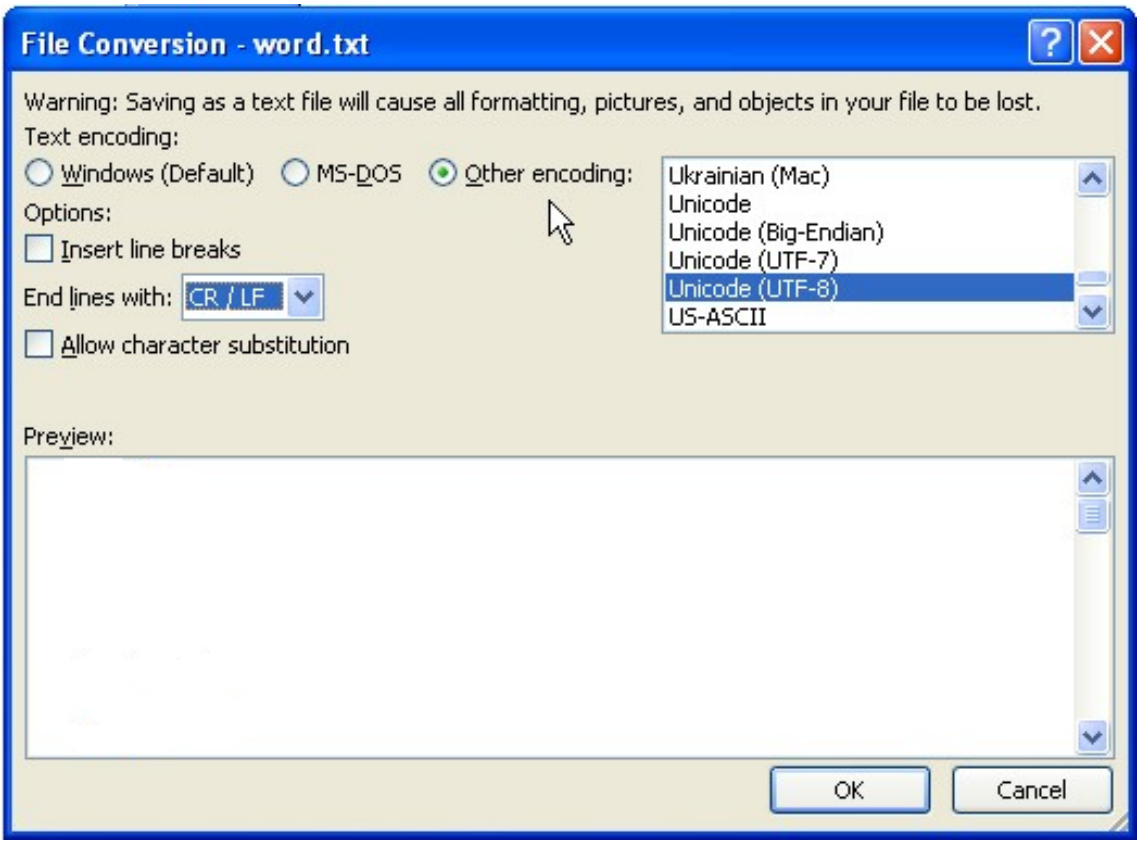
Kada je u pitanju paketna konverzija kodnog rasporeda u UTF-8, većina programa za uređivanje teksta nema tu mogućnost. Izuzetak su programi koji mogu da „snime" niz pojedinačnih komandi korisnika kao jednu složenu komandu ili **makro** (eng. **macro**), tj. poseduju interni programski jezik na kome se mogu specifikovati akcije koje inače zadaje korisnik. Tipičan primer programa koji koristi makroe je Microsoft Word koji omogućava programiranje akcija korisnika na programskom jeziku VBA[16]. Primenom makroa se, na primer, svaka datoteka zadatog kataloga može otvoriti, snimiti u zadatom kodnom rasporedu i zatvoriti. Primer paketne konverzije u UTF-8 pomoću VBA-makroa u programu Microsoft Word je opisan u Dodatku E.1. Nedostatak ovog makroa je što pravilno konvertuje samo tekstove čiji se kodni raspored koristi za karakterski skup Unicode (UTF-8, UTF-16 LE, UTF-16 BE), dok tekstove sa proizvoljnim osmobitnim kodnim rasporedom tretira kao da su tekstovi sa kodnim rasporedom CP-1252 (ANSI).

Za automatsku paketnu konverziju kodnog rasporeda korpusnih tekstova koristi se modul Conversion programa CorpusPreprocessor (v. pododeljak CorpusPreprocessor, str. 270). Podržani su svi kodni rasporedi koji se koriste za kreiranje elektronskog teksta na srpskom jeziku (ISO-646 IRV, ISO 8859-2, ISO 8859-5, CP-1250, CP-1251, UTF-16 LE, UTF-16 BE, UTF-8). Pri paketnoj konverziji kodnog rasporeda modulom Conversion bitan preduslov je da svi dokumenti koje treba zajedno obraditi imaju isti zadati ulazni kodni raspored, i svi ulazni dokumenti se konvertuju u isti zadati izlazni kodni raspored.

DOC i DOCX Format DOCX je uvek konvertovan u DOC, bilo direktno preko Microsoft Word-a (verzija MS Word 2007 i kasnije), bilo posredno uz pomoć programa *FileFormatConverters.exe*[17] koji omogućava starijim verzijama programa (pre svega verziji MS Word 2003) da otvore datoteke u formatu DOCX tako što ih kon-

[16] Skr. od Visual Basic for Applications.

[17] `MicrosoftOfficeCompatibilityPackforWord,Excel,andPowerPointFileFormats,http://www.microsoft.com/en-us/download/details.aspx?id=3`.



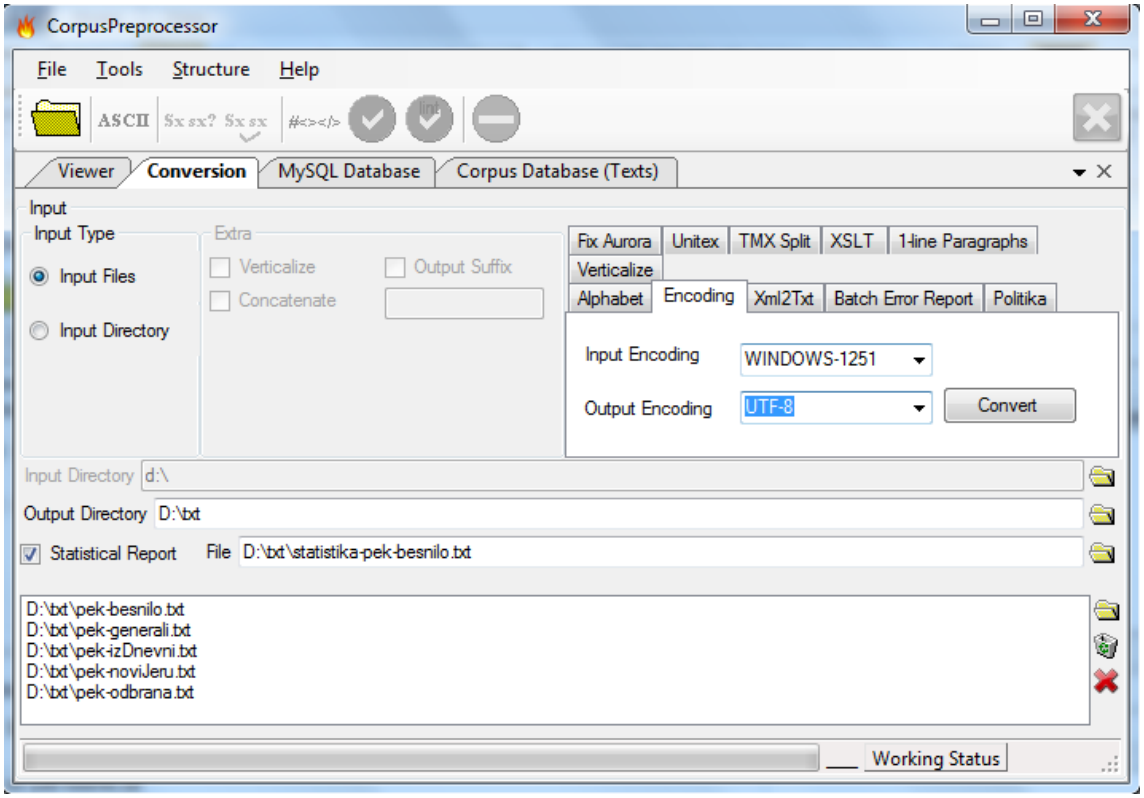
Slika 6.2: Microsoft Word: dijalog za izbor kodnog rasporeda pri snimanju dokumenta u formatu čistog teksta.

vertuju u format DOC. Datoteke u formatu DOC su konvertovane u format čistog teksta korišćenjem već pomenutog makroa opisanog u Dodatku E.1.

HTML, XHTML i XML Konverzija datoteka u formatu HTML, XHTML i XML se odvijala zavisno od toga da li su u polaznom dokumentu korišćene etikete za označavanje logičke strukture teksta, pre svega (pod)naslova i pasusa. U slučaju da takve etikete nisu postojale, polazni dokument je konvertovan u format čistog teksta automatskim paketnim brisanjem etiketa za šta je korišćen modul Conversion programa CorpusPreprocessor (v. pododeljak CorpusPreprocessor, str. 270).

Ukoliko su postojale etikete za označavanje logičke strukture teksta (naslovi i pasusi), obrada je zavisila od konkretnog teksta, kao i od toga da li postoji još tekstova sa istovetnom strukturom (HTML), odnosno koji koriste istu XML-šemu[18]

[18]Videti odeljak 2.4, str. 126.



Slika 6.3: CorpusPreprocessor, modul Conversion: paketna konverzija kodnog rasporeda.

(XHTML i XML). U slučaju kada je postojalo više tekstova u formatu XHTML ili XML sa istom XML-šemom, najpre je kreirana odgovarajuća XSL-transformacija kojom su polazni dokumenti konvertovani u TEI/XML-dokumente paketnom obradom u okviru programa CorpusPreprocessor (v. pododeljak CorpusPreprocessor, str. 270). HTML-dokumenti sa istovetnom strukturom su najpre konvertovani u XHTML[19], a potom XSL-transformacijom u TEI/XML. U slučaju tekstova sa jedinstvenom strukturom u nekom od formata HTML, XHTML, XML, obrada je zavisila od težine konverzije, tako da je ponekad rađena ručno, a ponekad je kreirana odgovarajuća XSL-transformacija za konverziju u format TEI/XML uz prethodnu konverziju formata HTML u XHTML.

[19]Konverzija formata HTML u XHTML se takođe može obaviti paketnom obradom pomoću modula Conversion programa CorpusPreprocessor (v. pododeljak CorpusPreprocessor, str. 270). Tokom konverzije pojedinačne datoteke CorpusPreprocessor poziva jedan od tri eksterna programa koji bira korisnik pre početka paketne obrade: *xmllint* (`http://xmlsoft.org`), *HTML Tidy* (`http://tidy.sourceforge.net`) ili *html2xhtml* (`http://www.it.uc3m.es/~jaf/html2xhtml/`).

PDF Od svih navedenih formata, konverzija formata PDF u format čistog teksta je svakako najsloženija. Razlog je način na koji se tekst čuva u okviru PDF-dokumenta. Pored toga što tekst u okviru PDF-dokumenta ne mora biti u istom redosledu kao u njegovoj vizuelnoj reprezentaciji na ekranu i može koristiti fontove koje operativni sistem ne poseduje, PDF-dokument koji vizuelno izgleda kao tekst, može zapravo biti digitalna slika. Zato je u praksi bilo izuzetno teško naći program koji može da konvertuje proizvoljan PDF-dokument u format čistog teksta[20], a da se pri tom dobije tekst takav:

- da je u istom redosledu kao u polaznom dokumentu;
- da, bez obzira na korišćeni font u polaznom dokumentu, vizuelni izgled bude ekvivalentan polaznom, tj. da odgovarajući glifovi polaznog i konvertovanog dokumenta odgovaraju istim apstraktnim karakterima (v. odeljak 2.2, str. 71);
- da nije „iseckan", tj. da posle konverzije slogovi tekstuelnih reči nisu razdvojeni razmacima ili crticama za rastavljanje reči na kraju retka.

Neretko se ispostavljalo da najbolje rezultate pri konverziji u format čistog teksta daje obično kopiranje teksta iz programa za pregled PDF-dokumenata (najčešće Adobe Reader[21]) u neki program za uređivanje teksta (najčešće Notepad ili PSPad[22]). Zato su PDF-dokumenti najčešće konvertovani pojedinačno i uz nadzor čoveka, a manji broj je konvertovan automatski, paketnom obradom.

EPUB, DJVU, MOBI Tekstovi su retko prikupljeni u nekom od formata EPUB, DJVU, MOBI, itd., s obzirom da kad god su bili dostupni u ovim formatima, uglavnom je postojala i verzija teksta u nekom od prethodno navedenih formata (DOC, DOCX, (X)HTML, XML, PDF).

[20] Na adresi `http://arhimed.matf.bg.ac.rs/~misko/flf/dl/zaduzenja/pdf2txt/pdf2txt-uporedna.html` je dostupan uporedni prikaz alata za paketnu konverziju PDF-dokumenata koji su pripremile Zvezdana Stojkanović i Mirjana Nešić u sklopu predispitnih obaveza za predmet *Elektronsko izdavaštvo i digitalne biblioteke* tokom školske 2010/2011. godine.

[21]

[22] `http://www.pspad.com`

Kodna shema aurora

Da bi se neutralizovao uticaj različitih pisama (ćirilica, latinica) i kodnih shema (YUSCII, ISO 8859-2, ISO 8859-5, CP-1250, CP-1251, UTF-16, UTF-8, itd.) koji se koriste za predstavljanje elektronskog teksta na srpskom jeziku, tekstovi korpusa su kodirani internim kodom *aurora*. Aurora je dobila naziv po istoimenom programskom sistemu Duška Vitasa u kome je prvi put primenjena ([Vitas, 1981]). Grupa za jezičke tehnologije Univerziteta u Beogradu koristi auroru kao interni kôd u sledećim jezičkim resursima za srpski jezik koje razvija:

- elektronski morfološki rečnici srpskog jezika u formatima LADL/DELA ([Vitas, 1993], Krstev [2008]), MULTEXT-East ([Krstev et al., 2004]) i NooJ ([Stanković et al., 2011]);
- leksička baza podataka WordNet ([Krstev et al., 2003], [Obradović et al., 2004], [Koeva et al., 2008]);
- semantička mreža vlastitih imena Prolex ([Krstev, Vitas, Maurel & Tran, 2005], [Tran et al., 2005], [Vitas et al., 2007], [Maurel et al., 2007]);
- jednojezični i paralelizovani korpusi ([Krstev & Vitas, 2005], [Vitas & Krstev, 2006], [Vitas, 2010]), itd.

Aurora je realizovana korišćenjem karakterskog skupa ASCII u kome nema karaktera specifičnih za srpski jezik (slova sa dijakriticima, latinični digrafi), kao i uvođenjem zapisa slova koji omogućava predstavljanje i tih specifičnih karaktera tako da se postigne jednoznačno razlikovanje digrafa od konsonantskih grupa (Tabela 6.3).

U trenutku kada je aurora nastala nije postojao nijedan od kodnih rasporeda koji danas mogu da se koriste za računarsku reprezentaciju teksta na srpskom jeziku (YUSCII, ISO 8859-2, ISO 8859-5, CP-1250, CP-1251, UTF-16, UTF-8). Posle više od tri decenije, pogotovo pojavom Unicode-a, postavlja se pitanje: da li je aurora i dalje najpogodniji interni kôd jezičkih resursa za srpski jezik? Najčešće navođeni nedostaci aurore su:

Tabela 6.3: Kodna shema aurora

Dijakritički karakteri					
Veliko slovo			**Malo slovo**		
ćirilica	**latinica**	**aurora**	**ćirilica**	**latinica**	**aurora**
Ђ	Đ	Dx	ђ	đ	dx
Ж	Ž	Zx	ж	ž	zx
Ћ	Ć	Cx	ћ	ć	cx
Ч	Č	Cy	ч	č	cy
Ш	Š	Sx	ш	š	sx
Digrafi					
Veliko slovo			**Malo slovo**		
ćirilica	**latinica**	**aurora**	**ćirilica**	**latinica**	**aurora**
Љ	LJ, Lj	Lx	љ	lj	lx
Њ	NJ, Nj	Nx	њ	nj	nx
Џ	DŽ, Dž	Dy	џ	dž	dy

- korišćenje digrafa, tj. predstavljanje slova specifičnih za srpski jezik pomoću dva karaktera, iako se mogu predstaviti jednim Unicode-karakterom (na primer `Š` i `Sx`);

- grafemske dvosmislice, tj. u tekstu koji je kodiran aurorom pojedini digrafi, na primer `dx`, mogu da se tumače i kao slovo đ, odnosno ђ, ali i kao zapis rimskog broja (u ovom primeru kao rimski broj čija je vrednost 510).

Dva su bitna razloga zbog kojih je aurora i dalje najpogodniji interni kôd jezičkih resursa za srpski jezik:

- aurora neutrališe uticaj različitih pisama (ćirilica, latinica) u tekstu;

- na tekst kodiran aurorom se uvek može primeniti kodna shema ASCII, što je posebno važno prilikom automatske obrade teksta. Naime, iako je Unicode prisutan skoro dve decenije, a kodni rasporedi ISO 8859 i Microsoft kodne strane i duže (v. odeljak 2.2), ostaje činjenica da mnogi korisni alati za obradu teksta i dalje ne podržavaju Unicode, već samo ASCII ili u najboljem slučaju ISO 8859-1, odnosno CP-1250[23].

U okviru programa CorpusPreprocessor (v. pododeljak CorpusPreprocessor, str. 270) postoji modul za automatsku paketnu konverziju alfabeta korpusnih tekstova u kôd

[23]Primer takvih alata je podskup paketa *coreutils*, odnosno *GNU Core Utilities* (`cat`, `comm`, `cut`, `join`, `paste`, `sort`, `tr`, `uniq`, `wc`, itd.) koji koriste operativni sistemi GNU/Linux, GNU/*BSD.

aurora. Ovaj modul podržava i međusobnu konverziju između ćiriličnog i latiničnog pisma, kao i međusobnu konverziju između tih pisama i kodnih shema YUSCII-latinica, YUSCII-ćirilica i aurora. U izradi modula su iskorišćeni prilagođeni otvoreni projekti *SrConvNet* Miloša Babovića za Visual Basic i C# 2005/2008, koji sadrže funkcije za transliteraciju pisama srpskog jezika za .NET[24]. Tekstovi dobijeni kao rezultat konverzije u kôd aurora se uvek generišu sa kodnim rasporedom UTF-8.

S obzirom da se u polaznom tekstu mogu koristiti i ćirilično i latinično pismo, konverzija se obavlja u dva koraka:

1. konverzija pisma polaznog teksta iz ćirilice u latinicu i kodnu shemu UTF-8;

2. konverzija latinične verzije teksta u kôd aurora i kodnu shemu UTF-8.

Pošto ćirilica nema digrafa, automatska konverzija u auroru je jednoznačna. Kad je latinica u pitanju, jednoznačnost konverzije u auroru nije zagarantovana u slučaju digrafa jer neki od njih mogu predstavljati jedno slovo (`dž` u `džem`, `nj` u `njiva`, `Nj` u `Njujork`) ili dva (`dž` u `nadživeti`, `nj` u `injekcija`, `NJ` u `TANJUG`), što dovodi do različitih rezultata konverzije (`dyem` i `nadzxiveti`, odnosno `nxiva` i `injekcija`, odnosno `Nxujork` i `TANJUG`). Uvidom u elektronski morfološki rečnik srpskog jezika se ispostavlja da je slučaj kada latinični digraf predstavlja dva slova daleko ređi, tj. da lema sa takvim digrafima ima svega nekoliko desetina, pri čemu su pojedine među njima izvedenice čiji zajednički koren sadrži digraf (`konjunktura`, `konjunkturan`, `konjunkturist`, `konjunkturistički`; `nadživeti`, `nadživljavanje`, `nadživljavan`, `nadživljavati`, itd.). Ukoliko se sastavi lista oblika takvih lema (označimo je sa L), problem nejednoznačne konverzije latiničnih digrafa može da se reši u dva koraka:

1. Konverzijom svih digrafa u tekstu pod pretpostavkom da predstavljaju jedno slovo;

2. Ako je L' lista reči dobijenih konverzijom reči iz liste L u auroru pod pretpostavkom da svaki digraf predstavlja jedno slovo, a L'' lista reči dobijenih

[24]Projekti SrConvNet mogu se besplatno preuzeti sa sajta „Praktikum na Webu" koji uređuje Dragan Grbić, `http://www.praktikum.rs/office/download/SrConvNet.zip`.

korektnom konverzijom reči iz liste L u auroru, tada se u drugom koraku vrši konverzija elemenata liste L' u odgovarajuće elemente liste L''.[25]

6.3 Anotacija

Bibliografska anotacija

Većina bibliografskih informacija o korpusnim tekstovima je prikupljena pretragom **Kooperativnog onlajn bibliografskog sistema i servisa** (skr. **COBISS.SR**)[26], odnosno **Uzajamne bibliografsko-kataloške baze podataka** (skr. **COBIB.SR**), dok se do bibliografskih informacija preostalih korpusnih tekstova došlo uvidom u njihova papirna izdanja koja su poslužila kao izvorni tekst za proces digitalizacije.

Sve bibliografske informacije o korpusnim tekstovima se nalaze u tabelama relacione baze podataka (Slika 6.4):

- `texts` (primarni ključ tabele je kolona `tid`, ceo pozitivan broj, identifikator teksta);
- `authors` (primarni ključ tabele je kolona `author_id`, ceo pozitivan broj, identifikator autora);
- `text_styles` (primarni ključ tabele je kolona `styleid`, ceo pozitivan broj, identifikator funkcionalnog stila);
- `text_sourcedesc` (primarni ključ tabele je kolona `sourcedescid`, ceo pozitivan broj, identifikator statusa teksta u odnosu na jezik na kom je nastao).

Ostale kolone tabela čuvaju sledeće bibliografske informacije:

- naziv datoteke korpusnog teksta (`filename`);
- naslov korpusnog teksta (`title`);

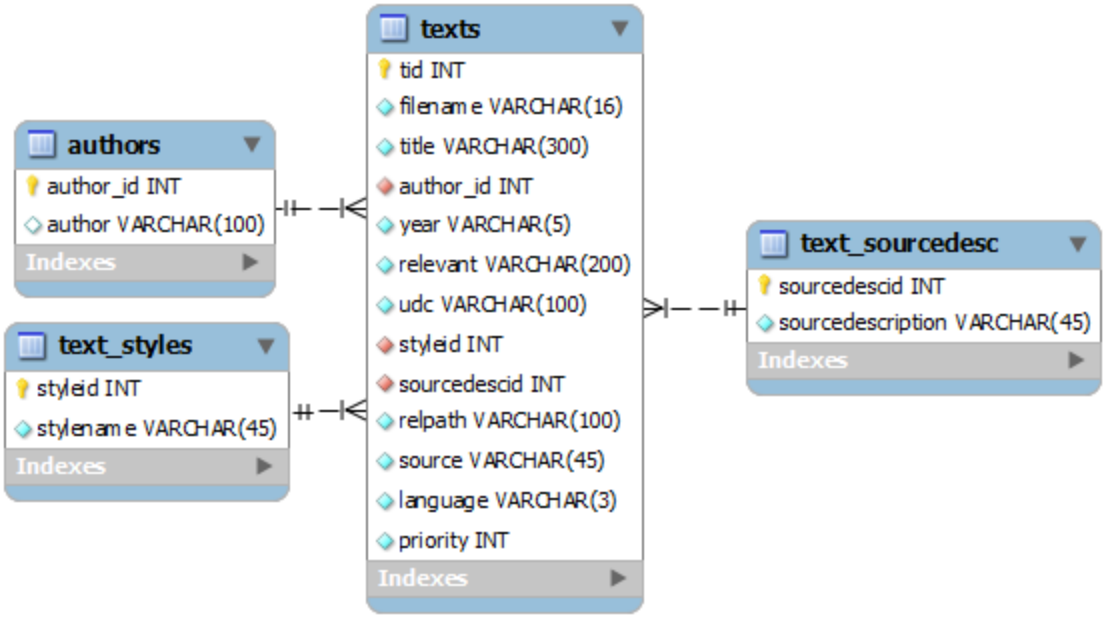
[25] Na primer, u prvom koraku se `nadživljavan` konvertuje u `nadyivlxavan`, a u drugom se `nadyivlxavan` konvertuje u `nadzxivlxavan`.

[26] `http://www.vbs.rs/cobiss/index-sc.html`

- godinu izdanja korpusnog teksta (`year`);
- relevantne podatke o izdavaču (naziv, adresa) i eventualnom prevodiocu korpusnog teksta (`relevant`);
- UDK korpusnog teksta (`udc`);
- relativnu adresu datoteke korpusnog teksta u repozitorijumu korpusnih tekstova (`relpath`);
- napomena o izvoru iz kog je potekla elektronska verzija korpusnog teksta (`source`);
- jezik korpusnog teksta (`language`);
- prioritet korpusnog teksta prilikom određivanja redosleda tekstova u SrpKor-u (`priority`);
- ime i prezime autora (`author`);
- naziv funkcionalnog stila (`stylename`);
- status teksta u odnosu na jezik na kom je nastao, tj. da li je u pitanju tekst čiji je original nastao na srpskom jeziku ili je u pitanju prevod na srpski jezik (`sourcedescription`).

Iako je dovoljno da token korpusa sadrži samo informaciju o identifikatoru teksta iz kog potiče, radi efikasnije pretrage su tokenima pridružene sledeće bibliografske informacije iz tabele `texts`:

- identifikator teksta,
- identifikator autora,
- godina izdanja,
- identifikator funkcionalnog stila i
- identifikator koji ukazuje da li je tekst u originalu nastao na srpskom jeziku ili je u pitanju prevod.



Slika 6.4: Dijagram tabela relacione baze podataka sa bibliografskim informacijama o tekstovima SrpKor-a. Dijagram je realizovan u programu MySQL Workbench (`http://www.mysql.com/products/workbench/`).

Naime, kada konkordancer generiše rezultat na osnovu zadatog upita, pre nego što se taj rezultat prikaže korisniku, odgovarajuće bibliografske informacije se preuzimaju iz tabele `texts` na osnovu identifikatora teksta pridruženog ključnim rečima i priključuju prikazu konkordanci. Međutim, kada korisnik želi da zada upit kojim se konkordance filtriraju po imenu autora, godini izdanja, funkcionalnom stilu ili statusu teksta u odnosu na jezik na kom je nastao (srpski u originalu ili prevod na srpski), a tokeni korpusa sadrže samo informaciju o identifikatoru teksta, upitni jezik nije dovoljno izražajan za takav upit.

Razmotrimo, na primer, da korisnik želi da pronađe sva pojavljivanja leme `most` i da pretraži samo tekstove Ive Andrića, pri čemu je broj 1 identifikator tog autora. Ukoliko tokeni sadrže samo informaciju o identifikatoru teksta, jedino rešenje je da se najpre prosledi SQL-upit relacionoj bazi podataka koji će kao rezultat vratiti sve identifikatore teksta (`tid`) čiji je autor Ivo Andrić (`authorid = 1`):

```
SELECT tid
FROM texts
```

```
WHERE authorid = 1;
```

Ukoliko bi rezultat SQL-upita bili parni celi brojevi između 10 i 20 (ne uključujući ta dva broja), veb-sučelje sistema za pretragu bi moralo da generiše sledeći upit i prosledi ga konkordanceru:[27]

```
[lemma="most" & tid="12" & tid="14" & tid="16" & tid="18"];
```

U ovom slučaju generisani upit nije previše složen zato što rezultat SQL-upita sadrži svega četiri identifikatora teksta. Međutim, da je kojim slučajem veličina rezultata SQL-upita nekoliko hiljada identifikatora teksta (na primer, kada korisnik želi da pretražuje samo novinske tekstove), generisani upit bi bio izuzetno složen i pitanje je da li bi ga konkordancer uopšte prihvatio, s obzirom na njegovu dužinu.

Pošto se izražajnost upitnog jezika trenutno ne može popraviti, jedino rešenje je da se tokeni dodatno anotiraju informacijama koje omogućavaju direktnu proveru da li su kriterijumi zadatog filtera u upitu ispunjeni ili ne. Time se veličina prostora koji korpus zauzima na memorijskom medijumu povećava za svaki novi element anotacije približno onoliko koliko zauzimaju sami tokeni korpusa.

Ako je token anotiran informacijom o identifikatoru autora, upit iz prethodnog primera se može zameniti sa:

```
[lemma="most" & authorid="1"];
```

Strukturna anotacija (primena specifikacije TEI-Lite)

Jedan deo korpusnih tekstova od kojih je sačinjen SrpKor je strukturno anotiran, tj. čuva se u formatu TEI/XML. Prvi strukturno anotirani tekstovi su predstavljali materijal za paralelizaciju, odnosno izgradnju paralelnih korpusa u kojima je srpski izvorni ili ciljni jezik (v. odeljak 6.5, str. 294). S obzirom da je bilo dovoljno da ulaz za korišćene alate za paralelizaciju predstavljaju samo dobro formirani XML-dokumenti, a ne istovremeno i validni u odnosu na neku XML-šemu, prvi strukturno anotirani tekstovi nisu bili validni u odnosu na XML-šemu koja opisuje TEI-Lite, standardizovani podskup najčešćih ili najvažnijih TEI-elemenata i atributa. Tek su

[27] Detaljan opis sintakse i semantike upitnog jezika CQL je dat u poglavlju 7.

kasniji tekstovi, pogotovo tokom pripreme korpusa SrpKor2011, strukturno anotirani tako da budu validni u odnosu na XML-šemu koja opisuje TEI-Lite P5, konkretno *teilite.dtd*[28]. Osim etiketa za strukturnu anotaciju teksta (Tabela 6.4), DTD-šema za TEI-Lite P5 opisuje i etikete za bibliografsku anotaciju teksta (Tabela 6.5).

Tabela 6.4: Podskup najčešće korišćenih TEI-elemenata i atributa za strukturnu anotaciju tekstova SrpKor-a. Elementi zaglavlja `teiHeader` su prikazani u posebnoj tabeli (Tabela 6.5).

TEI-element	**objašnjenje**	**TEI-atributi**	**vrednosti atributa**	
`TEI`	koreni element TEI-dokumenta			
`teiHeader`	zaglavlje TEI-dokumenta (bibliografski podaci)			
`text`	jedan tekst proizvoljne vrste (na primer roman, uključujući predgovor i pogovor)			
`body`	telo teksta (na primer tekst romana bez predgovora i pogovora)			
`div`	jedinica teksta (na primer deo ili poglavlje romana, članak i odeljci članka, itd.)			
		`type`	book, foreword, part, chapter, article, section, ...	tip jedinice teksta (knjiga, predgovor, deo, poglavlje, članak, odeljak, itd.)
		`n`	pozitivan ceo broj (1, 2, 3, ...)	redni broj jedinice teksta (na primer poglavlje 1, odeljak 1.2, itd.)
`head`	naslov i podnaslovi			
`p`	pasus			
`seg`	segment (obično rečenica)			
`pb`	završetak stranice			
		`n`	pozitivan ceo broj (1, 2, 3, ...)	redni broj stranice
`note`	napomena (na primer, fusnota)			
		`n`	pozitivan ceo broj (1, 2, 3, ...)	redni broj napomene

[28]Primer takve datoteke se može naći na adresi `http://www.tei-c.org/release/xml/tei/custom/schema/dtd/tei_lite.dtd`. U praksi korisnici mogu da konstruišu svoju sopstvenu verziju te datoteke, koristeći alat Roma (`http://www.tei-c.org/Roma/`).

S obzirom da kompletan opis TEI-elemenata i atributa za strukturnu i bibliografsku anotaciju tekstova izlazi iz okvira ovog rada, za detalje pogledati [Burnard & Sperberg-McQueen, 2012] i primer u Dodatku B.

Tabela 6.5: Podskup najčešće korišćenih TEI-elemenata i atributa za bibliografsku anotaciju tekstova SrpKor-a

TEI-element	**objašnjenje**
`teiHeader`	zaglavlje TEI-dokumenta (bibliografski podaci)
`fileDesc`	kompletan bibliografski opis elektronske verzije teksta
`titleStmt`	grupisanje informacija o naslovu i autorima teksta (elektronske ili izvorne verzije)
`publicationStmt`	grupisanje informacija o izdavanju i distribuciji teksta (elektronske ili izvorne verzije)
`sourceDesc`	opis izvora na osnovu kojeg je nastala elektronska verzija teksta
`respStmt`	izjava kojom se utvrđuje odgovornost za intelektualnu sadržinu teksta, različita izdanja, kreiranje elektronske verzije teksta, korekcije grešaka, itd.

Morfološka anotacija SrpKor-a

Da bi se tekstovi SrpKor-a morfološki anotirali, neophodno je (v. odeljak 2.4, str. 140):

- precizno definisati skup morfoloških deskriptora (etiketa) koji se pridružuju pojedinačnim tokenima;
- izabrati programski **alat za automatsko etiketiranje vrstom reči** (eng. **Part of Speech tagger**, skr. **PoS-tagger**);
- pripremiti pomoćne resurse na osnovu kojih se alat za automatsko etiketiranje vrstom reči obučava za rad, odnosno koje koristi pri samom etiketiranju.

Skup morfoloških deskriptora (etiketa) Za formiranje skupa morfoloških deskriptora (etiketa) za anotaciju SrpKor-a (u daljem tekstu: skup etiketa) je iskorišćena postojeća morfosintaksička notacija primenjena u elektronskom morfološkom

rečniku srpskog jezika u formatu LADL/DELA (v. odeljak 2.4, str. 145). Prilikom formiranja skupa etiketa se mora voditi računa o tome da veličina skupa etiketa, tj. količina informacija koju etikete sadrže, značajno utiče na preciznost anotacije, odnosno da se povećanjem skupa etiketa otežava precizno pridruživanje etiketa i obrnuto. Zato je izabran bazični skup etiketa, sastavljen, pre svega, od deset oznaka različitih vrsta reči u srpskom jeziku, kao i šest dodatnih etiketa namenjenih za anotaciju nekih specifičnih tokena koji zahtevaju posebnu obradu:

1. N (imenica),
2. PRO (zamenica),
3. A (pridev),
4. NUM (broj),
5. V (glagol),
6. ADV (prilog),
7. PREP (predlog),
8. CONJ (veznik),
9. INT (uzvik),
10. PAR (rečca ili partikula),
11. ABB (skraćenica)
12. PREF (prefiks)
13. RN (rimski broj)
14. PUNCT (znak interpunkcije)
15. SENT (oznaka kraja rečenice)
16. ? (oznaka za ostalo: strane reči u tekstu, sufikse poput `ve` u `1971-ve`, `og` u `25-og`, itd.).

Izbor alata za automatsko etiketiranje vrstom reči Od raspoloživih alata za automatsko etiketiranje vrstom reči, detaljno su razmotrena tri:

- Unitex (v. odeljak 4.3, str. 216),
- TnT ([Brants, 2000]) i
- TreeTagger ([Schmid, 1994]).

S obzirom da izbor konkretnog alata za automatsko etiketiranje vrstom reči utiče na način na koji je potrebno adaptirati postojeće morfosintaksičke opise za srpski jezik, Unitex je predstavljao prvi izbor, pošto koristi isti format (LADL/DELA) kao i elektronski morfološki rečnik srpskog jezika. Pored formata LADL/DELA, Unitex ima i prednost zato što tokom etiketiranja pridružuje korpusnim rečima isključivo one leksičke interpretacije koje postoje u rečniku.

U trenutku kada je donošena odluka o izboru alata za etiketiranje vrstom reči, Unitex je imao na raspolaganju samo jedan mehanizam za uklanjanje višeznačnosti prilikom automatske morfološke anotacije teksta, a u međuvremenu je razvio još jedan, zasnovan na mašinskom učenju ([Paumier, 2011]).

U prvom slučaju, Unitex se oslanja na *automat teksta* (eng. *text automaton*) koji se sastoji od segmenata (najčešće rečenica) i svih mogućih leksičkih interpretacija korpusnih reči u okviru svakog segmenta. Svaki segment automata teksta se može predstaviti grafom, pri čemu svaki put od početnog do završnog čvora u grafu opisuje jedno moguće etiketiranje korpusnih reči segmenta. Prvi mehanizam koji koristi Unitex je formalizam ELAG ([Laporte & Monceaux, 1998]), zasnovan na ručno kreiranim pravilima za etiketiranje, implementiranim u obliku automata. ELAG razrešava višeznačnost tako što specifikuje dozvoljeni ili zabranjeni kontekst korpusne reči etiketirane na određeni način.

Međutim, dva ozbiljna nedostatka nisu presudila u korist Unitex-a. Prvi je značajan napor potreban kako bi se kreirale gramatike formalizma ELAG koje bi postigle bar približnu preciznost kao i programi za etiketiranje zasnovani na mašinskom učenju (poput TnT-a i TreeTagger-a). Drugi razlog je što se razvoj ELAG-gramatika za srpski jezik još uvek nalazi u početnoj fazi.

Popović (2008), koristeći različite alata za etiketiranje vrstom reči, evaluira morfosintaksičku anotaciju pomoću unakrsnog testa (v. odeljak 2.4, pri čemu se koristi skup od 908 etiketa - morfosintaksičkih opisa za srpski jezik definisanih u okviru projekta MULTEXT-East[29] ([Erjavec, Krstev, Petkevič, Simov, Tadić & Vitas, 2003]) i korpus veličine 105 hiljada korpusnih reči (18 hiljada korpusnih tipova i 7,6 hiljada lema). Tom prilikom se pokazuje da TnT i TreeTagger postižu najbolje rezultate nad istim korpusima za obuku (TnT: 93,86%, TreeTagger: 91,78%), pri čemu se TnT bolje pokazao u etiketiranju nep(rep)oznatih) reči (TnT: 58,36%, TreeTagger: 36,71%).

Pošto TnT nema mogućnost da korpusnim rečima pridruži informaciju o lemi, na kraju je TreeTagger izabran kao alat za automatsku morfološku anotaciju tekstova SrpKor-a. Detalji o pripremi TreeTagger-a za anotaciju tekstova na srpskom jeziku se mogu naći u [Utvić, 2011].

6.4 Indeksiranje teksta

U odeljku 4.3 je dat uporedni prikaz sistema integrisanih korpusnih alata. Uzimajući u obzir sve izložene karakteristike prikazanih sistema, u uži izbor za sistem kojim će se indeksirati i pretraživati SrpKor su ušli IMS OCWB i grupacija sistema SketchEngine i NoSketchEngine (Manatee i Bonito). Pošto Grupa za jezičke tehnologije podržava korišćenje slobodnog softvera, SketchEngine je potisnut na treće mesto, tako da se, u stvari, biralo između sistema IMS OCWB i sistema NoSketchEngine. Zapravo, u trenutku kada je donošena odluka, NoSketchEngine nije postojao pod tim imenom, već kao skup dva alata, Manatee i Bonito, pri čemu je Bonito korišćen kao samostalna klijentska aplikaciju na programskom jeziku Tcl/Tk. S obzirom da je Bonito koristio portu (eng. port) koja je bila zabranjena u okviru ogranka Akademske mreže Srbije (skr. AMRES) na Univerzitetu u Beogradu, izbor je na kraju pao na IMS OCWB kojim je realizovana i prva verzija SrpKor-a[30].

[29] `http://nl.ijs.si/me/V3/msd/html/`

[30] U međuvremenu je stara verzija programa Bonito zamenjena novom koja koristi veb sučelje generisano na programskom jeziku Python i koja će, zbog svojih naprednih mogućnosti, svakako biti kandidat za sistem korpusnih alata u izgradnji sledeće verzije SrpKor-a.

Da bi se tekst indeksirao pomoću sistema IMS OCWB, neophodno je da tekst bude konvertovan iz formata za čuvanje u format za indeksiranje. Format za indeksiranje koji koristi IMS OCWB je tipičan primer vertikalnog formata teksta, pri čemu svaka kolona odgovara jednom pozicionom atributu (v. odeljak 2.5). U prvoj koloni vertikalnog formata se nalaze tokeni korpusa, u svakoj liniji po jedan, te se prvi pozicioni atribut naziva *korpusna reč* (eng. *word*). Ostali atributi, ako postoje, se koriste kao elementi anotacije. U slučaju SrpKor-a, vertikalni format za indeksiranje korpusnih tekstova sadrži ukupno osam pozicionih atributa (Slika 6.5):

1. `word` (token, odnosno korpusna reč),
2. `tid` (identifikator korpusnog teksta),
3. `authorid` (identifikator autora teksta),
4. `year` (godina izdanja),
5. `styleid` (identifikator funkcionalnog stila),
6. `sourcedescid` (identifikator statusa teksta u odnosu na jezik na kom je nastao),
7. `pos` (vrsta reči) i
8. `lemma` (lema).

Kada su korpusni tekstovi konvertovani u vertikalni format za indeksiranje, sam proces indeksiranja se realizuje pomoću IMS OCWB-alata `cwb-encode` i `cwb-makeall`, tj. u dva koraka ([Evert & The OCWB Development Team, 2010a]):

1. alatom `cwb-encode` se za svaki pozicioni atribut kreira njegova celobrojna reprezentacija i skup različitih vrednosti pozicionog atributa (leksikon), tj. datoteke `word.corpus`, `word.lexicon` i `word.lexicon.idx` (i analogno za ostale pozicione atribute);
2. alatom `cwb-makeall` se kreiraju invertovani indeksi pozicionih atributa i liste učestanosti njihovih vrednosti, tj. `word.lexicon.srt`, `word.corpus.rdx`,

`word.corpus.rev` i `word.corpus.cnt` (i analogno za ostale pozicione atribute).

word	tid	authorid	year	styleid	sourcedescid	pos	lemma
Nisam	53	88	1999	1	1	V	jesam
poklopac	53	88	1999	1	1	N	poklopac
,	53	88	1999	1	1	PUNCT	,
nego	53	88	1999	1	1	CONJ	nego
ono	53	88	1999	1	1	PRO	onaj
sxto	53	88	1999	1	1	PRO	sxto
jesam	53	88	1999	1	1	V	jesam
,	53	88	1999	1	1	PUNCT	,
zato	53	88	1999	1	1	ADV	zato
za	53	88	1999	1	1	PREP	za
strplxenxe	53	88	1999	1	1	N	strplxenxe
molim	53	88	1999	1	1	V	moliti
,	53	88	1999	1	1	PUNCT	,
da	53	88	1999	1	1	CONJ	da
se	53	88	1999	1	1	PAR	se
procyita	53	88	1999	1	1	V	procyitati
red	53	88	1999	1	1	N	red
po	53	88	1999	1	1	PREP	po
red	53	88	1999	1	1	N	red
.	53	88	1999	1	1	SENT	.

Slika 6.5: Vertikalni format za indeksiranje korpusnog teksta SrpKor-a (Antonije Isaković, *Molba iz 1950*). Prvi red je naveden samo radi objašnjenja značenja pojedinih kolona, inače nije deo vertikalizovanog korpusnog teksta.

6.5 Priprema paralelizovanih korpusa

U odeljku 1.3 (str. 29) paralelizovani korpus je definisan kao kolekcija sastavljena od tekstova na izvornom jeziku i njihovih prevoda na jedan ili više ciljnih jezika, u praksi najčešće realizovana kao dvojezični, odnosno skup dvojezičnih paralelizovanih korpusa (ako su tekstovi zastupljeni na više od dva jezika), a ređe kao skup različitih prevoda na jedan ciljni jezik istih tekstova čiji izvorni jezik ili jezici nisu prisutni u korpusu. Tako je, na primer, paralelni bošnjačko-hrvatsko-srpski korpus koji sadrži paralelizovane tekstove Dejtonskog sporazuma iz 1995. godine realizovan kao skup tri dvojezična korpusa (bošnjačko-hrvatskog, hrvatsko-srpskog i bošnjačko-srpskog),

dok je na osnovu dva različita prevoda Volterovog romana Kandid sa francuskog na srpski napravljen srpsko-srpski bitekst.

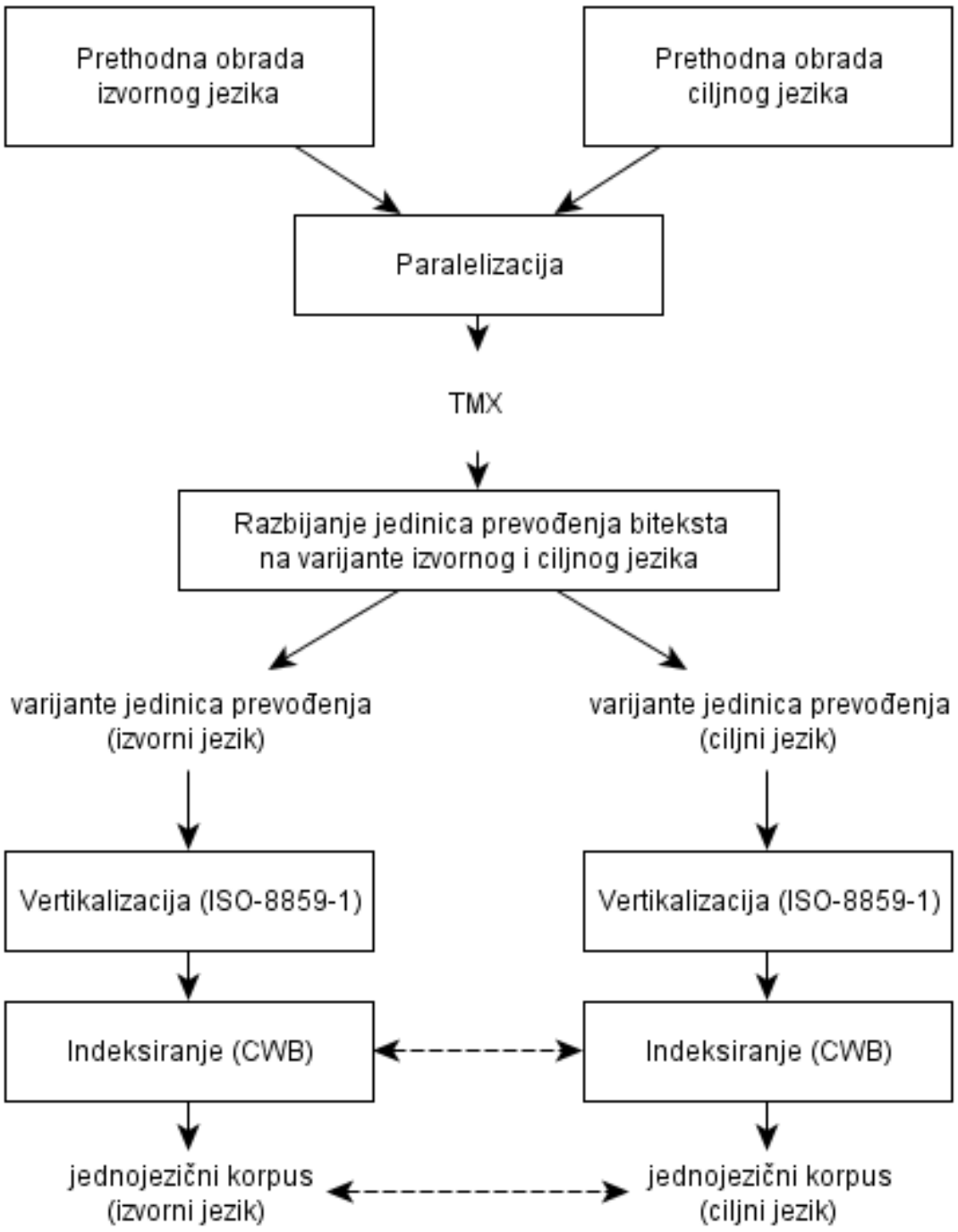
U ovom odeljku biće razmotrene faze u kreiranju dvojezičnog paralelizovanog korpusa u kome srpski jezik predstavlja izvorni ili ciljni jezik tekstova. U daljem opisu, bez umanjenja opštosti, možemo pretpostaviti da je srpski ciljni jezik.

Grupa za jezičke tehnologije Univerziteta u Beogradu primenjuje postupak koji pripremu dvojezičnog paralelizovanog korpusa svodi na pripremu dva jednojezična korpusa, pri čemu se tokom njihovog kreiranja uspostavlja veza između strukturnih elemenata tekstova na izvornom jeziku i odgovarajućih elemenata tekstova na ciljnom jeziku (Slika 6.6). Dvojezični paralelizovani korpus se može posmatrati kao skup **jedinica prevođenja** (eng. **Translation Unit**, skr. **TU**). Svaka jedinica prevođenja dvojezičnog korpusa se sastoji iz dve **varijante jedinice prevođenja** (eng. **Translation Unit Variant**, skr. **TUV**) koje predstavljaju semantički ekvivalentne delove teksta na izvornom i ciljnom jeziku (Slika 6.7).

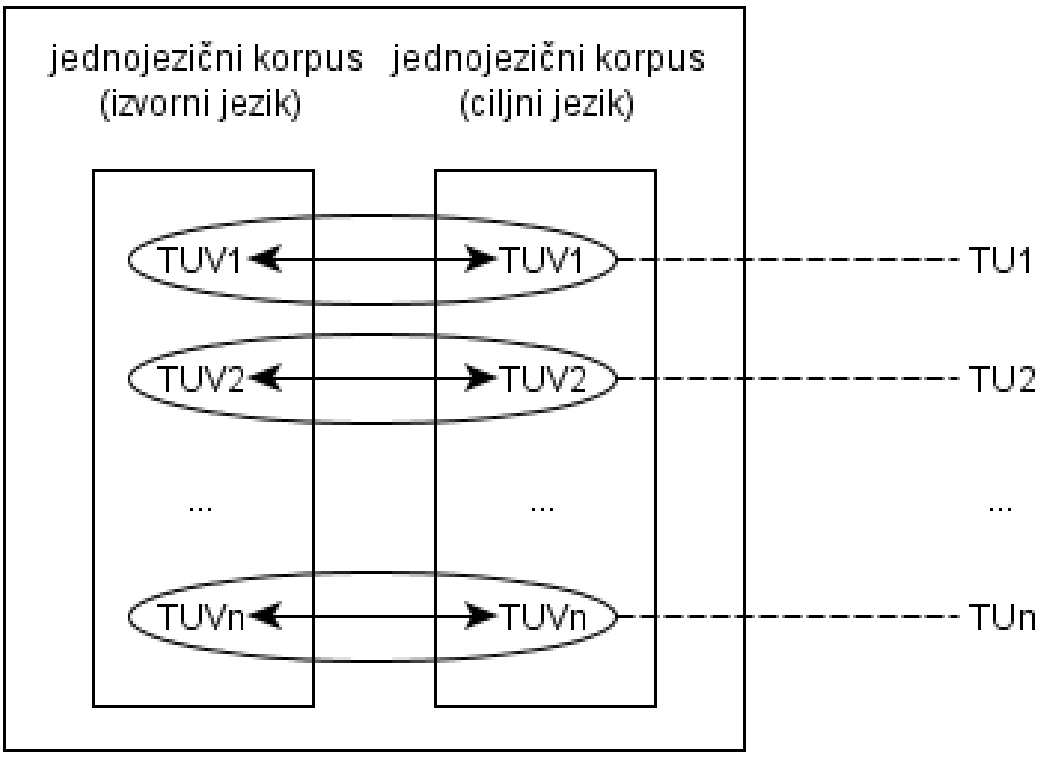
Proces formiranja jedinica prevođenja, tj. uspostavljanje veza između odgovarajućih varijanti jedinica prevođenja naziva se **paralelizacija** ili **uparivanje** (eng. **alignment**). Proces paralelizacije je nezavisan od samog kreiranja korpusa, ali je neophodan kao prethodni korak pre no što se pristupi izgradnji paralelizovanog korpusa.

U opštem slučaju varijante jedinice prevođenja mogu predstavljati različite strukturne nivoe teksta: ceo dokument, poglavlja, odeljke, pasuse, rečenice, reči. Alati za automatsko kreiranje paralelizovanog teksta (biteksta) koje koristi Grupa za jezičke tehnologije Univerziteta u Beogradu zahtevaju da izvorni i ciljni tekst budu strukturno anotirani na nivou odeljaka, pasusa i segmenata. Pritom se kao varijante jedinice prevođenja koriste segmenti, ali se uparivanje odgovarajućih delova teksta vrši na sva tri nivoa radi veće preciznosti. U praksi je segment najčešće rečenica, ali se zbog različite strukture izvornog i ciljnog teksta mogu dogoditi i sledeći slučajevi:

- segment izvornog jezika je jedna rečenica, a odgovarajući segment ciljnog jezika se sastoji od dve ili više rečenica i obrnuto;
- segment izvornog jezika je jedna rečenica, a odgovarajući segment ciljnog jezika



Slika 6.6: Faze u kreiranju dvojezičnog korpusa



Slika 6.7: Model dvojezičnog korpusa

je deo rečenice i obrnuto;

- segment izvornog jezika je jedna rečenica, a odgovarajući segment ciljnog jezika ne postoji i obrnuto;
- i segment izvornog jezika i odgovarajući segment ciljnog jezika se sastoje od dve ili više rečenica, ali odgovarajuće rečenice nisu u istom redosledu u izvornom i ciljnom tekstu.

Svi navedeni slučajevi se najčešće pojavljuju u slučaju paralelnih tekstova književnoumetničkog stila kao posledica „slobodnog prevoda", s obzirom da prevod takvih tekstova ne pretenduje na doslovnost, već na pronalaženje jezičkog izraza ciljnog jezika koji će najpribližnije predstaviti originalnu poruku autora.[31] Neki od navedenih slučajeva se javljaju i prilikom uparivanja odgovarajućih pasusa: u izvornom ili

[31]Kada su u pitanju prevodi na srpski jezik, nema bolje ilustracije od prevoda Stanislava Vinavera koji je često, uz prevod originalne rečenice, dodavao i par svojih rečenica.

ciljnom tekstu se može dogoditi da neki pasus nedostaje ili da su neki pasusi spojeni u jedan.

Faze prethodne obrade tekstova (prikupljanje tekstova za korpus, konverzija tekstova u format čistog teksta, primena kodne sheme aurora (samo u tekstovima na srpskom jeziku), detekcija i ispravljanje grešaka) realizuju se na isti način kao i u slučaju jednojezičnog korpusa (v. odeljke 6.1, 6.2). Faza strukturne anotacije nije više opciona već obavezna, a detalji će biti detaljnije razmotreni u pododeljku Paralelizacija tekstova, str. 294.

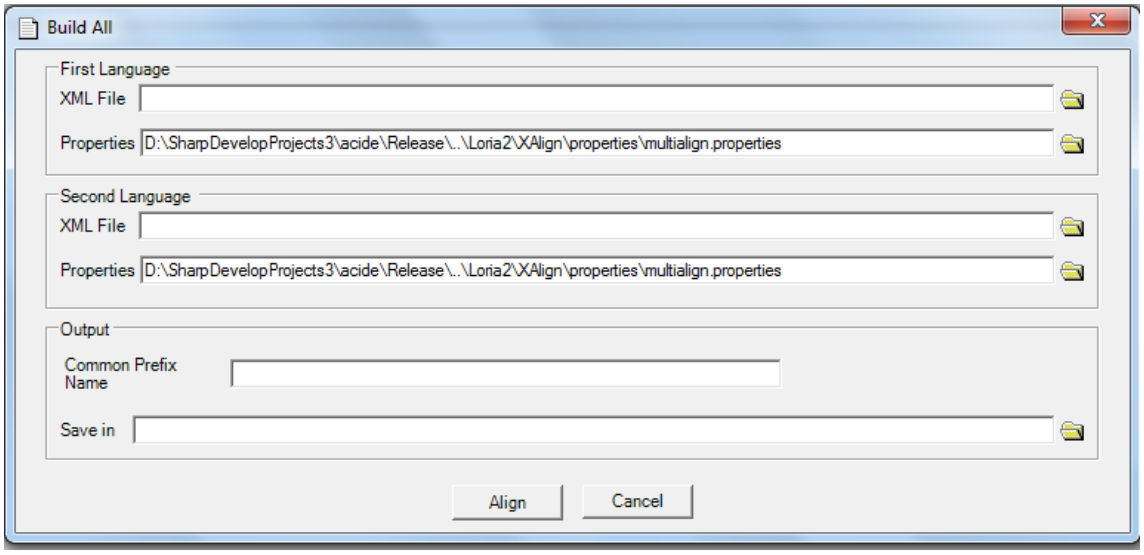
U slučaju tekstova na izvornom jeziku koji nije srpski, tekstovi do sada kreiranih korpusa su, kao i srpski, koristili isti kodni raspored ISO 8859-1 (Latin-1). To je, s jedne strane, bilo neophodno zbog ograničenja softvera za indeksiranje (v. odeljak 6.4), a sa druge strane, svi simboli alfabeta stranih jezika prisutnih u kreiranim paralelizovanim korpusima (engleski, francuski) su mogli da se predstave kodnom shemom Latin-1.

Paralelizacija tekstova

Grupa za jezičke tehnologije Univerziteta u Beogradu razvila je programski paket ACIDE za pripremu paralelnih korpusa ([Obradović et al., 2008]). Naziv ACIDE je skraćenica za *Aligned Corpora Integrated Development Environment* (integrisano razvojno okruženje za paralelne korpuse). Programski paket ACIDE se sastoji iz dva modula, Loria i TMX.

Modul Loria (Slika 6.8) predstavlja grafičko korisničko sučelje sa ciljem da omogući lakše korišćenje postojećih alata za paralelizaciju tekstova — XAlign i Concordancier. XAlign ([Bonhomme et al., 2001]) i Concordancier ([Nguyen & O'Rourke, 2003]) su razvijeni u programskom jeziku JAVA kao alati koji se koriste iz komandne linije, a nastali su u okviru **Lorenske laboratorije za informatička istraživanja i primene** (fr. **Laboratoire Lorrain de Recherche en Informatique et ses Applications**, skr. **LORIA**).

XAlign se koristi za automatsko kreiranje biteksta u formatu zasnovanom na specifikaciji TEI. Radi veće preciznosti pri povezivanju odgovarajućih varijanti jedinica prevođenja, XAlign pokušava da uparuje delove teksta na tri strukturna nivoa



Slika 6.8: ACIDE: dijalog za zadavanje ulaznih parametara paralelizacije

koji su označeni sa DIV, PARAG i PHRASE, a kojima u praksi najčešće odgovaraju nivoi odeljka, pasusa i segmenta (tj. nivo rečenice, a ponekad i dela rečenice). Da bi XAlign mogao da sprovede uparivanje na pomenutim strukturnim nivoima, neophodno je da oni budu anotirani u skladu sa specifikacijom TEI. XAlign koristi posebnu konfiguracionu datoteku[32] u kojoj se za svaki TEI-element ulaznog teksta mora navesti oznaka da li se koristi za anotaciju nekog strukturnog nivoa (DIV, PARAG ili PHRASE) ili oznaka da je TEI-element nebitan za proces paralelizacije (IGNORE ili TRANSP). TEI-element označen sa IGNORE ili TRANSP se ignoriše tokom paralelizacije, pri čemu se te dve oznake razlikuju po tretmanu dece označenog TEI-elementa. Ukoliko se želi ignorisanje i označenog TEI-elementa i njegove dece, koristi se oznaka IGNORE. Ukoliko se ignoriše samo označeni TEI-element, dok se njegova deca dalje analiziraju (tj. „deca su transparentna"), koristi se oznaka TRANSP. Ukoliko se neki TEI-element koristi u tekstu, a nije naveden u konfiguracionoj datoteci, tretira se kao da je naveden sa oznakom TRANSP, tj. on se ignoriše, ali ne i njegova deca.

Sama konfiguraciona datoteka za XAlign koristi format čistog teksta, pri čemu je svaka linija oblika `TEI-element=strukturni_nivo` (Slika 6.9).

[32]Primer konfiguracione datoteke *multialign.properties* se isporučuje sa instalacijom programa XAlign.

```
TEI=TRANSP
teiheader=IGNORE
text=TRANSP
body=TRANSP
div=DIV
p=PARAG
lg=PARAG
seg=PHRASE
s=PHRASE
l=PHRASE
head=IGNORE
pb=IGNORE
```

Slika 6.9: Primer konfiguracione datoteke programa XAlign

Važno ograničenje je da svi TEI-elementi označeni kao PHRASE moraju biti u okviru TEI-elemenata označenih kao PARAG, a svaki TEI-element označen kao PARAG mora biti u okviru nekog TEI-elementa označenog kao DIV. Takođe, PHRASE-elementi ne smeju sadržati druge elemente označene kao PHRASE, PARAG ili DIV, a ni PARAG-elementi ne smeju sadržati druge elemente označene kao PARAG ili DIV. S druge strane, DIV-elementi mogu sadržati proizvoljan broj ugnježdenih TEI-elemenata označenih kao DIV.

U praksi rezultat strukturne anotacije polaznih tekstova na izvornom i ciljnom jeziku predstavljaju XML-dokumenti validni u odnosu na DTD koji opisuje osnovni podskup specifikacije TEI-Lite (*teilite.dtd*), pri čemu se kao PHRASE-elementi najčešće koriste `seg` ili `s`, kao PARAG-element — `p`, a kao DIV-element — `div`. Sam XAlign ne zahteva da ulazni dokumenti budu validni u odnosu na *teilite.dtd*, već da budu dobro formirani XML-dokumenti čiji elementi tipa PHRASE, PARAG i DIV zadovoljavaju navedena ograničenja. Stoga, ako postoji potreba da se najpre formira bitekst, a naknadno popuni zaglavlje ulaznih TEI-dokumenta (element `teiHeader`), dovoljno je da ulazni dokumenti budu validni u odnosu na minimalni DTD (Slika 6.10). Minimalni DTD zahteva da:

- sadržina celog dokumenta bude između etiketa `<body>` i `</body>` (tj. `body` je koreni element XML-dokumenta);
- jedina deca elementa `body` mogu biti jedan ili više elemenata `div`;
- jedina deca elementa `div` mogu biti jedan ili više elemenata `p`;

- jedina deca elementa `p` mogu biti jedan ili više elemenata `seg`;
- sadržaj elementa `seg` mogu biti isključivo parsirani karakterski podaci (tj. `seg` je tekstuelni čvor XML-dokumenta).

```
<!ELEMENT body (div+)>
<!ELEMENT div (p+)>
<!ELEMENT p (seg+)>
<!ELEMENT seg (#PCDATA)>
```

Slika 6.10: Minimalni DTD za ulazne dokumente programa XAlign

Proces automatske paralelizacije tekstova pomoću programa XAlign teče u dve faze (Slika 6.11):

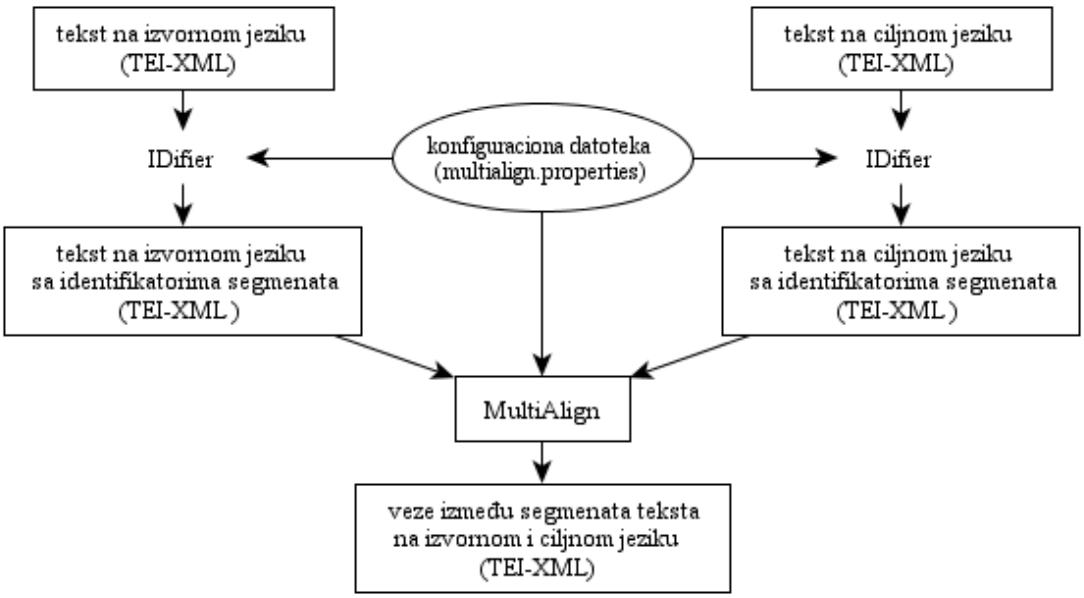
(X1) Najpre se oba ulazna teksta dodatno anotiraju, tj. svakom elementu tipa PHRASE se automatski pridružuje XML-atribut čija je vrednost jedinstveni identifikator u okviru teksta[33]. Za ovu fazu se dva puta poziva klasa[34] IDifier programa XAlign, jednom za ulazni tekst na izvornom jeziku, a jednom za ulazni tekst na ciljnom jeziku.

(X2) Klasa MultiAlign programa XAlign, kojom je implementiran algoritam automatske paralelizacije opisan u [Bonhomme & Romary, 1995; Romary & Bonhomme, 2000], pokušava da upari elemente sva tri strukturna nivoa (DIV, PARAG, i PHRASE), a rezultat je TEI-dokument sa vezama između elemenata tipa PHRASE uspostavljenih preko njihovih jedinstvenih identifikatora u tekstu.

Trenutno postoje dve verzije programa XAlign koje se prevashodno razlikuju u formatima izlaznih dokumenata koje proizvode klase IDifier i MultiAlign. Klase IDifier i MultiAlign i formati koje proizvode su detaljno objašnjeni u Dodatku C.

Concordancier je program za pregled biteksta u starijem XAlign-formatu (Dodatak C.2), ručno ispravljanje pogrešnih uparivanja varijanti jedinica prevođenja izvornog i ciljnog jezika, kao i za pretragu biteksta (po čemu je i dobio ime). Ovaj

[33]Identifikator ne sme da sadrži beline (razmak, tabulator, znak za novi red).
[34]Ovde se pod klasom podrazumeva klasa u programskom jeziku JAVA.

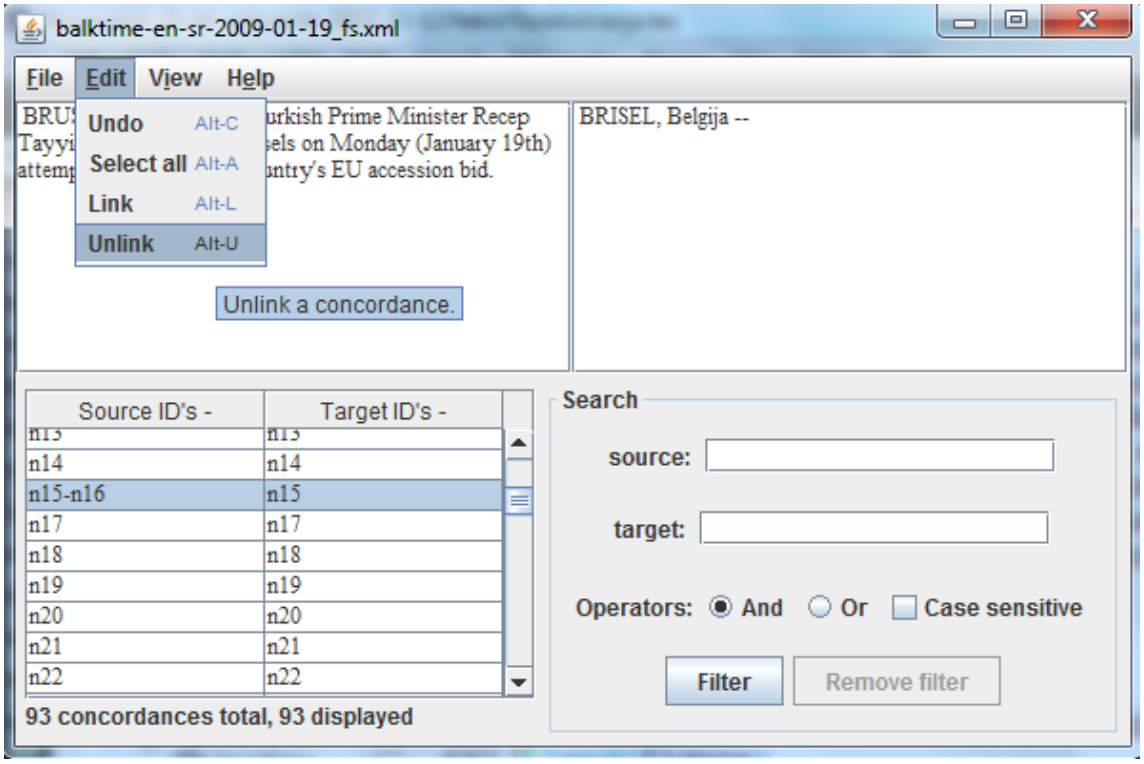


Slika 6.11: Automatska paralelizacija pomoću programa XAlign

program predstavlja bitekst kao tabelu od dve kolone (Slika 6.12). U levoj koloni tabele su predstavljene varijante jedinice prevođenja teksta na izvornom jeziku, a u desnoj — varijante jedinice prevođenja teksta na ciljnom jeziku. Dve varijante jedinice prevođenja su uparene ako i samo ako se nalaze u istom redu tabele, tj. red tabele predstavlja jednu jedinicu prevođenja. Ćelije tabele sadrže listu identifikatora segmenata iz odgovarajuće varijante jedinice prevođenja. Elementi liste identifikatora segmenata su razdvojeni crticom kao separatorom. Tako, ako u jednom redu tabele stoji $n15$–$n16$ u levoj koloni, a $n15$ u desnoj koloni, to znači da segmenti teksta na izvornom jeziku označeni sa $n15$ i $n16$ predstavljaju jednu varijantu jedinice prevođenja kojoj u tekstu na ciljnom jeziku odgovara varijanta jedinice prevođenja sastavljena od samo jednog segmenta označenog sa $n15$[35]. Opcije Link i Unlink menija Edit omogućavaju ručno uspostavljanje i raskidanje veza između skupova segmenata, tj. formiranje i rasformiranje jedinica prevođenja i njihovih varijanti na izvornom i ciljnom jeziku.

Modul TMX omogućava da se na osnovu biteksta u formatu XAlign generišu verzije istog biteksta u drugim formatima kao što su Vanila, TMX i HTML. Modul je

[35]Oznaka $n15$ znači da je u pitanju petnaesti po redu segment u tekstu.



Slika 6.12: Englesko-srpski bitekst u programu Concordancier

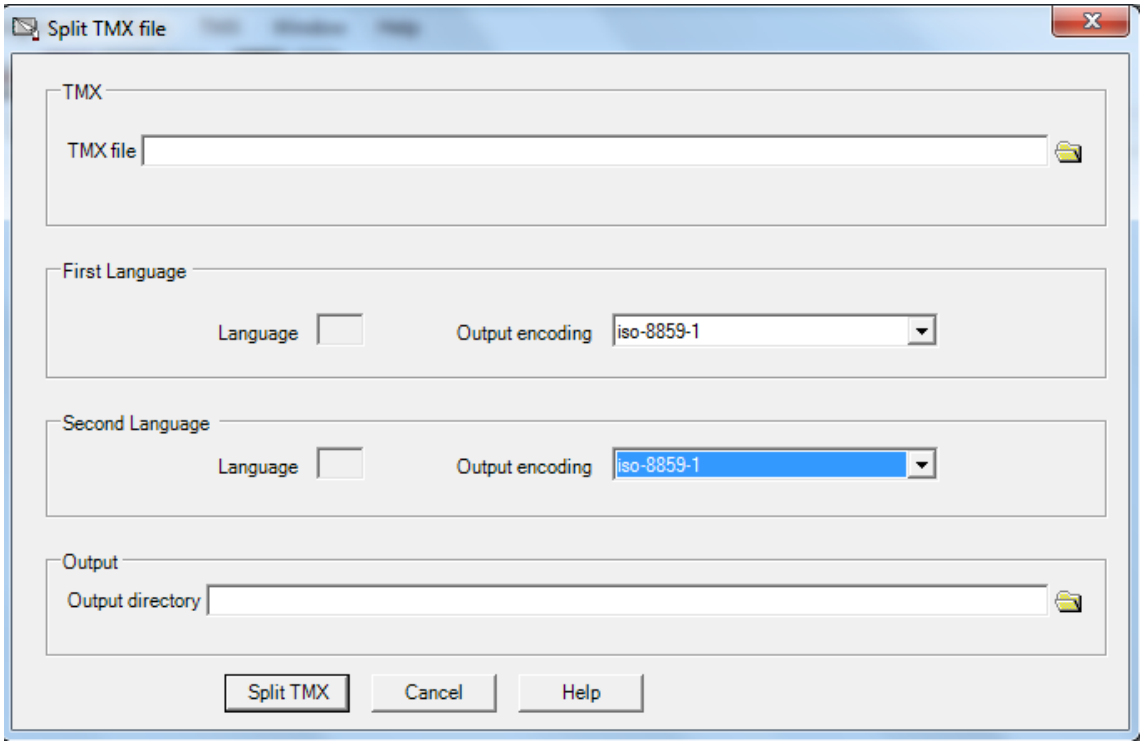
dobio ime po istoimenom formatu koji se koristi u narednim fazama izrade paralelizovanog korpusa. TMX je skraćenica za **Translation Memory eXchange** (srp. **prevodilačka memorija za razmenu**). Kao što ime formata sugeriše, TMX je format paralelizovanog teksta namenjen jednostavnoj razmeni podataka između različitih softverskih prevodilačkih alata, kao i između različitih firmi koje se bave održavanjem prevodilačkih memorija ([TMX, 2005]). **Prevodilačka memorija** (eng. **Translation Memory eXchange**, skr. **TM**) je posebna vrsta baze podataka koju koriste programski alati namenjeni kao ispomoć prevodiocima (eng. **Computer-Assisted Translation Software**, skr. **CAT Software**).

TMX je otvoreni standard zasnovan na XML-u, pri čemu koristi standarde Međunarodne organizacije za standardizaciju (ISO) za predstavljanje datuma i vremena, kao i standardne kodove imena država i jezika. Nastao je 1998. godine u okviru **Društva za standarde u industriji lokalizacije** (eng. **Localization Industry Standards Association**, skr. **LISA**). Iako se ova organizacija ugasila 2011. godine, brigu o lokalizaciji i formatu TMX preuzeo je **Evropski institut za teleko-**

munikacijske standarde (eng. **European Telecommunications Standards Institute**, skr. **ETSI**).

TMX-dokument se sastoji od zaglavlja i tela dokumenta. Zaglavlje dokumenta (element `header`) čine metapodaci koji opisuju paralelizovane tekstove. Telo dokumenta (element `body`) se sastoji iz skupa jedinica prevođenja (elementi `tu`). Jedna jedinica prevođenja obuhvata dve ili više semantički ekvivalentne varijante jedinice prevođenja (element `tuv`). Redosled jezika u svakom elementu `tu` je isti, pri čemu prvi element `tuv` najčešće predstavlja deo teksta na izvornom jeziku, a drugi — na ciljnom jeziku. Svaki element `tu` poseduje atribut `xml:lang` koji označava jezik teksta u okviru tog elementa.

U okviru modula TMX postoji opcija za razlaganje TMX-datoteke (eng. Split TMX file) kojom se bitekst u TMX formatu deli na dve datoteke (Slika 6.13), pri čemu prva datoteka sadrži anotirane varijante jedinica prevođenja na izvornom jeziku, a druga — na ciljnom. Za razlaganje TMX-datoteke koristi se XSL-transformacija opisana u Dodatku D.3. Datoteke dobijene razlaganjem predstavljaju ulazne tekstove za kreiranje dva jednojezična korpusa i validne su u odnosu na odgovarajući DTD (Slika 6.14).



Slika 6.13: ACIDE: razlaganje TMX-datoteke

```
<!ELEMENT text (tu+)>
<!ELEMENT tu (seg+)>
<!ELEMENT seg (#PCDATA)>
```

Slika 6.14: DTD za datoteke dobijene razlaganjem TMX-dokumenta

7
Pretraga

7.1 CQL (CQP-upitni jezik)

Kao što je već pomenuto u odeljku 4.3, CQL se koristi kao akronim u tri različita slučaja:

- da označi *CQP-upitni jezik* (eng. *CQP Query Language*) koji koristi IMS OCWB (v. odeljak 4.3, str. 209),
- da označi *Korpusni upitni jezik* (eng. *Corpus Query Language*) koji koriste SketchEngine i NoSketchEngine (v. odeljak 4.3, str. 222) i
- da označi *Korpusni upitni jezik* (eng. *Corpus Query Language*) koji koristi Xaira[1] (v. odeljak 4.3, str. 233).

Ovde će prevashodno biti reči o prvom slučaju, tj. o CQP-upitnom jeziku ([Evert & The OCWB Development Team, 2010b]). CQP je akronim za *alat za obradu korpusnih upita* (eng. *Corpus Query Processor*), koji se koristi za pretragu korpusa indeksiranih pomoću IMS OCWB-a i koji predstavlja sastavni deo sistema korpusnih alata IMS OCWB.

[1] Nema nikakve veze sa istoimenim jezikom koji koriste SketchEngine i NoSketchEngine.

CQL-upiti su zasnovani na regularnim izrazima koji koriste sintaksu usklađenu sa specifikacijom standarda POSIX[2].

Najjednostavijim CQL-upitom (Tabela 7.1)[3] se zadaje uslov koji moraju da zadovolje pozicioni atributi jedne korpusne pozicije (u daljem tekstu: *p-uslov*). P-uslov zapravo predstavlja logički (bulovski) izraz, tj. izraz čija je vrednost *tačno* ili *netačno*. Prosti p-uslov se navodi u formi `p = v` ili `p != v`, pri čemu su `p` i `v` redom naziv i vrednost pozicionog atributa. Vrednost pozicionog atributa `v` se može zadati u formi POSIX-regularnog izraza nad azbukom karaktera[4]. Prosti p-uslov u formi `p = v` zahteva da pozicioni atribut `p` korpusne pozicije ima vrednost `v`, dok se p-uslovom u formi `p != v` nalaže suprotno, tj. da je vrednost pozicionog atributa `p` korpusne pozicije različita od `v`. Korišćenjem binarnih logičkih (bulovskih) operacija konjunkcije (&) i disjunkcije (|) i unarne bulovske operacije negacije (`!`), može se konstruisati složeni p-uslov koji pozicioni atributi jedne korpusne pozicije moraju zadovoljavati. Sam CQL-upit nad jednom korpusnom pozicijom se zadaje u formi

```
[ p-uslov ] ;
```

to jest, `p-uslov` se navodi između srednjih zagrada, a karakter `;` se koristi kao terminator upita.

Prilikom zadavanja vrednosti `v` pozicionog atributa u formi regularnog izraza, treba uzeti u obzir da CQL-upit pravi razliku između velikih i malih slova. Da bi CQL-upit nad jednom korpusnom pozicijom ignorisao razliku između velikih i malih slova, koristi se marker %c:

```
[ p-uslov ] %c ;
```

pri čemu oznaka potiče od engleskog izraza *case insensitive* (eng. *„neosetljiv" na velika i mala slova*).

Radi jednostavnijeg zadavanja upita, CQP omogućava korisniku da zada podrazumevani pozicioni atribut, a ukoliko to ne učini, prvi pozicioni atribut (`word`) se smatra za podrazumevani. Ukoliko je `p` podrazumevani atribut, upit

[2]Preciznije, sintaksa CQL-regularnih izraza je podskup sintakse POSIX-regularnih izraza.

[3]Imena pozicionih atributa (`pos`, `lemma`) i oznake vrsta reči (`A`, `V`, itd.) zavise od konkretnog korpusa. Ovde su navedena imena i oznake koje koristi SrpKor2013 (v. odeljak 6.3, str. 285, kao i odeljak 6.4, str. 288).

[4]IMS OCWB je donedavno podržavao isključivo osmobitne kodove, pre svega ISO 8859-1, a odnedavno, uz izvesna ograničenja, omogućava i primenu rasporeda UTF-8.

Tabela 7.1: CQP-upiti kao uslovi zadati nad pozicionim atributima jedne korpusne pozicije. Svi primeri su preuzeti kao rezultati pretrage korpusa SrpKor2013.

CQL-upit	**objašnjenje**	**regularni skup**
`[ word = "most" ] ;`	sva pojavljivanja tokena `most`	$\{most\}$
`[ lemma = "most" ] ;`	svi oblici leme `most`	$\{most, mosta, mostu, \dots\}$
`[ lemma = "most" ] %c ;`	svi oblici leme `most`, ignoriše se razlika između velikih i malih slova	$\{most, mosta, mostu,$ $Most, Mosta, Mostu,$ $MOST, MOSTA, MOSTU, \dots\}$
`[ lemma = "most.*" ] %c ;`	svi oblici lema koje počinju sa `most`, ignoriše se razlika između velikih i malih slova	$\{most, mosta, mostu,$ $Mostar, Mostara, Mostaru,$ $mostarina, mostarine, \dots\}$
`[ pos = "A" & lemma = ".*ski" ] ;`	svi oblici prideva čija se lema završava na `ski`	$\{srpski, srpskog, pariski,$ $sremska, tursko, \dots\}$
`[ word = "more" & lemma = "moriti" ] ;`	oblik `more` leme `moriti`	$\{more\}$
`[ word = "more" & pos = "V" ] ;`	`more` kao oblik glagola	$\{more\}$
`"most" ;`	sva pojavljivanja tokena `most` pod pretpostavkom da je `word` podrazumevani pozicioni atribut	$\{most\}$
`"most" ;`	svi oblici leme `most` pod pretpostavkom da je `lemma` podrazumevani pozicioni atribut	$\{most, mosta, mostu, \dots\}$

```
[ p = v ] ;
```

se može zadati i kao

```
v ;
```

(videti poslednja dva reda Tabele 7.1).

Opšti CQP-upit je zapravo regularni izraz nad azbukom čiji su elementi („slova") CQP-upiti nad jednom korpusnom pozicijom (Tabela 7.2). CQP-regularni izrazi koriste metakaraktere POSIX-regularnih izraza: `|`, `*`, `+`, `?`, `.`, `[` i `]` za klase karaktera, zagrade za grupisanje i izbegavanje prioriteta, `\` u uobičajenom značenju. CQP-regularni izrazi ne koriste sidra (`^` i `$`), ali se metakarakter `^` koristi u okviru negativnih klasa karaktera u istom značenju kao u proširenim regularnim izrazima standarda POSIX. Pored tačke kao „džoker-znaka" koji zamenjuje proizvoljni karakter azbuke, CQP koristi i `[]` kao „džoker-token", tj. par srednjih zagrada `[]` zamenjuje proizvoljan token u korpusu. „Džoker-token" omogućava da se par leksema traži na određenom rastojanju (v. poslednji red Tabele 7.2).

7.2 Pretraga SrpKor-a

Veb-sučelje omogućava jednostavnu i naprednu pretragu SrpKor-a. Napredna pretraga koristi većinu mogućnosti upitnog jezika CQL (opisanog u odeljku 7.1). Jednostavna pretraga omogućava samo osnovne opcije pretraživanja, sve zarad pojednostavljene sintakse kojom se zadaje upit.

U trenutku pisanja rada, veb-sučelje je još uvek u izradi. Trenutne mogućnosti jednostavne pretrage korišćenjem veb-sučelja su (Slika 7.3):

(1) jednostavnoj pretrazi je dostupan isključivo prvi pozicioni atribut (`word`), tj. mogu se pretraživati samo korpusni tekstovi, ali ne i anotacija;

(2) prilikom zadavanja upita nad jednom korpusnom pozicijom, dovoljno je koristiti samo vrednost pozicionog atributa `word` bez pratećih navodnika. Na primer, umesto upita `[ word = "Beograd" ]`, odnosno `"Beograd"`, dovoljno je navesti samo `Beograd`;

Tabela 7.2: CQP-upiti kao regularni izrazi nad korpusnim tokenima. Radi preglednosti, svi upiti su prikazani u više linija kako bi se razdvojili CQL-upiti koji se odnose na jednu korpusnu poziciju. Prilikom zadavanja upita CQP-u, znake za novi red treba zameniti razmacima. Svi primeri su preuzeti kao rezultati pretrage korpusa SrpKor2013.

CQL-upit	**objašnjenje**	**regularni skup**
`[ word = "novi" ]` `[ word = "most" ] ;`	sva pojavljivanja sintagme `novi most`	{*novi most*}
`[ lemma = "nov" ]` `[ lemma = "most" ] ;`	sve varijacije oblika lema `nov` i `most`	{*novi most, novog mosta, novom mostu, ...*}
`[ pos = "N" ]` `[ pos = "V" ] ;`	niz tokena koji čine imenica i glagol	{*čovek stremi, ljudi žive, škole svršio, kraja uhvatiti, ...*}
`[ pos = "A" ]{2,}` `[ pos = "N" ] ;`	niz tokena koji čine bar dva uzastopna oblika prideva i oblik imenice	{*prvi sunčani dan, stari zimski kaput, težak celoj zemlji, veliki zeleni livadski skakavac, ...*}
`[ pos = "A" & lemma = ".*ski" ] {2}` `[ pos = "N" ] ;`	niz tokena koji čine tačno dva uzastopna oblika prideva čija se lema završava na `ski` i imenica	{*španskom građanskom ratu, beogradskom Pionirskom parku, japanske carske armije, ...*}

`[ word != "se" ]`
`"odmaram"`
`[word != "se" ];`
	token `odmaram` ispred i iza kog se ne nalaze povratne zamenice (`se`, `sebe`)	{*da odmaram ekipu,*
		Sada odmaram do,
		da odmaram u, ...}

`"je"`
`[]{0,2}`
`"radio" ;`
	tokeni `je` i `radio` između kojih se nalaze najviše dva tokena	{*je radio, je dugo radio,*
		je veoma rano radio, ...}

(3) prilikom formulisanja upita, vrednosti pozicionih atributa se moraju kodirati kodom aurora (v. odeljak 6.2). Na primer, ukoliko se traže pojavljivanja tokena `šećer`, odgovarajući upit se formuliše kao `sxecxer`;

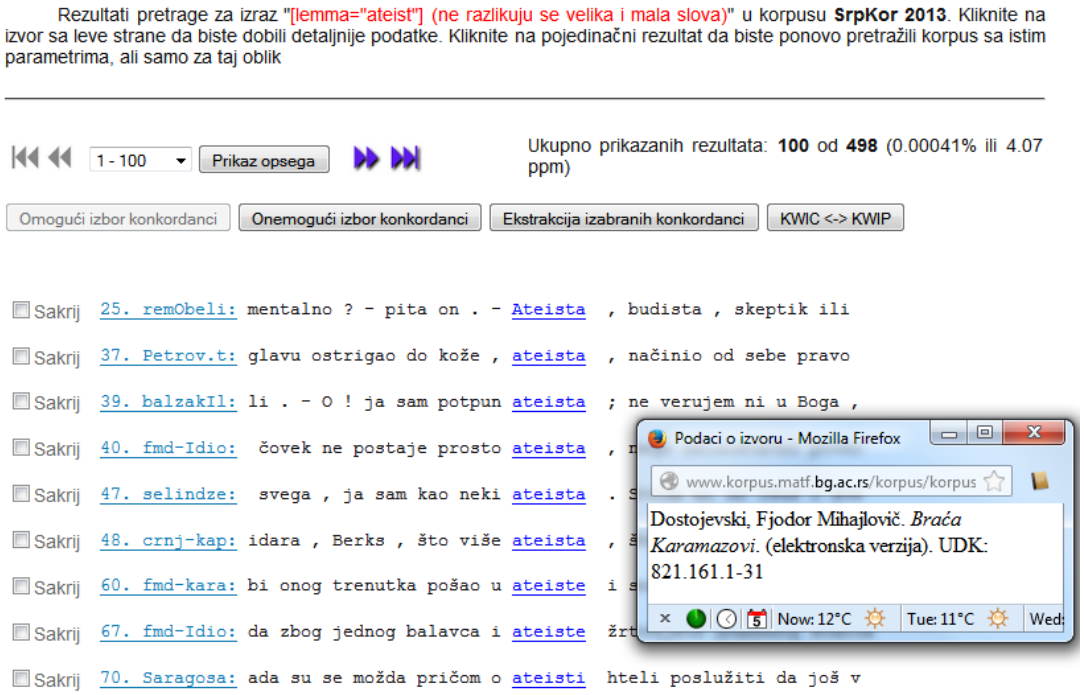
(4) regularni izrazi se mogu koristiti i u okviru vrednosti pozicionog atributa `word` i kao regularni izrazi nad tokenima (v. odeljak 7.1). Na primer, CQL-upit

```
"je" []{0,2} "usta(o|la|lo)"
```

se može koristiti u jednostavnoj pretrazi u obliku

```
je []{0,2} usta(o|la|lo);
```

(5) rezultati pretrage su dostupni u obliku konkordanci, pri čemu format konkordanci može biti KWIC (v. odeljak 3.2, str. 181 i Sliku 7.1) ili KWIP (eng. Key Word In Paragraph), tj. format koji konkordance prikazuje kao pasuse sa istaknutim ključnim rečima (Slika 7.2).



Slika 7.1: Pretraga SrpKor-a: prikaz konkordanci u formatu KWIC sa bibliografskim informacijama o izvoru za upit `[ lemma = "ateist" ]`.

(6) korisnik ima mogućnost da pregleda konkordance stranu po stranu (podrazumevani broj konkordanci po strani je 100), kao i da pristupi određenoj strani u rezultatu.

(7) korisnik može da izabere koje konkordance želi da zadrži u prikazu, kao i da sačuva izdvojene konkordance.

(8) (ne)razlikovanje velikih i malih slova prilikom pretraživanja;

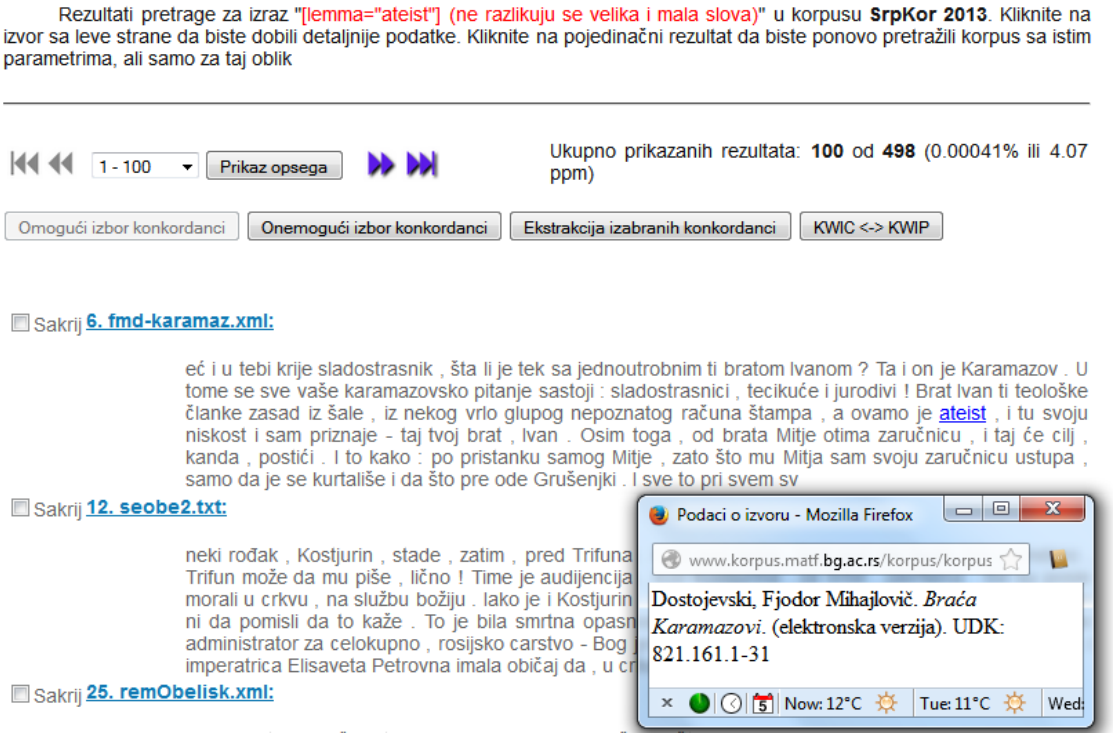
(9) sortiranje konkordanci po levom ili desnom kontekstu (sa interpunkcijom);

(10) podešavanje dužine levog i desnog konteksta ključne reči u prikazanim konkordancama;

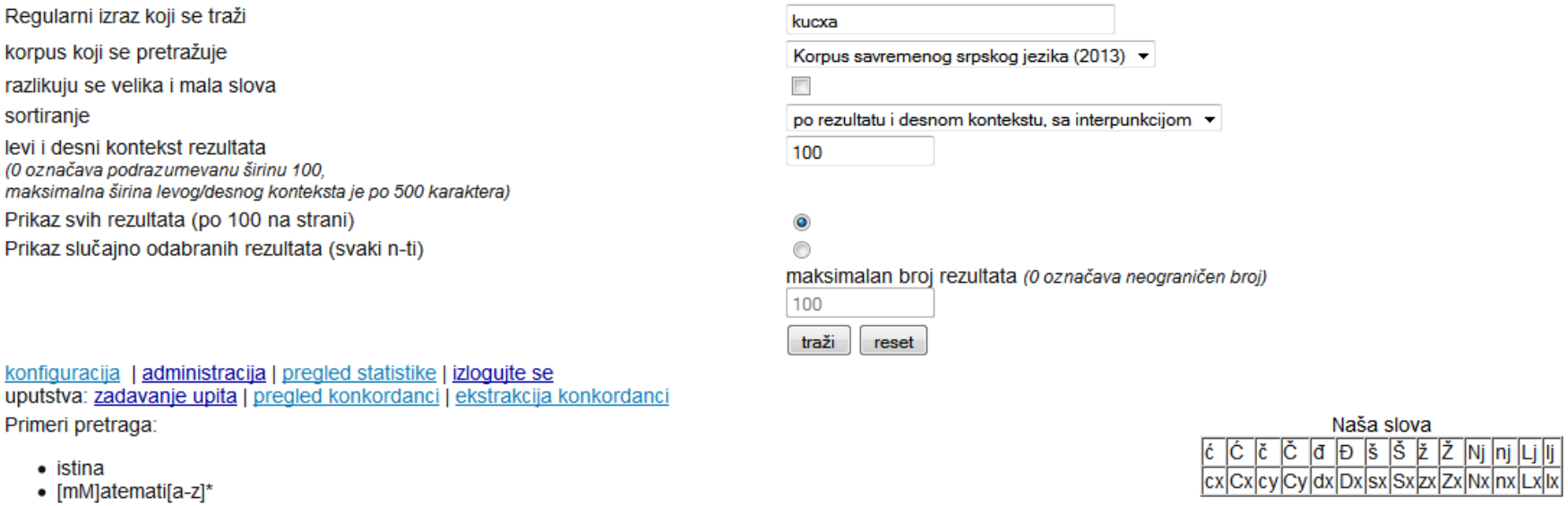
(11) mogućnost pregleda svih konkordanci i mogućnost pregleda (pseudo)slučajno generisanih n konkordanci (korisnik zadaje prirodan broj n);

(12) prikaz ukupnog broja pronađenih konkordanci u formi apsolutnih i relativnih učestanosti.

Napredna pretraga ima sva navedena svojstva jednostavne pretrage osim svojstava (1) i (2), uz mogućnost pretraživanja dodatnih pozicionih atributa. U trenutku pisanja ovog rada, dostupna je pretraga po morfološkoj anotaciji korpusa, tj. po pozicionim atributima `pos` (vrsta reči) i `lemma` (lema).



Slika 7.2: Pretraga SrpKor-a: prikaz konkordanci u formatu KWIP sa bibliografskim informacijama o izvoru za upit `[ lemma = "ateist" ]`.



Slika 7.3: Pretraga SrpKor-a: opcije veb-sučelja.

8

Zaključak i dalji rad

8.1 Zaključak

U ovom radu je razmatran problem izgradnje korpusa savremenog srpskog jezika kao referentnog jezičkog resursa.

U prvom delu rada je dat pregled osnovnih pojmova korpusne lingvistike kao metodologije istraživanja jezika. Izloženi su dosadašnji pokušaji da se precizira pojam korpusa i utvrde jasni kriterijumi njegove izgradnje kao reprezentativnog uzorka jezika, uključujući i aktuelna kritička preispitivanja definicije korpusa uslovljena uticajem veba. Analizirani su parametri korpusa (nosač, domen i namena, obim (veličina), period, izvor/medijum, anotacija i višejezičnost) i na osnovu njih su klasifikovani korpusi. Nacionalni korpusi slovenskih jezika su posebno analizirani, kako bi se iskoristilo njihovo iskustvo i uporedili rezultati na izgradnji odgovarajućeg elektronskog, dinamičkog, sinhronog, balansiranog i anotiranog korpusa srpskog jezika.

U drugom delu rada su detaljno razmatrane konkretne faze izgradnje jednog korpusa, od prikupljanja, digitalizacije i klasifikacije tekstova za korpus, preko konverzije korpusnih tekstova u odgovarajući format elektronskog teksta, lingvističke obrade, anotacije elektronskih tekstova za korpus, zaključno sa indeksiranjem i kompresijom korpusnih tekstova. Takođe su iscrpno analizirani mehanizmi pretrage korpusa, metodi vizuelizacije rezultata pretrage, mogućnosti statističke analize kor-

pusa. Različiti sistemi integrisanih korpusnih alata su upoređivani po sledećim parametrima: licenca, platforma, klijent-server arhitektura i veb, ažurnost i podrška, proširivost, jezički resursi, tipovi pretrage i raspoložive statističke funkcije.

U trećem delu rada je izložen proces konstrukcije Korpusa savremenog srpskog jezika (SrpKor), zaključno sa aktuelnom verzijom SrpKor2013. Izložen je istorijat izgradnje SrpKor-a i predstavljena su njegova svojstva izražena preko sledećih parametara: nosač, domen i namena, obim (veličina), period, izvor/medijum, anotacija, mogućnosti pretrage. Svaka faza konstrukcije korpusa (prikupljanje tekstova, konverzija tekstova u format za čuvanje i indeksiranje, anotacija i indeksiranje) je detaljno obrađena, uključujući i izgradnju paralelnih korpusa kod kojih je izvorni ili ciljni jezik srpski. Takođe su opisane mogućnosti upitnog jezika i veb-sučelja koje se koristi za jednostavnu i naprednu pretragu SrpKor-a.

Rezultati izloženi u ovom radu sugerišu da je ostvaren dobar deo zacrtanih ciljeva. Izgrađen je SrpKor, elektronski, sinhroni i dinamički korpus savremenog srpskog jezika, veličine 122 miliona reči. Tokom izgradnje se pristupilo raznim oblicima anotacije korpusa, tako da su korpusnim tekstovima pridružene bibliografske informacije, jedan deo korpusnih tekstova je strukturno anotiran, tokeni korpusnih tekstova su anotirani morfološkim informacijama o vrsti reči i lemi. Treba napomenuti da balansiranost korpusa još uvek nije postignuta, ali s obzirom da je korpus dinamički i da se neprekidno ažurira, to je ostvarljiv cilj u budućnosti. Takođe, veb-sučelje za pretraživanje korpusa još uvek treba da se dorađuje jer ne dopušta pretragu po celokupnoj anotaciji korpusa. Veb-sučelje trenutno omogućava pretragu korpusnih tekstova po tokenima i postojećoj morfološkoj anotaciji, a prilikom prikazivanja rezultata pretrage, dostupne su bibliografske informacije o izvorima.

8.2 Dalji rad

Izgradnja korpusa je složen proces, pri čemu se uvek može postaviti pitanje da li korpus zaista predstavlja reprezentativan uzorak celokupnog jezika ili jezičkog podskupa koji je predmet istraživanja zasnovanog na korpusu. S obzirom da se reprezentativnost korpusa dovodi u usku vezu sa balansiranošću korpusa, prvi zada-

tak je nastaviti dalje ažuriranje SrpKor-a, dodavanjem novih i zamenom postojećih tekstova. Digitalna biblioteka Filološkog fakulteta je prvi kandidat za uključivanje novih tekstova u korpus.

U pogledu anotacije, prvi zadatak je završetak veb-sučelja koje bi omogućilo pretragu po postojećoj bibliografskoj anotaciji korpusa. S obzirom na raspoložive leksičke resurse za srpski, pre svega elektronski morfološki rečnik srpskog jezika, drugi zadatak je osmisliti način za kompletnu morfosintaksičku anotaciju SrpKor-a, tj. pored informacija o lemi i vrsti reči, dodati i informacije o ostalim morfološkim kategorijama (rod, broj, padež, lice, glagolski oblik, itd.). Takođe, bez obzira što je semantička mreža WordNet za srpski jezik još uvek u izgradnji, treba razmotriti mogućnosti delimične semantičke anotacije SrpKor-a.

U pogledu strukturne anotacije, potrebno je nastaviti konverziju korpusnih tekstova u format TEI/XML, tako da nova verzija SrpKor-a sadrži i elemente logičke strukture teksta, tj. označene rečenice, a u drugom koraku i pasuse.

Uz postojeće paralelne korpuse, englesko-srpski i francusko-srpski, treba započeti sa izgradnjom drugih paralelnih korpusa čiji je izvorni ili ciljni jezik srpski, na primer španski-srpski, rusko-srpski, itd.

Iskustvo na formiranju SrpKor-a kao jezičkog resursa i pratećih paralelnih korpusa je još jednom potvrdilo da je izgradnja korpusa iterativni proces i da je potrebno neprekidno ažurirati rezultate, kako bi dobijeni resursi zaista predstavljali kvalitetnu osnovu za istraživanje srpskog jezika.

Bibliografija

Aarts, J. & Meijs, W. (eds) [1984]. *Corpus Linguistics I: Recent Developments in the Use of Computer Corpora*, Rodopi, Amsterdam.

Aarts, J. & van den Heuvel, T. [1982]. Grammars and Intuitions in Corpus Linguistics, *in* S. Johansson (ed.), *Computer Corpora in English Language Research*, Norwegian Computing Centre for Humanities, Bergen, pp. 66–84.

Abeillé, A. (ed.) [2003]. *Treebanks: Building and Using Parsed Corpora*, Kluwer, Dordrecht.

Abney, S. [1991]. Parsing By Chunks, *Principle-Based Parsing*, Kluwer Academic Publishers, pp. 257–278.

Abney, S. [1996]. Partial Parsing via Finite-State Cascades, *Natural Language Engineering* .

Ahn, D., Tjong Kim Sang, E. & Wilcock, G. (eds) [2006]. *NLPXML '06: Proceedings of the 5th Workshop on NLP and XML: Multi-Dimensional Markup in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, USA.

Aijmer, K. [2008]. Parallel and Comparable Corpora, *in* Lüdeling & Kytö [2008], chapter 16, pp. 275–292.

Allwood, J. [2008]. Multimodal Corpora, *in* Lüdeling & Kytö [2008], chapter 12, pp. 207–225.

Anthony, L. [2005]. AntConc: A Learner and Classroom Friendly, Multi-Platform Corpus Analysis Toolkit, *Proceedings of IWLeL 2004: An Interactive Workshop on Language e-Learning*, Waseda University, Tokyo, pp. 7–13.

Antoine, J.-Y., Mokrane, A. & Friburger, N. [2008]. Automatic Rich Annotation of Large Corpus of Conversational Transcribed Speech: the Chunking Task of the EPAC Project, *in* N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis & D. Tapias (eds), *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), Marrakech, Morocco.
URL: *http://www.lrec-conf.org/proceedings/lrec2008/*

Apresjan, J., Boguslavsky, I., Iomdin, B., Iomdin, L., Sannikov, A. & Sizov, V. [2006]. A Syntactically and Semantically Tagged Corpus of Russian: State of the Art and Prospects, *in* Calzolari et al. [2006], pp. 1378–1381.

Arhar, Š. [2007]. *Kaj početi z referenčnim korpusom Fidaplus*, Filozofska fakulteta, UL, Ljubljana.
URL: *http://www.fidaplus.net/Files/Kaj_poceti_s_korpusom_FidaPLUS_knjizica.pdf*

Arhar, Š., Gorjanc, V. & Krek, S. [2007]. Fidaplus corpus of slovenian: the new generation of the slovenian reference corpus: its design and tools, *Proceedings of the Corpus Linguistics Conference*, pp. 95–110.

Aston, G. & Burnard, L. [1998]. *The BNC Handbook: Exploring the British National Corpus with SARA*, Edinburgh University Press.

Baker, P., Hardie, A. & McEnery, T. [2006]. *A Glossary of Corpus Linguistics*, Edinburgh University Press Ltd.

Bański, P. [2010]. Why TEI Stand-Off Annotation Doesn't Quite Work, *Manuscript, University of Warsaw* .

Bański, P. & Przepiórkowski, A. [2009]. Stand-off TEI Annotation: The Case of the National Corpus of Polish, *Proceedings of the Third Linguistic Annotation*

Workshop, ACL-IJCNLP '09, Association for Computational Linguistics, pp. 64–67.

Bański, P. & Przepiórkowski, A. [2010]. TEI P5 as a Text Encoding Standard for Multilevel Corpus Annotation, *Digital Humanities 2010 Conference Abstracts*, pp. 98—-100.
URL: *http://dh2010.cch.kcl.ac.uk/academic-programme/abstracts/papers/pdf/ab-616.pdf*

Barlow, M. [2003]. *ParaConc: A Concordancer for Parallel Texts (Draft 3/03)*, Athelstan, Houston, TX.
URL: *http://www.athel.com/paraconc.pdf*

Barlow, M. [2004]. *Collocate 1.0: Locating Collocations and Terminology*, Athelstan, Houston, TX.

Barlow, M. [2008]. *ParaConc and Parallel Corpora in Contrastive and Translation Studies*, Athelstan, Houston, TX.

Barlow, M. [2012]. *Concordancing and Corpus Analysis Using MP2. 2*, Athelstan, Houston, TX.
URL: *http://www.cch.kcl.ac.uk/legacy/teaching/avmlan/lectures/AVMLAN%2004.4%20Barlow,%20Manual%20for%20Monoconc.pdf*

Barnbrook, G. [1996]. *Language and Computers. A Practical Introduction to the Computer Analysis of Language*, Edinburgh Textbooks in Empirical Linguistics, Edinburgh University Press.

Baroni, M. & Bernardini, S. [2004]. BootCaT: Bootstrapping Corpora and Terms from the Web, *Proceedings of LREC 2004*, ELDA, Lisbon, pp. 1313–1316.

Baroni, M., Bernardini, S., Ferraresi, A. & Zanchetta, E. [2009]. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora, *Language Resources and Evaluation* **43**(3): 209–226.

Baroni, M., Kilgarriff, A., Pomikálek, J. & Rychlỳ, P. [2006a]. WebBootCaT: A Web Tool for Instant Corpora, *Proceedings of the EuraLex Conference*, Vol. 1, pp. 123–132.

Baroni, M., Kilgarriff, A., Pomikálek, J. & Rychlỳ, P. [2006b]. WebBootCaT: Instant Domain-Specific Corpora to Support Human Translators, *Proceedings of EAMT*, pp. 247–252.

Bergenholtz, H. & Schaeder, B. (eds) [1979]. *Empirische Textwissenschaft: Ausbau und Auswertung von Text-Corpora*, Scriptor, Königstein.

Bergh, G. & Zanchetta, E. [2008]. Web Linguistics, *in* Lüdeling & Kytö [2008], chapter 18, pp. 309–327.

Bernardini, S., Baroni, M. & Evert, S. [2006]. A Wacky Introduction, *in* M. Baroni & S. Bernardini (eds), *Wacky! Working Papers on the Web as Corpus*, GEDIT, Bologna, pp. 9–40.

Bernot, E. & Alarcón, E. [2005]. Index Thomisticus by Roberto Busa SJ and Associates.
URL: *http://www.corpusthomisticum.org/it/index.age*

Biber, D. [1988]. *Variation across Speech and Writing*, Cambridge University Press.

Biber, D. [1993]. Representativeness in Corpus Design, *Literary and Linguistic Computing* **8**(4): 243–257.

Biber, D. & Conrad, S. [2009]. *Register, Genre, and Style*, Cambridge University Press, New York.

Biber, D., Conrad, S. & Reppen, R. [1998]. *Corpus Linguistics: Investigating Language Structure and Use*, Cambridge University Press.

Biber, D. & Finegan, E. [1991]. On the Exploitation of Computerized Corpora in Variation Studies, *in* K. Aijmer & B. Altenberg (eds), *English Corpus Linguistics. Studies in Honour of Jan Svartvik*, Longman, London and New York, pp. 44–63.

Biron, P. V. & Malhotra, A. [2004]. *XML Schema Part 2: Datatypes Second Edition*. W3C Recommendation 28 October 2004.
URL: *http://www.w3.org/TR/xmlschema-2/*

Boag, S., Chamberlin, D., Fernández, M. F., Florescu, D., Robie, J. & Siméon, J. [2010]. *XQuery 1.0: An XML Query Language (Second Edition)*. W3C Recommendation 14 December 2010.
URL: *http://www.w3.org/TR/xquery/*

Böhmová, A., Hajič, J., Hajičová, E. & Barbora, H. [2003]. The Prague Dependency Treebank: A Three-Level AnnotationvScenario, *in* Abeillé [2003], chapter 7, pp. 103–127.

Bongers, H. [1947]. *The History and Principles of Vocabulary Control*, Wocopi, Woerden.

Bonhomme, P., Nguyen, T. M. H. & O'Rourke, S. [2001]. XAlign (Alignement multilingue), *Technical report*, LORIA.
URL: *led.loria.fr/en_outils.php.html#4*

Bonhomme, P. & Romary, L. [1995]. The Lingua Parallel Concordancing Project. Managing Multilingual Texts for Educational Purpose, *Proceedings of Language Engineering*, Montpellier.

Brants, T. [2000]. TnT: A Statistical Part-of-Speech Tagger, *Proceedings of the Sixth Conference on Applied Natural Language Processing*, Association for Computational Linguistics, pp. 224–231.

Bray, T., Paoli, J., Maler, E., Yergeau, F. & Sperberg-McQueen, C. M. [2008]. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation 26 November 2008.
URL: *http://www.w3.org/TR/REC-xml/*

Brill, E. [1995]. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging, *Computational Linguistics* **21**(4): 543–565.

Brill, E. & Mooney, R. J. [1998]. An Overview of Empirical Natural Language Processing, *The AI Magazine* **18**(4): 13–24.

Brown, K. (ed.) [2005]. *The Encyclopedia of Language and Linguistics*, Vol. 1–14, second edn, Elsevier, Oxford.

Bugarski, R. [1995]. *Uvod u opštu lingvistiku*, Zavod za udžbenike i nastavna sredstva, Beograd.

Bulajić, V. [2009]. *UDK u teoriji i praksi: priručnik za bibliotekare*, HESPERIAedu, Beograd.

Bungarten, T. [1979]. Das Korpus als empirische Grundlage in der Linguistik und Literaturwissenschaft, *in* Bergenholtz & Schaeder [1979], pp. 28–51.

Burnage, G. & Dunlop, D. [1993]. Encoding the British National Corpus, *in* J. Aarts, P. d. Haan & N. Oostdijk (eds), *English Language Corpora: Design, Analysis and Exploitation*, Rodopi, Amsterdam, pp. 79–95.

Burnard, L. [1992]. *Corpus Document Interchange Format: Version 1.2*, Oxford Computing Centre, Oxford. BNC Working Paper TGC.W30.

Burnard, L. [2006]. XAIRA : Software for Language Analysis, *in* L. Burnard, M. Dobreva, N. Fuhr & A. Lüdeling (eds), *Digital Historical Corpora*, Vol. 06491 of *Dagstuhl Seminar Proceedings*, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

Burnard, L. & Bauman, S. (eds) [2009]. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, TEI Consortium.

Burnard, L. (ed.) [2007]. *Reference Guide for the British National Corpus (XML Edition)*, Research Technologies Service at Oxford University Computing Services. **URL:** *http://www.natcorp.ox.ac.uk/docs/URG/*

Burnard, L. & Sperberg-McQueen, C. M. (eds) [2012]. *TEI Lite: Encoding for Interchange: An Introduction to the TEI — Final Revised Edition for TEI P5*,

TEI Consortium.
URL: *http://www.tei-c.org/Guidelines/Customization/Lite/*

Busemann, S. [2012]. Shallow Analysis: Light Parsing & Named Entity Extraction.
URL: *http://www.coli.uni-saarland.de/courses/LT1/2012/slides/SNLP-NEE.pdf*

Calzolari, N., Choukri, K., Gangemi, A., Maegaard, Bente Mariani, J., Odijk, J. & Tapias, D. (eds) [2006]. *Proceedings of the Fifth Language Resources and Evaluation Conference (LREC 2006)*, ELRA-ELDA, Genoa, Italy.

Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M. & Tapias, D. (eds) [2010]. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, European Language Resources Association (ELRA), Valletta, Malta.

Carroll, J. [2003]. Text Segmentation, in *The Oxford Handbook of Computational Linguistics* Mitkov [2003], chapter 12, pp. 233–248.

Chinchor, N. [1995]. MUC-6 Named Entity Task Definition (Version 2.1).
URL: *http://cs.nyu.edu/cs/faculty/grishman/NEtask20.book_1.html*

Chinchor, N., Brown, E., Ferro, L. & Robinson, P. [1999]. Named Entity Recognition Task Definition, *MITRE and SAIC* .
URL: *ftp://jaguar.ncsl.nist.gov/ace/phase1/ne99_taskdef_v1_4.pdf*

Chinchor, N. & Marsh, E. [1998]. MUC-7 Information Extraction Task Definition, *Proceedings of MUC-7, Fairfax, Virginia* .
URL: *http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ie_task.html*

Chomsky, N. [1957]. *Syntactic Structures*, Mouton, Paris, France.

Chomsky, N. [1962]. Paper given at the University of Texas 1958. 3rd Texas Conference on Problems of Linguistic Analysis in English.

Chomsky, N. [1987]. *Generative Grammar: Its Basis, Development and Prospects*, Kyoto University of Foreign Studies, Kyoto.

Christ, O. [1994a]. A Modular and Flexible Architecture for an Integrated Corpus Query System, *Proceedings of COMPLEX'94*, Budapest, pp. 23–32.

Christ, O. [1994b]. The IMS Corpus Workbench Corpus Administrator's Manual, *Technical report*, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

Clark, J. [1999]. *XSL Transformations (XSLT) Version 1.0*. W3C Recommendation 16 November 1999.
URL: *http://www.w3.org/TR/xslt*

Clark, J. [2001]. *RELAX NG Specification*, Organization for the Advancement of Structured Information Standards (OASIS). Committee Specification 3 December 2001.
URL: *http://www.oasis-open.org/committees/relax-ng/spec-20011203.html*

Clark, J. & DeRose, S. [1999]. *XML Path Language (XPath) Version 1.0*. W3C Recommendation 16 November 1999.
URL: *http://www.w3.org/TR/xslt*

Collins, M., Ramshaw, L., Hajič, J. & Tillmann, C. [1999]. A Statistical Parser for Czech, *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 505–512.

Cornish, G. P. [1999]. *Copyright: Interpreting the Law for Libraries, Archives and Information Services*, 3 edn, Library Association Publishing.

Courtois, B. & Silberztein, M. (eds) [1990]. *Dictionnaires électroniques du français*, Langue Française.

Cunningham, H., Maynard, D., Bontcheva, K. & Tablan, V. [2002]. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and

Applications, *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

Čermák, F. & Schmiedtová, V. [2003]. The Czech National Corpus Its Structure and Use, *in* B. Lewandowska-Tomaszczyk (ed.), *PALC 2001 – Practical Applications in Language and Computers*, Lódz Studies in Language, Peter Lang, Frankfurt, pp. 207–224.

Дарчук, .-. [2012a]. Корпус Украинского Языка, *Prace Filologiczne* **LXIII**: 099–108.

Дарчук, .-. [2012b]. Морфологічне анотування Корпусу української мови, *Комп'ютерна лінгвістика: сучасне та майбутнє*, Київський національний лінгвістичний університет, Київ, pp. 16–19.

Davies, M. [2002]. Legal Aspects of Corpora Compiling, In Corpora List Archive (Thu Oct 24 2002 - 05:34:01 MET DST).
URL: *http://helmer.hit.uib.no/corpora/2002-4/0017.html*

Delić, V., Sečujski, M., Jakovljević, N., Pekar, D., Mišković, D., Popović, B. M., Ostrogonac, S., Bojanić, M. & Knežević, D. [2013]. Speech and Language Resources within Speech Recognition and Synthesis Systems for Serbian and Kindred South Slavic Languages, *in* Zelezný et al. [2013], pp. 319–326.

Демська-Кульчицька, . [2005]. *Основи національного корпусу української мови*, Інститут української мови Національної академії наук України.

DeRose, S. [2004]. Markup Overlap: A Review and a Horse, *Extreme Markup Languages*, Vol. 2004.

DeRose, S., Daniel, R. J., Grosso, P., Maler, E., Marsh, J. & Walsh, N. [2002]. *XML Pointer Language (XPointer)*. W3C Working Draft 16 August 2002.
URL: *http://www.w3.org/TR/xptr/*

Добровольский, . ., Кретов, . . & Шаров, . . [2005]. Корпус параллельных текстов: архитектура и возможности использования, *in* NKRJ [2005], pp. 263–296. Сборник статей.

Erjavec, T. [2010]. MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora, *Proceedings of the LREC 2010, Malta, 19-21 May 2010.*

Erjavec, T., Evans, R., Ide, N. & Kilgarriff, A. [2003]. From Machine Readable Dictionaries To Lexical Databases: the CONCEDE experience, *COMPLEX 2003, 7th Conference on Computational Lexicography and Text Research*, Budapest, Hungary.

Erjavec, T., Gorjanc, V. & Stabej, M. [1998]. Korpus fida (the fida corpus), *Proceedings of the Conference 'Language Technologies for the Slovene Language'*, Institute "Jožef Stefan", Ljubljana, Slovenia, pp. 124–127.

Erjavec, T., Krstev, C., Petkevič, V., Simov, K., Tadić, M. & Vitas, D. [2003]. The MULTEXT-east Morphosyntactic Specifications for Slavic Languages, *MorphSlav '03: Proceedings of the 2003 EACL Workshop on Morphological Processing of Slavic Languages*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 25–32.

Erjavec, T., Lawson, A. & Romary, L. (eds) [1998]. *East Meets West – A Compendium of Multilingual Resources*, TELRI Association e.V., Institut für deutsche Sprache, Mannheim.

Estoup, J. B. [1916]. *Gammes Sténographiques*, 4 edn, L'Institut sténographique de France.

Evert, S. & Hardie, A. [2011]. Twenty-First Century Corpus Workbench: Updating a Query Architecture for the New Millennium, *Proceedings of the Corpus Linguistics 2011 Conference*, University of Birmingham.

Evert, S. & The OCWB Development Team [2010a]. *Corpus Encoding Tutorial*. The IMS Open Corpus Workbench (CWB 3.0), 5 January 2010.
URL: *http://cwb.sourceforge.net/files/CWB_Encoding_Tutorial.pdf*

Evert, S. & The OCWB Development Team [2010b]. *CQP Query Language Tutorial*. The IMS Open Corpus Workbench (CWB 3.0), 17 February 2010.
URL: *http://cwb.sourceforge.net/files/CQP_Tutorial.pdf*

Fairon, C. [2000]. GlossaNet: Parsing a Web Site as a Corpus, *Lingvisticae Investigationes* **22**(2): 327–340.

Fallside, D. C. & Walmsley, P. [2004]. *XML Schema Part 0: Primer Second Edition*. W3C Recommendation 28 October 2004.
URL: *http://www.w3c.org/TR/xmlschema-0*

Farhan, H. & D'Agostino, D. [2012]. Web Index 2012, *Technical report*, World Wide Web Foundation.
URL: *http://thewebindex.org/2012/10/2012-Web-Index-Key-Findings.pdf*

Ferraresi, A., Zanchetta, E., Baroni, M. & Bernardini, S. [2008]. Introducing and Evaluating ukWaC, a Very Large Web-Derived Corpus of English, *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pp. 47–54.

Fitschen, A. & Gupta, P. [2008]. Lemmatising and Morphological Tagging, *in* Lüdeling & Kytö [2008], chapter 25, pp. 552–564.

Fletcher, W. H. [2004a]. Facilitating the Compilation and Dissemination of Ad-Hoc Web Corpora, *Corpora and Language Learners* **271**.

Fletcher, W. H. [2004b]. Making the Web More Useful as a Source for Linguistic Corpora, *Language and Computers* **52**(1): 191–205.

Fletcher, W. H. [2006]. Concordancing the Web: Promise and Problems, Tools and Techniques, *Language and Computers* **59**(1): 25–45.

Fletcher, W. H. [2012]. Corpus Analysis of the World Wide Web, *in* C. A. Chapelle (ed.), *The Encyclopedia of Applied Linguistics*, Wiley Online Library, pp. 1339––1347.

Fletcher, W. H. et al. [2001]. Concordancing the Web with KWiCFinder, *Third North American Symposium on Corpus Linguistics and Language Teaching*, Citeseer, pp. 23–25.

Francis, N. W. [1982]. Problems of Assembling and Computerizing Large Corpora, *in* S. Johansson (ed.), *Computer Corpora in English Language Research*, Norwegian Computing Centre for the Humanities, Bergen, pp. 7–24.

Francis, N. W. [1992]. Language Corpora B. C., *in* J. Svartvik (ed.), *Directions in Corpus Linguistics. Proceedings of the Nobel Symposium 82*, Vol. 65 of *Trends in Linguistics. Studies and Monographs*, Mouton de Gruyter, Berlin/New York, pp. 17–32.

Francis, W. N. [1979]. Problems of Assembling and Computerizing Large Corpora, *in* Bergenholtz & Schaeder [1979], pp. 110–123.

Francis, W. N. & Kučera, H. [1964]. Brown Corpus Manual, *Technical report*, Department of Linguistics, Brown University, Providence, Rhode Island, US. Revised 1971; Revised and Amplified 1979.
URL: *http://icame.uib.no/brown/bcm.html*

Friburger, N. [2002]. *Reconnaissance automatique des noms propres — Application à la classification automatique de textes journalistiques*, PhD thesis, Université François-Rabelais, Tours.

Friburger, N. & Maurel, D. [2004]. Finite-State Transducer Cascades to Extract Named Entities in Texts, *Theor. Comput. Sci.* **313**(1): 93–104.
URL: *http://dx.doi.org/10.1016/j.tcs.2003.10.007*

Friedl, J. E. F. [2002]. *Mastering Regular Expressions*, 2 edn, O'Reilly.

Gatto, M. [2009]. *From Body to Web*, Studi Anglo-Germanici e dell'Europa Orientale, Editori Laterza - University Press Online (Bari-Roma).
URL: *http://www.lingue.uniba.it/dag/pagine/personale/gatto/Materiale%20didattico%202008_09/GattoIII_light.pdf*

Gavrilidou, M., Labropoulou, P., Piperidis, S., Giouli, V., Calzolari, N., Monachini, M., Soria, C. & Choukri, K. [2006]. Language Resources Production Models: The Case of INTERA Multilingual Corpus and Terminology, *in* Calzolari et al. [2006].

Gianitsová, L. [2005]. Morphological Analysis of the Slovak National Corpus, *in* M. Šimková (ed.), *Insight into the Slovak and Czech Corpus Linguistics*, Veda, Bratislava, pp. 166–178.

Głowińska, K. & Przepiórkowski, A. [2010]. The Design of Syntactic Annotation Levels in the National Corpus of Polish, *in* Calzolari et al. [2010].

Гришина, . . [2005]. Два новых проекта для Национального корпуса: мультимедийный подкорпус и подкорпус названий, *in* NKRJ [2005], pp. 233–250. Сборник статей.

Гришина, . . [2005]. Устная речь в Национальном корпусе русского языка, *in* NKRJ [2005], pp. 94–110. Сборник статей.

Гришина, . . [2009]. История русского ударения, *in* Плунгян et al. [2009], pp. 150–174. Сборник статей.

Гришина, . . [2009]. Мультимедийный русский корпус (МУРКО): проблемы аннотации, *in* Плунгян et al. [2009], pp. 175–214. Сборник статей.

Grishina, E. [2009]. Multimodal Russian Corpus (MURCO): General Structure and User Interface, *in* J. Levická & R. Garabík (eds), *NLP, Corpus Linguistics, Corpus Based Grammar Research. Fifth International Conference. Smolenice, Slovakia, 25-27 November 2009. Proceedings*, pp. 119–131.

Гришина, . . & Савчук, . . [2009]. Корпус устных текстов в НКРЯ: состав и структура, *in* Плунгян et al. [2009], pp. 129–149. Сборник статей.

Grishman, R. [2003]. Information Extraction, in *The Oxford Handbook of Computational Linguistics* Mitkov [2003], chapter 30, pp. 545–559.

Гришина, . ., Корчагин, . ., Плунгян, . . & Сичинава, . . [2009]. Поэтический корпус в рамках НКРЯ: общая структура и перспективы использования, *in* Плунгян et al. [2009], pp. 71–113. Сборник статей.

Gross, M. [1993]. Local Grammars and Their Representation by Finite Automata, *Data, Description, Discourse. Papers on the English Language in Honour of John McH. Sinclair* pp. 26–38.

Gross, M. [1997]. The Construction of Local Grammars, *in* Roche & Shabes [1997], pp. 329–354.
URL: *http://halshs.archives-ouvertes.fr/halshs-00278316*

Gucul-Milojević, S., Radulović, V. & Krstev, C. [2008]. Usage of NooJ Graphs and Annotation for Information Extraction, *in* X. Blanco & M. Silberztein (eds), *Proceedings of the 2007 International Nooj Conference*, Cambridge Scolars Publishing, pp. 103–120.

Hardie, A. [n.d.]. CQPweb — Combining Power, Flexibility and Usability in a Corpus Analysis Tool. Draft available online at `http://www.lancs.ac.uk/staff/hardiea/cqpwebpaper.pdf`.

Heiden, S. [2010]. The TXN Platform: Building Open-Source Textual Analysis Software Compatible with the TEI Encoding Scheme, *in* K. I. Ryo Otoguro (ed.), *24th Pacific Asia Conference on Language, Information and Computation*, Institute for Digital Enhancement of Cognitive Development, Waseda University, Sendai, Japan, pp. 389–398.
URL: *http://halshs.archives-ouvertes.fr/halshs-00549764/en*

Hoffmann, S., Evert, S., Smith, N., Lee, D. & Berglund Prytz, Y. [2008]. *Corpus Linguistics with BNCweb: A Practical Guide*, Peter Lang, Frankfurt am Main.

Horák, A., Gianitsová, L., Šimková, M., Šmotlák, M. & Garabík, R. [2004]. Slovak National Corpus, *in* P. Sojka, I. Kopeček & K. Pala (eds), *Text, Speech and Dialogue*, Vol. 3206 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg,

pp. 89–93.
URL: *http://dx.doi.org/10.1007/978-3-540-30120-2_12*

Hundt, M., Nesselhauf, N. & Biewer, C. (eds) [2007]. *Corpus Linguistics and the Web*, Rodopi, Amsterdam.

Hunston, S. [2002]. *Corpora in Applied Linguistics*, Cambridge University Press.

Hutchins, W. J. [1986]. *Machine Translation: Past, Present, Future*, Computers and Their Applications. Ellis Horwood Series in Engineering Science, Ellis Horwood, Chichester, UK.

Ide, N. [1998]. Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora, *First International Conference on Language Resources and Evaluation, LREC'98*, ELRA, Granada, pp. 463–470.
URL: *http://www.cs.vassar.edu/CES/*

Ide, N., Bonhomme, P. & Romary, L. [2000]. XCES: An XML-based Encoding Standard for Linguistic Corpora, *Second International Conference on Language Resources and Evaluation, LREC'00*.

Ide, N. & Romary, L. [2006]. Representing Linguistic Corpora and Their Annotations, *in* Calzolari et al. [2006].

Ide, N. & Véronis, J. [1994]. Multext (Multilingual Tools and Corpora), *Proceedings of the 15th International Conference on Computational Linguistics*, ACL, Kyoto, pp. 90–96.

ISS [2012]. *Interna pravila standardizacije — Deo 1: Donošenje, objavljivanje, održavanje, preispitivanje i povlačenje srpskih standarda i srodnih dokumenata.*
URL: *http://www.iss.rs/images/upload/IPS/ips_1_drugo_izdanje.pdf*

Ivanovska-Naskova, R. [2006]. Development of the First LRs for Macedonian: Current Projects, *in* Calzolari et al. [2006].

Jakubiček, M., Kilgarriff, A., McCarthy, D. & Rychlỳ, P. [2010]. Fast Syntactic Searching in Very Large Corpora for Many Languages, *PACLIC*, Vol. 24, pp. 741–747.

Janičić, P. & Marić, F. [2011]. *Programiranje I - Beleške sa predavanja*, Matematički fakultet, Beograd.
URL: *http://poincare.matf.bg.ac.rs/~filip/p1i/p1.pdf*

Janus, D. & Przepiórkowski, A. [2007a]. Poliqarp 1.0: Some Technical Aspects of a Linguistic Search Engine for Large Corpora, *in* J. Waliński, K. Kredens & S. Goźdź-Roszkowski (eds), *The Proceedings of Practical Applications of Linguistic Corpora PALC 2005*, Peter Lang, Frankfurt am Main.

Janus, D. & Przepiórkowski, A. [2007b]. Poliqarp: An Open Source Corpus Indexer and Search Engine with Syntactic Extensions, *Proceedings of the 45th Annual Meeting of the ACL 2007 Demo and Poster Sessions*, Association for Computational Linguistics, Prague, pp. 85–88.

Johansson, S. [2008]. Some Aspects of the Development of Corpus Linguistics in the 1970s and 1980s, *in* Lüdeling & Kytö [2008], chapter 3, pp. 33–53.

Jurafsky, D. & Martin, J. H. [2008]. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, second edn, Prentice Hall.

Karlsson, F. [2008]. Early Generative Linguistics and Empirical Methodology, *in* Lüdeling & Kytö [2008], chapter 2, pp. 14–32.

Karlsson, F., Voutilainen, A., Heikkilä, J. & Anttila, A. [1995]. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*, Vol. 4 of *Natural Language Processing*, De Gruyter Mouton.

Kennedy, G. [1998]. *An Introduction to Corpus Linguistics*, Studies in Language and Linguistics, Longman.

Kešelj, V. & Šipka, D. [2008]. A Suffix Subsumption-Based Approach to Building Stemmers and Lemmatizers for Highly Inflectional Languages with Sparse Resources, *INFOtheca* **9**(1–2): 23a–33a.

Kilgarriff, A. [2001]. Web As Corpus, *Proceedings of Corpus Linguistics 2001 Conference*, UCREL, Lancaster, pp. 342–344.
URL: *http://ucrel.lancs.ac.uk/publications/CL2003/CL2001%20conference/papers/kilgarri.pdf*

Kilgarriff, A. & Grefenstette, G. [2003]. Introduction to the Special Issue on the Web as Corpus, *Computational Linguistics* **29**(3): 333–347.

Kilgarriff, A., Rychlý, P., Smrz, P. & Tugwell, D. [2004]. The Sketch Engine, *in* G. Williams & S. Vessier (eds), *Proceedings of the Eleventh International Congress of Euralex 2004*, Université de Bretagne-Sud, Bretagne, France, pp. 105—-116.

Kiss, T. & Strunk, J. [2006]. Unsupervised Multilingual Sentence Boundary Detection, *Computational Linguistics* **32**(4): 485–525.
URL: *http://dx.doi.org/10.1162/coli.2006.32.4.485*

Kleene, S. C. [1951]. Representation of Events in Nerve Nets and Finite Automata, *Technical Report RM-704*, RAND Corporation. RAND Research Memorandum†.

Kleene, S. C. [1956]. Representation of Events in Nerve Nets and Finite Automata, *in* Claude Shannon and John McCarthy (ed.), *Automata Studies*, Princeton University Press, Princeton, NJ, pp. 3–41.

Klikovac, D. [2008]. *Jezik i moć: ogledi iz sociolingvistike i stilistike*, Biblioteka XX vek, Biblioteka XX vek.

Klimova, J. [1996]. Czech National Corpus, *Program and On-line Papers from the Tübingen East-West Computational Linguistics Meeting, Tübingen 16–27 September 1996.*
URL: *http://www.sfs.uni-tuebingen.de/~dm/events/EastWest96/cnc.html*

Koehn, P. [2010]. *Statistical Machine Translation*, Cambridge University Press, New York.

Koeva, S. & Genov, A. [2011]. Bulgarian Language Processing Chain, *Proceedings of Integration of Multilingual Resources and Tools in Web Applications. Workshop in Conjunction with GSCL*, Vol. 26.

Koeva, S., Krstev, C. & Vitas, D. [2008]. Morpho-semantic Relations in WordNet - a Case Study for two Slavic Languages, *in* A. T. et al (ed.), *Proceedings of the Global WordNet Conference (GWC 2008)*, University of Szeged, pp. 239–253.

Koeva, S., Stoyanova, I., Leseva, S., Dekova, R., Dimitrova, T. & Tarpomanova, E. [2012]. The Bulgarian National Corpus: Theory and Practice in Corpus Design, *Journal of Language Modelling* **1**(1): 65–110.

Kolkovska, S., Georgieva, C., Blagoeva, D. & Kostova, N. [2012]. The Bulgarian Nnational Corpus and its Application in Bulgarian Academic Lexicography, *Prace Filologiczne* **LXIII**: 37–50.

Kolkovska, S., Koeva, S. & Blagoeva, D. [2012]. Levels of Annotation in the Bulgarian National Corpus, *Prace Filologiczne* **LXIII**: 147–154.

Kostić, A. [2012]. Elektronski korpus srpskog jezika od XII do XVIII veka, *Nove tehnologije i standardi: Digitalizacija nacionalne baštine 2012. Jedanaesta nacionalna konferencija sa međunarodnim učešćem*, Matematički fakultet Univerziteta u Beogradu, pp. 12–13.

Kostić, D. [2001]. *Kvantitativni opis strukture srpskog jezika: Korpus srpskog jezika*, Institut za eksperimentalnu fonetiku i patologiju govora i Laboratorija za eksperimentalnu psihologiju Filozofskog fakulteta u Beogradu.

Kotsyba, N. [2013]. Praktyczny przewodnik po korpusach języka ukraińskiego, *in* M. Hebal-Jezierska (ed.), *Praktyczny przewodnik po korpusach języków słowiańskich*, Warsaw. (U pripremi).

Krek, S. [2012]. Legal Aspects of Corpora Compiling, In Corpora List Archive (Wed Oct 3 21:47:27 CEST 2012).

URL: *http://mailman.uib.no/public/corpora/2012-October/016271.html*

Kristal, D. [1996]. *Kembrička enciklopedija jezika*, Nolit, Beograd.

Krstev, C. [2008]. *Processing of Serbian – Automata, Text and Electronic Dictionaries*, Faculty of Philology, Belgrade.

Krstev, C. [2010]. Uloga i mesto slobodnog softvera u bibliotekama i javnom sektoru, *in* A. Vraneš & L. Marković (eds), *Zbornik radova sa međunarodne naučne konferencije ,,Etika u nauci i kulturi", 25-27. septembra 2009*, Filološki fakultet, Univerzitet u Beogradu, Beograd, Srbija, pp. 233–247.

Krstev, C. & Gucul, S. [2007]. Ka dečjem rečniku - automatska obrada dečje literature, *Bibliotekar: časopis za teoriju i praksu bibliotekarstva* **XLIX**(3–4): 295—-310.

Krstev, C., Jaćimović, J. & Vitas, D. [2012]. Recognition and Normalization of Some Classes of Named Entities in Serbian, *Proceedings of the Fifth Balkan Conference in Informatics*, BCI '12, ACM, New York, NY, USA, pp. 52–57.
URL: *http://doi.acm.org/10.1145/2371316.2371327*

Krstev, C., Pavlović-Lažetić, G., Obradović, I. & Vitas, D. [2003]. Corpora Issues in Validation of Serbian Wordnet, *in* V. Matousek & P. Mautner (eds), *TSD*, Vol. 2807 of *Lecture Notes in Computer Science*, Springer, pp. 132–137.

Krstev, C. & Vitas, D. [2005]. Corpus and Lexicon — Mutual Incompletness, *in* P. Danielsson & M. Wagenmakers (eds), *Proceedings of the Corpus Linguistics Conference, 14–17 July 2005, Birmingham.*
URL: *http://www.corpus.bham.ac.uk/PCLC/*

Krstev, C. & Vitas, D. [2007]. Treatment of Numerals in Text Processing, *in* Z. Vetulani (ed.), *Proceedings of 3nd Language & Technology Conference*, IMPRESJA Widawnictwa Elektroniczne S.A., pp. 418–422.

Krstev, C., Vitas, D. & Erjavec, T. [2004]. MULTEXT-East Resources for Serbian, *in* T. Erjavec & J. Zganec Gros (eds), *Zbornik 7. mednarodne multikonference ,,Informacijska druzba IS 2004" Jezikovne tehnologije 9-15 Oktober 2004, Ljubljana, Slovenija*, Institut „Jozef Stefan", Ljubljana, Slovenija.

Krstev, C., Vitas, D. & Gucul, S. [2005]. Recognition of Personal Names in Serbian Texts, *Proc. of the International Conference Recent Advances in Natural Language Processing*, pp. 21–23.

Krstev, C., Vitas, D., Maurel, D. & Tran, M. [2005]. Multilingual Ontology of Proper Names, *in* Z. Vetulani (ed.), *Proceedings of 2nd Language & Technology Conference (LTC'05)*, Wydawnictwo Poznańskie Sp. z o.o., Poznań, Poland, pp. 116–119.
URL: *http://www.matf.bg.ac.yu/~cvetana/biblio/ltc_048_krstev_4.pdf*

Krstev, C., Vitas, D., Obradović, I. & Utvić, M. [2011]. E-Dictionaries and Finite-State Automata for the Recognition of Named Entities, *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*, Association for Computational Linguistics, Blois, France, pp. 48–56.
URL: *http://www.aclweb.org/anthology/W11-4407*

Krstev, C., Vitas, D. & Savary, A. [2006]. Prerequisites for a comprehensive dictionary of serbian compounds, *in* T. Salakoski, F. Ginter, S. Pyysalo & T. Pahikkala (eds), *FinTAL*, Vol. 4139 of *Lecture Notes in Computer Science*, Springer, pp. 552–563.

Kučera, H. [2002]. Obituary for W. Nelson Francis, *Journal of English Linguistics* **30**(4): 306–309.

Laporte, E. & Monceaux, A. [1998]. Elimination of lexical ambiguities by grammars : The ELAG system, *Lingvisticæ Investigationes* **22**: 341–367.

LCL [2012]. *Statistics Used in the Sketch Engine: Documentation at http://www.sketchengine.co.uk*, Lexical Computing Ltd.

URL: *http://trac.sketchengine.co.uk/attachment/wiki/SkE/DocsIndex/ske-stat.pdf?format=raw*

Lee, D. Y. [2001]. Genres, Registers, Text Types, Domains and Styles: Clarifying the Concepts and Navigating a Path through the BNC Jungle, *Language Learning and Technology* **5**(3): 37–72.

Leech, G. [1991]. The State of the Art in Corpus Linguistics, *in* K. Aijmer & B. Altenberg (eds), *English Corpus Linguistics. Studies in Honour of Jan Svartvik*, Longman, London and New York, pp. 8–29.

Leech, G. [1993]. Corpus Annotation Schemes, *Literary and Linguistic Computing* **8**(4): 275–281.

Leech, G. [2007]. New Resources, or Just Better Old Ones? The Holy Grail of Representativeness, *in* Hundt et al. [2007], pp. 133–149.

Lehmberg, T. & Wörner, K. [2008]. Annotation Standards, *in* Lüdeling & Kytö [2008], chapter 22, pp. 484–501.

Léon, J. [2005]. Claimed and Unclaimed Sources of Corpus Linguistics, *Henry Sweet Society Bulletin* **44**: 36—-50.

Летучий, . . [2005]. Корпус диалектных текстов: задачи и проблемы, *in* NKRJ [2005], pp. 215–232. Сборник статей.

Летучий, . . [2009]. Диалектный корпус: состав и особенности разметки, *in* Плунгян et al. [2009], pp. 114–128. Сборник статей.

Lindquist, H. [2009]. *Corpus Linguistics and the Description of English*, Edinburgh Textbooks on English Language, Edinburgh University Press, Edinburgh.

Lüdeling, A. & Kytö, M. (eds) [2008]. *Corpus Linguistics: An International Handbook*, Vol. 1 of *Handbücher zur Sprach- und Kommunikationswissenschaft*, Mouton de Gruyter.

Malmkjær, K. (ed.) [2001]. *The Linguistics Encyclopedia*, 2 edn, Routledge (Taylor and Francis), New York.

Mandelbrot, B. [1954]. Structure formelle des textes et communication, *Word* **10**: 1–27.

Manning, C. D. & Schütze, H. [1999]. *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA.

Marković, D. [2009]. *Dizajn i implementacija tehnike kombinovanog pretraživanja teksta i slike u multimodalnim elektronskim poslovnim dokumentima*, Doktorska disertacija, Univerzitet Singidunum, Departman poslediplomskih studija, Beograd.

Maurel, D. [2008]. Prolexbase: A Multilingual Relational Lexical Database of Proper Names, *LREC*, European Language Resources Association.

Maurel, D. & Guenthner, F. [2005]. *Automata and Dictionaries*, Texts in computing, King's College.

Maurel, D., Vitas, D., Krstev, C. & Koeva, S. [2007]. Prolex: A Lexical Model for Translation of Proper Names. Application to French, Serbian and Bulgarian, *Bulag - Bulletin de Linguistique Appliquée et Générale, Les langues slaves et le français : approches formelles dans les études contrastives* **32**: 55–72.

McCarthy, M. & O'Keeffe, A. [2010]. Historical Perspective: What Are Corpora and How Have They Evolved?, *in* A. O'Keeffe & M. McCarthy (eds), *The Routledge Handbook of Corpus Linguistics*, Routledge, London and New York, pp. 3–13.

McCulloch, W. S. & Pitts, W. [1943]. A Logical Calculus of Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics* **5**(4): 115–133. Reprinted in *Neurocomputing: Foundations of Research, ed. by J. A. Anderson and E Rosenfeld. MIT Press 1988.*

McEnery, T. & Hardie, A. [2012]. *Corpus Linguistics: Method, Theory and Practice*, Cambridge Textbooks in Linguistics, Cambridge University Press.

McEnery, T. & Wilson, A. [2001]. *Corpus Linguistics*, Edinburgh Textbooks in Empirical Linguistics, Edinburgh University Press.

McEnery, T. & Wilson, A. [2011]. Corpus Linguistics, Information and Communications Technology for Language Teachers (ICT4LT).
URL: *http://www.ict4lt.org/en/en_mod3-4.htm*

McEnery, T., Xiao, R. & Tono, Y. [2006]. *Corpus-Based Language Studies: An Advanced Resource Book*, Routledge, London.

Mel'čuk, I. [1988]. *Dependency Syntax: Theory and Practice*, Suny Series in Linguistics, State University Press of New York.

Merialdo, B. [1994]. Tagging English Text with a Probabilistic Model, *Computational Linguistics* **20**(2): 155–171.

Meyer, C. [2008]. Pre-electronic Corpora, *in* Lüdeling & Kytö [2008], chapter 1, pp. 1–14.

Mikheev, A. [2000]. Tagging Sentence Boundaries, *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, Seattle, WA, pp. 264–271.

Mikheev, A. [2002]. Periods, Capitalized Words, etc., *Computational Linguistics* **28**(3): 289–318.

Mikheev, A. [2003]. Text Segmentation, in *The Oxford Handbook of Computational Linguistics* Mitkov [2003], chapter 10, pp. 201–218.

Mitkov, R. [2003]. *The Oxford Handbook of Computational Linguistics*, Oxford Handbooks in Linguistics, Oxford University Press.

Młodzki, R. & Przepiórkowski, A. [2011]. The WSD Development Environment, *Human Language Technology. Challenges for Computer Science and Linguistics*, Springer, pp. 224–233.

Nadeau, D. & Sekine, S. [2009]. A Survey of Named Entity Recognition and Classification, *in* S. Sekine & E. Ranchhod (eds), *Named Entities: Recognition, Classification and Use*, John Benjamins Pub. Co., Amsterdam/Philadelphia, pp. 3–28.

Nederhof, M.-J. [2000]. Practical Experiments with Regular Approximation of Context-Free Languages, *Computational Linguistics* **26**(1): 17–44.

Nguyen, T. M. H. & O'Rourke, S. [2003]. Concordancier (Alignement multilingue)), *Technical report*, LORIA.
URL: *led. loria. fr/ en_ outils. php. html# 5*

Nivre, J. [2008]. Treebanks, *in* Lüdeling & Kytö [2008], chapter 13, pp. 225–241.

NKRJ [2005]. *Национальный корпус русского языка: 2003—2005*, Индрик, Москва. Сборник статей.

Oakes, M. P. [1998]. *Statistics for Corpus Linguistics*, Edinburgh University Press, Edinburgh.

Obradović, I., Krstev, C., Pavlović-Lažetić, G. & Vitas, D. [2004]. Corpus Based Validation of WordNet Using Frequency Parameters, *in* P. Sojka, K. Pala, P. Smrž, C. Fellbaum & P. Vossen (eds), *Proceedings of the GWC : Second International WordNet Conference, Brno, Czech Republic, January 20–23, 2004*, Masaryk University, Brno, pp. 181–186.

Obradović, I., Stanković, R. & Utvić, M. [2008]. Integrisano okruženje za pripremu paralelizovanog korpusa, *in* B. Tošović (ed.), *Die Unterschiede zwischen dem Bosnischen/Bosniakischen, Kroatischen und Serbischen*, LitVerlag, Muenster, pp. 563–578. Zbornik 1. simpozijuma Razlike između bosanskog/bošnjačkog, hrvatskog i srpskog jezika.

Ogrodniczuk, M., Garabík, R., Koeva, S., Krstev, C., Pęzik, P., Pintér, T., Przepiórkowski, A., Szaszák, G., Tadić, M., Váradi, T. & Vitas, D. [2012]. Central and South-European Language Resources in META-SHARE, *INFOtheca* **13**(1): 3–26.
URL: *http: // infoteka. bg. ac. rs/ PDF/ Eng/ 2012-1/ INFOTHECA_ XIII_ 3_ may_ 3-26. pdf*

Palmer, D. D. [2010]. Text Preprocessing, *in* N. Indurkhya & F. Damerau (eds), *Handbook of Natural Language Processing*, 2 edn, Machine Learning & Pattern Recognition Series, CRC Press, Taylor and Francis Group, chapter 2, pp. 9–29.

Palmer, D. D. & Hearst, M. A. [1997]. Adaptive Multilingual Sentence Boundary Disambiguation, *Computational Linguistics* **23**: 241–267.

Paumier, S. [2011]. *Unitex 3.0 User Manual*.
URL: *http://www-igm.univ-mlv.fr/~unitex/UnitexManual3.0.pdf*

Petalinkar, S. [2011]. *Obrada elemenata logičkog izgleda teksta pod sistemom Unitex*, Master rad, Matematički fakultet, Univerzitet u Beogradu, Beograd.

Petrović, Lj. [2007]. *Teorija uzoraka i planiranje eksperimenata*, Centar za izdavačku delatnost Ekonomskog fakulteta.

Pešikan, M., Jerković, J. & Pižurica, M. [2009]. *Pravopis srpskog jezika*, Matica srpska, Novi Sad.

Pierce, J. R., Carroll, J. B., Hamp, E. P., Hays, D. G., Hockett, C. F., Oettinger, A. G. & Perlis, A. [1966]. Language and Machines — Computers in Translation and Linguistics. ALPAC Report, *Technical report*, National Academy of Sciences, National Research Council, Washington, DC. A report by the Automatic Language Processing Advisory Committee, Division of Behavioral Sciences.

Плунгян, . ., Рахилина, . . & Резникова, . . (eds) [2009]. *Национальный корпус русского языка: 2006—2008. Новые результаты и перспективы*, Нестор-История, Санкт-Петербург. Сборник статей.

Polovina, V. & Panić Cerovski, N. [2012]. Aspekti anotacije digitalnih video korpusa u lingvističkim istraživanjima, *in* A. Vraneš, L. Marković & G. Aleksander (eds), *Digitalizacija kulturne baštine, univerzitetski repozitorijumi i učenje na daljinu, knj. 3: Digitalni izvori u društveno-humanističkim istraživanjima*, Filološki fakultet, Beograd, pp. 15–22.

Polovina, V. & Panić Cerovski, N. [2013]. Prozodijska obeležja u javnom i neformalnom razgovoru, *in* S. Gudurić (ed.), *Jezici i kulture u vremenu i prostoru 1*, Filozofski fakultet, Novi Sad, pp. 441–450.

Popović, B. M., Sečujski, M., Delić, V., Janev, M. & Stanković, I. [2013]. Automatic Morphological Annotation in a Text-to-Speech System for Hebrew, *in* Zelezný et al. [2013], pp. 78–85.

Popović, Z. [2008]. *Evaluacija programa za obeležavanje (etiketiranje) teksta na srpskom jeziku*, Master rad, Matematički fakultet, Univerzitet u Beogradu, Beograd.

Popović, Z. [2010]. Programi za etiketiranje teksta na srpskom jeziku, *Infoteka* **XI**(2): 19–36.

Porter, M. F. [1980]. An Algorithm for Suffix Stripping, *Program: Electronic Library and Information Systems* **14**(3): 130–137.

Przepiórkowski, A. [2004]. *The IPI PAN Corpus: Preliminary Version*, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Przepiórkowski, A. [2009]. TEI/XCES, *FLaReNet/CLARIN Workshop on Standards at NEERI, 30 September 2009*, Helsinki.

Przepiórkowski, A., Górski, R. L., Łaziński, M. & Pęzik, P. [2010]. Recent Developments in the National Corpus of Polish, *in* Calzolari et al. [2010].

Przepiórkowski, A. & Murzynowski, G. [2009]. Manual Annotation of the National Corpus of Polish with Anotatornia, *The Proceedings of Practical Applications in Language and Computers PALC-2009*, Peter Lang, Frankfurt am Main.

Pustejovsky, J. & Stubbs, A. [2012]. *Natural Language Annotation for Machine Learning*, O'Reilly Media.

Рахилина, . . [2009]. Корпус как творческий проект, *in* Плунгян et al. [2009], pp. 7–26. Сборник статей.

Rehm, G., Schonefeld, O., Trippel, T. & Witt, A. [2010]. Sustainability of Linguistic Resources Revisited, *Proceedings of the International Symposium on XML for the*

Long Haul: Issues in the Long-term Preservation of XML. Balisage Series on Markup Technologies, Vol. 6.

Renouf, A. [2003]. WebCorp: Providing a Renewable Data Source for Corpus Linguists, *in* S. Granger & S. Petch-Tyson (eds), *Extending the Scope of Corpus-Based Research: New Applications, New Challenges*, Rodopi, Amsterdam, pp. 39–58.

Renouf, A., Kehoe, A. & Banerjee, J. [2006]. WebCorp: An Integrated System for Web Text Search, pp. 47–67.

Reppen, R. & Ide, N. [2004]. The American National Corpus: Overall Goals and the First Release, *Journal of English Linguistics* **32**(2): 105–113.

Reynar, J. C. & Ratnaparkhi, A. [1997]. A Maximum Entropy Approach to Identifying Sentence Boundaries, *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 16–19.

Rieger, B. [1979]. Repräsentativität: von der Unangemessenheit eines Begriffs zur Kennzeichnung eines Problems linguistischer Korpusbildung, *in* Bergenholtz & Schaeder [1979], pp. 52–70.

Roche, E. & Shabes, Y. (eds) [1997]. *Finite-State Language Processing*, MIT Press, Cambridge, MA, USA.

Romary, L. & Bonhomme, P. [2000]. Parallel Alignment of Structured Documents, *in* J. Véronis (ed.), *Parallel Text Processing*, pp. 233–253.

Rychlý, P. [2000]. *Korpusové manažery a jejich efektivní implementace*, PhD thesis, Fakulta informatiky, Masarykova univerzita, Brno.

Rychlý, P. [2007]. Manatee/Bonito — A Modular Corpus Manager, *in* P. Sojka & A. Horák (eds), *First Workshop on Recent Advances in Slavonic Natural Language Processing*, Masaryk University, Brno, pp. 65–70.
URL: *http://nlp.fi.muni.cz/raslan/2007/papers/12.pdf*

Sampson, G. [2003]. Thoughts on Two Decades of Drawing Trees, *in* Abeillé [2003], chapter 2, pp. 23–41.

Samuelsson, C. & Voutilainen, A. [1997]. Comparing a Linguistic and a Stochastic Tagger, *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Madrid, Spain, pp. 246–253.

Savary, A. [2005]. A Formalism for the Computational Morphology of Multi-Word Units, *Archives of Control Sciences* **15**(3): 437–449.

Savary, A., Waszczuk, J. & Przepiórkowski, A. [2010]. Towards the Annotation of Named Entities in the National Corpus of Polish, *in* Calzolari et al. [2010].

Савчук, . . & Сичинава, . . [2009]. Обучающий корпус русского языка и его использование в преподавательской практике, *in* Плунгян et al. [2009], pp. 317–334. Сборник статей.

Schmid, H. [1994]. Probabilistic Part-of-Speech Tagging Using Decision Trees, *Proceedings of International Conference on New Methods in Language Processing*, Vol. 12, Manchester, UK, pp. 44–49.

Schmid, H. [2000]. Unsupervised Learning of Period Disambiguation for Tokenisation, *Technical report*, IMS, University of Stuttgart.

Schmid, H. [2008]. Tokenizing and Part-of-Speech Tagging, *in* Lüdeling & Kytö [2008], chapter 24, pp. 527–551.

Scott, M. [2010]. *WordSmith Tools Help (Version 6).*
URL: *http://www.lexically.net/downloads/version6/HTML/index.html (last accessed 10.07.2012.)*

Scott, M. [2012a]. *Step-by-Step Guide to WordSmith.*
URL: *http://www.lexically.net/wordsmith/step_by_step_English6/index.html (last accessed 10.07.2012.)*

Scott, M. [2012b]. WordSmith Tools (Version 6).
URL: *http://www.lexically.net/wordsmith/version6/index.html (last accessed 10.07.2012.)*

Sekine, S. & Nobata, C. [2004]. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy, *LREC*, Lisbon, Portugal.

SEU [2010]. A Brief History of the Survey of English Usage.
URL: *http://www.ucl.ac.uk/english-usage/about/history.htm* *(last accessed 19.04.2011)*

Sečujski, M. [2009]. *Automatska morfološka anotacija tekstova na srpskom jeziku*, Doktorska disertacija, Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad.

Sečujski, M. & Delić, V. [2008]. A Software Tool for Automatic Part-of-Speech Tagging in Serbian Language, *Primenjena lingvistika* **9**(1): 97–103.

Sečujski, M., Obradović, R., Pekar, D., Jovanov, L. & Delić, V. [2002]. AlfaNum System for Speech Synthesis in Serbian Language, *in* P. Sojka, I. Kopecek & K. Pala (eds), *TSD*, Vol. 2448 of *Lecture Notes in Computer Science*, Springer, pp. 237–244.

Sečujski, M., Pekar, D. & Jakovljević, N. [2011]. Automatic Prosody Generation for Serbo-Croatian Speech Synthesis Based on Regression Trees, *INTERSPEECH*, ISCA, pp. 3157–3160.

Sharoff, S. [2006]. Methods and Tools for Development of the Russian Reference Corpus, *Language and Computers* **56**(1): 167–180.

Сичинава, . . [2005]. Национальный корпус русского языка: очерк предыстории, *in* NKRJ [2005], pp. 21–30. Сборник статей.

Silberztein, M. [1999]. Text Indexation with INTEX, *Computers and the Humanities* **33**(3): 265–280.

Silberztein, M. [2003]. *NooJ Manual*.
URL: *http://www.nooj4nlp.net/NooJManual.pdf*

Simić, R. & Jovanović, J. [2002]. *Osnovi teorije funkcionalnih stilova*, Jasen.

Sinclair, J. [1987]. *Looking Up: An Account of the COBUILD Project in Lexical Computing and the Development of the Collins COBUILD English Language Dictionary*, Collins ELT.

Sinclair, J. [1991]. *Corpus, Concordance, Collocation*, Oxford University Press.

Sinclair, J. & Ball, J. [1996]. EAGLES Preliminary Recommendations on Text Typology, EAGLES Documents EAG-TCWG-TTYP/P.

Sofer, M. [2006]. *The Translator's Handbook*, Translator's Handbook Series, Schreiber Pub.

Sperberg-McQueen, C. M. & Burnard, L. (eds) [1994]. *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*, Text Encoding Initiative, Chicago and Oxford.

Sperberg-McQueen, C. M. & Burnard, L. (eds) [2004]. *TEI P4 Guidelines for Electronic Text Encoding and Interchange — XML-compatible Edition*, Text Encoding Initiative, Chicago and Oxford.

Spoor, J. [1996]. The Copyright Approach to Copying on the Internet: (Over)Stretching the Reproduction Right?, *in* H. Hugenholtz (ed.), *The Future of Copyright in a Digital Environment*, Kluwer Law International, Dordrecht.

Stamatatos, E., Fakotakis, N. & Kokkinakis, G. [1999]. Automatic Extraction of Rules for Sentence Boundary Disambiguation, *Proceedings of the Workshop on Machine Learning in Human Language Technology*, Chania, Greece, pp. 88–92.

Stanković, R. [2009]. *Modeli ekspanzije upita nad tekstuelnim resursima*, Doktorska disertacija, Rudarsko-geološki fakultet, Beograd.

Stanković, R., Utvić, M., Vitas, D., Krstev, C. & Obradović, I. [2011]. On the Compatibility of Lexical Resources for NooJ, *in* K. Vučković, B. Bekavac & M. Silberztein (eds), *Automatic Processing of Various Levels of Linguistic Phenomena: Selected Papers from the 2011 International Nooj Conference*, Cambridge Scholars Publishing, pp. 96–108.

Stanojević, B. [2001]. Unicode kodne strane, fontovi i YU slova, *PC Press* **68**. Specijalni dodatak (jun 2001).

Stanojčić, Ž. & Popović, Lj. [2008]. *Gramatika srpskog jezika: za gimnazije i srednje škole*, Zavod za udžbenike, Beograd.

Stubbs, M. [2004]. Language Corpora, *in* A. Davies & C. Elder (eds), *The Handbook of Applied Linguistics*, Blackwell Handbooks in Linguistics, Blackwell, Malden, pp. 106–132.

Svartvik, J. (ed.) [1991]. *Directions in Corpus Linguistics. Proceedings of Nobel Symposium*, Vol. 31 of *Trends in Linguistics. Studies and Monographs*, Mouton de Gruyter, Berlin, New York.

Šimková, M. [2005]. Slovak National Corpus – History and Current Situation, *in* M. Šimková (ed.), *Insight into the Slovak and Czech Corpus Linguistics*, Veda, Bratislava, pp. 152–159.

Широков, ., Бугаков, ., Грязнухіна, ., Костишин, ., Кригін, ., Любченко, ., Рабулець, ., Сидоренко, ., Сидорчук, ., Шевченко, ., Шипнівська, . & Якименко, . [2005]. *Корпусна лінгвістика*, Довіра, Київ.

Широков, ., Сидорчук, ., Бугаков, . & Кригін, . [2011]. Застосування Українського національного лінгвістичного корпусу в лексикографії та лінгвістичних експертизах, *Українська лексикографія в загальнослов'янському контексті: теорія, практика, типологія*, Інститут української мови Національної академії наук України, Київ, pp. 285–294.

Tadić, M. [2002]. Building the Croatian National Corpus, *LREC2002 Proceedings, Las Palmas, ELRA, Pariz-Las Palmas*, Vol. 2, pp. 441–446.

Tadić, M. [2006]. Developing the Croatian National Corpus and Beyond, *Contributions to the Science of Text and Language*, Springer, pp. 295–300.

Tadić, M. [2009]. New Version of the Croatian National Corpus, *in* D. Hlaváčková, A. Horák, K. Osolsobě & P. Rychlý (eds), *After Half a Century of Slavonic Natural Language Processing*, Masaryk University, Brno, Czech Republic, pp. 199–205.

URL: *http://nlp.fi.muni.cz/publications/festkp2009/festkp2009.pdf*

Tanasijević, I., Sikimić, B. & Pavlović-Lažetić, G. [2012]. Multimedia Database of the Cultural Heritage of the Balkans, *in* N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dog(an, B. Maegaard, J. Mariani, J. Odijk & S. Piperidis (eds), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA), Istanbul, Turkey.

Taylor, A., Marcus, M. & Santorini, B. [2003]. The Penn Treebank: An Overview, *in* Abeillé [2003], chapter 1, pp. 5–22.

Taylor, C. [2008]. What is Corpus Linguistics? What the Data Says, *ICAME Journal, Computers in English Linguistics* **32**: 179–200.

Thompson, H. S., Beech, D., Maloney, M. & Mendelsohn, N. [2004]. *XML Schema Part 1: Structures Second Edition*. W3C Recommendation 28 October 2004.
URL: *http://www.w3.org/TR/xmlschema-1/*

Thompson, K. [1968]. Regular Expression Search Algorithm, *Communications of the ACM* **11**(6): 419–422.

TMX [2005]. *TMX 1.4b Specification. Open Standards for Container/Content Allowing Re-use (OSCAR) Recommendation.*
URL: *http://www.gala-global.org/oscarStandards/tmx/tmx14b.html (last accessed 20.07.2011)*

Tognini-Bonelli, E. & Sinclair, J. [2005]. Corpora, *in* Brown [2005], pp. 206–219.

Tošović, B. [2002]. *Funkcionalni stilovi*, Beogradska knjiga, Beograd.

Tran, M., Maurel, D., Vitas, D. & Krstev, C. [2005]. A French-Serbian Web Collaborative Work on a Multilingual Dictionary of Proper Names, Papillon 2005 Workshop on Multilingual Lexical Databases, in Association with the Sixth Symposium on Natural Language Processing (SNLP 2005).

Utvić, M. [2008]. *Konačni automati u regularnoj imenskoj derivaciji*, Magistarski rad, Matematički fakultet, Univerzitet u Beogradu, Beograd.

Utvić, M. [2011]. Annotating the Corpus of Contemporary Serbian, *INFOtheca* **12**(2): 36a–47a.

Váradi, T. [2001]. The Linguistic Relevance of Corpus Linguistics, *in* P. Rayson, A. Wilson, T. McEnery, A. Hardie & S. Khoja (eds), *In Proceedings of the Corpus Linguistics 2001 Conference*, Vol. 13, UCREL Technical Papers, Lancaster University, pp. 587–593.

Váradi, T. [2011]. Introducing the CESAR Project, *INFOtheca* **12**(1): 71–74.
URL: *http://infoteka.bg.ac.rs/PDF/Eng/2011-1/INFOTHECA_XII_1_August2011_71a-74a.pdf*

Vitas, D. [1981]. Generisanje imeničkih oblika u srpskohrvatskom jeziku, *Informatica* **3**(81): 49–55.

Vitas, D. [1993]. *Matematički model morfologije srpskohrvatskog jezika (imenska fleksija)*, Doktorska disertacija, Matematički fakultet, Univerzitet u Beogradu, Beograd.

Vitas, D. [2007]. Lokalne gramatike srpskog jezika, *Zbornik Matice srpske za slavistiku* **71–72**: 305–317.

Vitas, D. [2010]. Resursi i metode za obradu srpskog - stanje i perspektive, *in* B. Golubović & C. Voß (eds), *Srpska lingvistika / Serbische Linguistik. Eine Bestandsaufnahme*, Vol. 7 of *Studies on Language and Culture in Central and Eastern Europe (SLCCEE)*, Verlag Otto Sagner, München, pp. 257–277.

Vitas, D. (ed.) [1990]. *Matematička i računarska lingvistika: teorija i praksa: zbornik radova sa simpozijuma u organizaciji Društva za primenjenu lingvistiku Srbije : Beograd, 13. i 14. novembar 1987*, Drustvo za primenjenu lingvistiku Srbije.

Vitas, D., Koeva, S., Krstev, C. & Obradović, I. [2008]. Tour du monde through the dictionaries, *in* M. Constant, T. Nakamura, M. De Gioia & S. Vecchiato (eds),

Actes du 27eme Colloque International sur le Lexique et la Grammaire, L'Aquila, 10–13 septembre 2008, Universite Paris-Est, Institut Gaspard-Monge, pp. 249–256.

Vitas, D. & Krstev, C. [1998]. Electronic Edition of Serbian Translation of Orwell's *1984* Aligned with 7 Languages, *in* Erjavec, Lawson & Romary [1998].

Vitas, D. & Krstev, C. [2006]. Literature and aligned texts, *in* M. Slavcheva, G. Angelova & K. Simov (eds), *Readings in Multilinguality*, Institute for Parallel Processing, Bulgarian Academy of Sciences, Sofia, Bulgaria, pp. 148–155.

Vitas, D., Krstev, C. & Laporte, E. [2006]. Preparation and Exploitation of Bilingual Texts, *Lux Coreana* **1**: 110–132.

Vitas, D., Krstev, C. & Maurel, D. [2007]. A Note on the Semantic and Morphological Properties of Proper Names in the Prolex Project, *Lingvisticae Investigationes, Special issue on Named Entities: Recognition, Classification and Use* **30**(1): 115–134.

Vitas, D., Krstev, C., Obradović, I., Popović, Lj. & Pavlović-Lažetić, G. [2003]. Processing Serbian Written Texts: An Overview of Resources and Basic Tools, *in* S. Piperidis & V. Karkaletsis (eds), *Workshop on Balkan Language Resources and Tools, 21 Novembar 2003, Thessaloniki, Greece*, pp. 97–104.

Vitas, D. M. [2006]. *Prevodioci i interpretatori (uvod u teoriju i metode kompilacije programskih jezika)*, Matematički fakultet, Beograd.

Vitas, D., Nenadić, G. & Krstev, C. [1998]. Electronic Edition of Serbian Translation of Plato's *Republic* Aligned with 17 Languages, *in* Erjavec, Lawson & Romary [1998].

Vitas, D. & Popović, Lj. [2003]. Konspekt za izgradnju referentnog korpusa srpskog standardnog jezika, *Naučni sastanak slavista u Vukove dane, 12–16.9.2001.*, Vol. 31, MSC, Beograd, Novi Sad, pp. 221–227.

Wichmann, A. [2008]. Speech Corpora and Spoken Corpora, *in* Lüdeling & Kytö [2008], chapter 11, pp. 187–207.

Wilcock, G. [2009]. Introduction to Linguistic Annotation and Text Analytics, *Synthesis Lectures on Human Language Technologies* **2**(1): 1–159.

Witten, I., Moffat, A. & Bell, T. [1999]. *Managing Gigabytes: Compressing and Indexing Documents and Images*, The Morgan Kaufmann Series in Multimedia Information and Systems Series, Morgan Kaufmann Publishers.

Wittenburg, P. [2008]. Preprocessing Multimodal Corpora, *in* Lüdeling & Kytö [2008], chapter 31, pp. 664–685.

Woliński, M. [2006]. Morfeusz — A Practical Tool for the Morphological Analysis of Polish, *Intelligent Information Processing and Web Mining*, Springer, pp. 511–520.

Wynne, M. [2008]. Searching and Concordancing, *in* Lüdeling & Kytö [2008], chapter 33, pp. 706–737.

Xiao, R. [2008]. Well-Known and Influential Corpora, *in* Lüdeling & Kytö [2008], chapter 20, pp. 383–457.

Xiao, R. [2010]. Corpus Creation, *in* N. Indurkhya & F. Damerau (eds), *Handbook of Natural Language Processing*, 2 edn, Machine Learning & Pattern Recognition Series, CRC Press, Taylor and Francis Group, chapter 7, pp. 147–165.

Zelezný, M., Habernal, I. & Ronzhin, A. (eds) [2013]. *Speech and Computer - 15th International Conference, SPECOM 2013, Pilsen, Czech Republic, September 1-5, 2013. Proceedings*, Vol. 8113 of *Lecture Notes in Computer Science*, Springer.

Zečević, A. & Vujičić-Stanković, S. [2013]. The Mysterious Letter J, *Proceedings of the Workshop on Adaptation of Language Resources and Tools for Closely Related Languages and Language Variants*, INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria, pp. 40–44.
URL: *http://www.aclweb.org/anthology/W13-5307*

Zipf, G. K. [1929]. Relative Frequency as a Determinant of Phonetic Change, *Harvard Studies in Classical Philology* **40**: 1–95.

Zipf, G. K. [1935]. *The Psycho-Biology of Language*, Houghton Mifflin, Boston, MA.

Zipf, G. K. [1949]. *Human Behavior and the Principle of Least Effort*, Addison-Wesley, Cambridge, MA.

Dodaci

A

Donatori Korpusa savremenog srpkog jezika

Autorskim tekstovima, prevodima i svojim izdanjima Korpus savremenog srpskog jezika su pomogli sledeći autori, izdavači, mediji i arhivi:

Autori

- prof. dr Mirjana Đurđević, Visoka građevinsko-geodetska škola, Beograd
- prof. dr Duška Klikovac, Filološki fakultet, Beograd
- prof. dr Zoran Lučić, Matematički fakultet, Beograd
- akademik Dragoslav Mihailović, SANU
- prof. dr Nedeljko Parezanović, Matematički fakultet, Beograd
- prof. dr Ljubomir Popović, Filološki fakultet, Beograd
- prof. dr Ljiljana Subotić, Filozofski fakultet, Novi Sad
- prof. dr Aleksandra Vraneš, Filološki fakultet, Beograd

Izdavači, mediji i arhivi

- Medijska dokumentacija *Ebart*, Beograd

- *Politika*, Beograd
- *Službeni glasnik*, Beograd

B

Primer strukturno anotiranog teksta u formatu TEI/XML

```
<?xml version="1.0" encoding='UTF-8' standalone="no"?>
<!DOCTYPE TEI SYSTEM "teilite.dtd">
<TEI xmlns="http://www.tei-c.org/ns/1.0" xml:lang="sr">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Intervju sa Cyimamandom Ngozi Adicyi</title>
        <author>Darija Tundzxa</author>
      </titleStmt>
      <editionStmt>
        <edition>Elektronska verzija</edition>
        <respStmt>
          <resp>Obradila</resp>
          <name>Snxezxana Furundyicx</name>
        </respStmt>
      </editionStmt>
      <publicationStmt>
        <publisher>Studenti master studija pri
        Katedri za bibliotekarstvo i informatiku
```

```
    Filolosxkog fakulteta u Beogradu</publisher>
    <address>
      <addrLine>Studentski trg 3</addrLine>
      <addrLine>11000</addrLine>
      <addrLine>Beograd</addrLine>
    </address>
    <pubPlace>Beograd</pubPlace>
    <date>2012</date>
    <availability>
      <p>Copyright © MOSTOVI</p>
    </availability>
  </publicationStmt>
  <sourceDesc>
    <biblFull>
      <titleStmt>
        <title>Intervju sa Cyimamandom Ngozi Adicyi</title>
        <author>Darija Tundzxa</author>
        <respStmt>
          <resp>Preveo s engleskog</resp>
          <name>Igor Cvijanovicx</name>
        </respStmt>
      </titleStmt>
      <editionStmt>
        <edition>Broj 141-2, januar-jun 2008, Sveska 1</edition>
        <respStmt>
          <resp>Glavni urednik</resp>
          <name>Drinka Gojkovicx</name>
        </respStmt>
        <respStmt>
          <resp>Redakcija</resp>
          <name>Arpad Vicko</name>
          <name>Vladislava Gordicx-Petkovicx</name>
        </respStmt>
```

```
    <respStmt>
      <resp>Sekeretar redakcije</resp>
      <name>Mirna Uzelac</name>
    </respStmt>
    <respStmt>
      <resp>Graficyko oblikovanxe</resp>
      <name>Sxkart</name>
    </respStmt>
    <respStmt>
      <resp>Prelom</resp>
      <name>Studio Cyavka - Nebojsxa Cyovicx</name>
    </respStmt>
    <respStmt>
      <resp>Lektura i korektura</resp>
      <name>Zorica Galonxa</name>
    </respStmt>
    <respStmt>
      <resp>Kompjuterska priprema</resp>
      <name>Radovan Galonxa</name>
    </respStmt>
    <respStmt>
      <resp>Sxtampa</resp>
      <name>Zuhra</name>
    </respStmt>
  </editionStmt>
  <publicationStmt>
    <publisher> Udruzxenxe knxizxevnih prevodilaca Srbije
    </publisher>
    <address>
      <addrLine>Francuska 7</addrLine>
      <addrLine>11000</addrLine>
      <addrLine>Beograd</addrLine>
    </address>
```

```
            <pubPlace>Beograd</pubPlace>
            <date>2008</date>
            <availability>
              <p>copyright © MOSTOVI</p>
            </availability>
          </publicationStmt>
          <seriesStmt>
            <title>Mostovi</title>
            <title>Cyasopis za prevodnu knxizxevnost i probleme prevodxenxa
            </title>
            <idno type="ISSN">0350-6525</idno>
            <idno type="COBISS.SR-ID">621583</idno>
          </seriesStmt>
        </biblFull>
      </sourceDesc>
    </fileDesc>
    <revisionDesc>
      <change when="" who="" />
    </revisionDesc>
  </teiHeader>
  <text>
    <body>
             <pb n="30" />
      <div>
        <head>Intervju sa
Cyimamandom Ngozi Adicyi</head>
   <p>Darija Tundzxa: Kriticyari vas najcyesxcxe klasifikuju kao
    nigerijsku autorku, feministkinxu ili cyak afro-americyku spisatelxicu.
     Da li smatrate da takve generalizacije mogu da budu reduktivne ili
     mislite da je klasifikacija nesxto pozitivno, u smislu da to sxto ste
     "novi glas nigerijske knxizxevnosti", na primer, mozxe nadahnuti
      mladxe nigerijske pisce da krenu vasxim stopama?</p>
      <p>Cyimamanda Ngozi Adicyi: Mislim da su generalizacije
```

```
        uvek reduktivne zato sxto vas svode sa neke celine
        na samo jedan deo. Ja sam Nigerijka, feministkinxa,
       crnkinxa, Igbo i josx mnogo toga, ali kada me klasifikuju
       kao jedno, gotovo je nemogucxe da me vide i kao sve ostalo,
       a to me sputava.</p>

<!-- neke stranice su izostavljene radi jednostavnosti -->
 <pb n="33" />
        <p>Cy.N.A: Da. Nadala sam se da cxe cyitaoci
        tako shvatiti Kambili. Ona zaista naucyi
       da nxega i nxegov svet dovodi u pitanxe,
       ali mnoga nxena pitanxa ostaju
        ucxutkana strahom, lxubavlxu i rutinom.</p>
        <p>Januar 2005.</p>
        <p>Darija Tundzxa (Daria Tunca) je asistent na
        Odseku za engleski jezik i knxizxevnost Univerziteta u Lijezxu.
        Posebna oblast nxenog interesovanxa su postkolonijalne studije,
        pre svega africyki roman. Darija Tundzxa uredxuje
       online-bibliografiju Bena Okrija (cyije su pripovetke
       "U gradu crvene prasxine" i "Kad se svetla vrate"
       objavlxene u Mostovima br. 97 i 100) i
       website posvecxen delu Cyimamande Ngozi Adicyi
     (http://www.ulg.ac.be/facphl/uer/d-german/L3/cnaindedy.html).
     </p>
      </div>
    </body>
  </text>
</TEI>
```

C

XAlign

C.1 Ulazni dokument programa XAlign koji zadovoljava minimalni DTD

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE body SYSTEM "body.dtd">
<body>
<div>
<p>
<seg> BRISEL, Belgija -- </seg>
<seg> Turski premijer Redyep Tajip Erdogan u ponedelxak
(19. januara) boravi u Briselu, u pokusxaju da ozxivi
kandidaturu zemlxe za pridruzxivanxe EU. </seg>
<seg> On cxe razgovarati sa predsednikom Evropske komisije
Zxozeom Manuelom Barozom, visokim predstavnikom EU za
spolxnu politiku i bezbednost Havijerom Solanom,
predsednikom Evropskog parlamenta (EP) Hans-Gert Poteringom
i liderima politicykih grupa u EP. </seg>
<seg> Poseta se smatra istorijskom, jer je prva te vrste na
premijerskom nivou od 1996. i prva Erdoganova poseta Briselu
od samita EU u decembru 2004. godine, kojem je Turska
```

```
prisustvovala kao zemlxa kandidat. </seg>
<seg> Mada cxe razgovori biti fokusirani na procenu stanxa
odnosa Turske i EU, ostale teme verovatno cxe uklxucyivati
globalnu finansijsku krizu i situaciju u Gazi. </seg>
<seg> (Sabah, Zaman, Milijet, Hurijet - 20/01/09; novinska
agencija Anadolu, AFP, Rojters, Fajnensxel tajms - 19/01/09)
</seg> </p>
<p>
<seg> Bugarska pocyela tehnicyke pripreme za ponovno
otvaranxe nuklearnog reaktora </seg> </p>
<p>
<seg> SOFIJA, Bugarska -- </seg>
<seg> Tehnicyke pripreme za ponovno otvaranxe cyetvrtog
reaktora nuklearne elektrane Kozloduj su u toku, saopsxtio
je u petak (16. januara) premijer Sergej Stanisxev. </seg>
<seg> On je dodao da cxe Bugarska otvoriti reaktor samo u
saradnxi sa EU, mada nije precizirao kada ili kako cxe se
to desiti. </seg>
<seg> Vlada je odlucyila da ucyini to u svetlu tekucxe krize
vezane za prirodni gas. </seg>
<seg> Medxutim, ona bi mogla da se okoncya ove nedelxe,
nakon sporazuma Rusije i Ukrajine koji je postignut u
subotu. </seg>
<seg> S obzirom da bugarski sistem snabdevanxa
elektricynom energijom nije ugrozxen i da se izvoz struje
nastavio tokom krize, Evropska komisija je nagovestila da
ne odobrava bilo kakvo ponovno pokretanxe Jedinice
cyetiri. </seg>
<seg> (Darik, Rojters, Standart - 18/01/09;
 bTV, Sega, Dnevnik, Darik - 16/01/09) </seg> </p>
 </div>
</body>
```

C.2 Concordancier-formati programa XAlign

Concordancier-formati programa XAlign su dobili ime po programskom alatu za vizuelizaciju paralelizacije koji koristi te formate. Concordancier, omogućava ručnu korekciju automatski kreiranih veza između segmenata teksta na izvornom i ciljnom jeziku, kao i za pretragu biteksta (v. pododeljak Paralelizacija tekstova odeljka 6.5, str. 297).

IDifier

Poziv klase IDifier programa XAlign dodatno anotira ulazni tekst tako što svakom elementu tipa PHRASE (u ovom primeru elementu `seg`) automatski pridružuje XML-atribut `id` čija je vrednost jedinstveni identifikator u okviru teksta. U slučaju Concordancier-formata identifikator se sastoji od slova n i rednog broja segmenta u tekstu ($n1$ označava prvi segment, $n2$ — drugi, itd.).

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE body SYSTEM "body.dtd">
<body>
<div>
<p>
<seg id="n1"> BRISEL, Belgija -- </seg>
<seg id="n2"> Turski premijer Redyep Tajip Erdogan u ponedelxak
(19. januara) boravi u Briselu, u pokusxaju da ozxivi
kandidaturu zemlxe za pridruzxivanxe EU. </seg>
<seg id="n3"> On cxe razgovarati sa predsednikom Evropske komisije
Zxozeom Manuelom Barozom, visokim predstavnikom EU za
spolxnu politiku i bezbednost Havijerom Solanom,
predsednikom Evropskog parlamenta (EP) Hans-Gert Poteringom
i liderima politicykih grupa u EP. </seg>
<seg id="n4"> Poseta se smatra istorijskom, jer je prva te vrste na
premijerskom nivou od 1996. i prva Erdoganova poseta Briselu
od samita EU u decembru 2004. godine, kojem je Turska
```

```
prisustvovala kao zemlxa kandidat. </seg>
<seg id="n5"> Mada cxe razgovori biti fokusirani na procenu stanxa
odnosa Turske i EU, ostale teme verovatno cxe uklxucyivati
globalnu finansijsku krizu i situaciju u Gazi. </seg>
<seg id="n6"> (Sabah, Zaman, Milijet, Hurijet - 20/01/09; novinska
agencija Anadolu, AFP, Rojters, Fajnensxel tajms - 19/01/09)
</seg> </p>
<p>
<seg id="n7"> Bugarska pocyela tehnicyke pripreme za ponovno
otvaranxe nuklearnog reaktora </seg> </p>
<p>
<seg id="n8"> SOFIJA, Bugarska -- </seg>
<seg id="n9"> Tehnicyke pripreme za ponovno otvaranxe cyetvrtog
reaktora nuklearne elektrane Kozloduj su u toku, saopsxtio
je u petak (16. januara) premijer Sergej Stanisxev. </seg>
<seg id="n10"> On je dodao da cxe Bugarska otvoriti reaktor samo u
saradnxi sa EU, mada nije precizirao kada ili kako cxe se
to desiti. </seg>
<seg id="n11"> Vlada je odlucyila da ucyini to u svetlu tekucxe krize
vezane za prirodni gas. </seg>
<seg id="n12"> Medxutim, ona bi mogla da se okoncya ove nedelxe,
nakon sporazuma Rusije i Ukrajine koji je postignut u
subotu. </seg>
<seg id="n13"> S obzirom da bugarski sistem snabdevanxa
elektricynom energijom nije ugrozxen i da se izvoz struje
nastavio tokom krize, Evropska komisija je nagovestila da
ne odobrava bilo kakvo ponovno pokretanxe Jedinice
cyetiri. </seg>
<seg id="n14"> (Darik, Rojters, Standart - 18/01/09;
 bTV, Sega, Dnevnik, Darik - 16/01/09) </seg> </p>
 </div>
</body>
```

MultiAlign

1–1 uparivanje

```
<?xml version="1.0" standalone="no"?>
<linkGrp crdate="empty" domains="b1 b1" evaluate="all"
         source="balktime-en-sr-2009-01-19_f_id.xml"
         targFunc="null null" targOrder="Y" targType="seg"
         target="balktime-en-sr-2009-01-19_s_id.xml"
         type="alignment">
    <xptr from="ID (n1)" id="x1"></xptr>
    <xptr from="ID (n2)" id="x2"></xptr>
    <xptr from="ID (n3)" id="x3"></xptr>
    <xptr from="ID (n4)" id="x4"></xptr>
    <xptr from="ID (n5)" id="x5"></xptr>
    <xptr from="ID (n6)" id="x6"></xptr>
    <xptr from="ID (n7)" id="x7"></xptr>
    <xptr from="ID (n8)" id="x8"></xptr>
    <xptr from="ID (n9)" id="x9"></xptr>
    <xptr from="ID (n10)" id="x10"></xptr>
    <xptr from="ID (n11)" id="x11"></xptr>
    <xptr from="ID (n12)" id="x12"></xptr>
    <xptr from="ID (n13)" id="x13"></xptr>
    <xptr from="ID (n14)" id="x14"></xptr>
    <xptr from="ID (n15)" id="x15"></xptr>
    <link targets="n1 x1"></link>
    <link targets="n2 x2"></link>
    <link targets="n3 x3"></link>
    <link targets="n4 x4"></link>
    <link targets="n5 x5"></link>
    <link targets="n6 x6"></link>
    <link targets="n7 x7"></link>
    <link targets="n8 x8"></link>
    <link targets="n9 x9"></link>
```

```
    <link targets="n10 x10"></link>
    <link targets="n11 x11"></link>
    <link targets="n12 x12"></link>
    <link targets="n13 x13"></link>
    <link targets="n14 x14"></link>
    <link targets="n15 x15"></link>
</linkGrp>
```

uparivanje 2–1 i 1–2

```
<?xml version="1.0" encoding = "UTF-8"?>
<linkGrp crdate="empty" domains="b1 b1" evaluate="all"
         source="balktime-en-sr-2009-01-19_f_id.xml"
         targFunc="null null" targOrder="Y" targType="seg"
         target="balktime-en-sr-2009-01-19_s_id.xml"
         type="alignment">
<xptr from="ID (n15)" id="x1"></xptr>
<xptr from="ID (n14)" id="x2"></xptr>
<xptr from="ID (n13)" id="x3"></xptr>
<xptr from="ID (n12)" id="x4"></xptr>
<xptr from="ID (n11)" id="x5"></xptr>
<xptr from="ID (n10)" id="x6"></xptr>
<xptr from="ID (n9)" id="x7"></xptr>
<xptr from="ID (n8)" id="x8"></xptr>
<xptr from="ID (n7)" id="x9"></xptr>
<xptr from="ID (n1)" id="x10"></xptr>
<xptr from="ID (n4)" id="x11"></xptr>
<xptr from="ID (n2)" id="x12"></xptr>
<xptr from="ID (n3)" id="x13"></xptr>
<xptr from="ID (n5)" id="x14"></xptr>
<xptr from="ID (n6)" id="x15"></xptr>
<link id="l1" targets="n3 n4" type="linking"></link>
<link id="l2" targets="x14 x15" type="linking"></link>
<link targets="n15 x1"></link>
```

```
<link targets="n14 x2"></link>
<link targets="n13 x3"></link>
<link targets="n12 x4"></link>
<link targets="n11 x5"></link>
<link targets="n10 x6"></link>
<link targets="n9 x7"></link>
<link targets="n8 x8"></link>
<link targets="n7 x9"></link>
<link targets="n1 x10"></link>
<link targets="n5 x11"></link>
<link targets="n2 x12"></link>
<link targets="l1 x13"></link>
<link targets="n6 l2"></link>
</linkGrp>
```

C.3 Unitex-formati programa XAlign (Unitex)

Unitex-formate programa XAlign koristi Unitex, programski sistem za kreiranje i obradu korpusa primenom leksičkih resursa (v. odeljak 4.3, pododeljak NooJ i Unitex, str. 216). Ovaj format je zasnovan na Smernicama TEI P5 ([Burnard & Bauman, 2009: odeljak 16.4, str. 506–513]), tj. datoteka veza između jedinica prevođenja je TEI-dokument (verzija P5). Kao i u slučaju Concordancier-formata, za uparivanje segmenata koristi se element `link` i njegov atribut `targets` čija vrednost predstavlja par identifikatora jedinica prevođenja (segmenata ili skupova segmenata). Identifikatori jedinica prevođenja navedeni kao vrednosti atributa `targets` razdvojeni su razmakom. Identifikator jedinice prevođenja koju čini jedan segment sastoji se iz adrese dokumenta u kome se nalazi segment, simbola # i pozicije segmenta u XML-drvetu dokumenta izražene preko rednih brojeva elemenata `div` i `p` u kojima se segment nalazi, uključujući i redni broj samog segmenta u okviru elementa `p`. Npr., ako se segment nalazi u dokumentu *bt-en-sr-f.xml*, i to kao drugi po redu segment (`s2`) u petom po redu pasusu (`p5`) elementa `div` koji je četvrti po redu u dokumentu (`d4`), njegov identifikator biće `bt-en-sr-f.xml#d4p5s2`. Identifikator

jedinice prevođenja koju čini skup segmenata označava se slovom l i rednim brojem (npr. $l1$, $l2$, itd.).

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI>
   <teiHeader>
      <fileDesc>
         <titleStmt><title/></titleStmt>
         <publicationStmt><p/></publicationStmt>
         <sourceDesc>
            <bibl>
               <ref>
                  <ptr target="bt-en-sr_f.xml"/>
                  <note type="status">source</note>
               </ref>
            </bibl>
            <bibl>
               <ref>
                  <ptr target="bt-en-sr_s.xml"/>
                  <note type="status">translation</note>
               </ref>
            </bibl>
         </sourceDesc>
      </fileDesc>
   </teiHeader>
   <text>
      <body>
         <div type="resultXAlign">
            <linkGrp type="segmentGroup">
               <link
      targets="bt-en-sr_f.xml#d1p1s3 bt-en-sr_f.xml#d1p1s4 "
      type="linking" xml:id="l1"/>
               <link
```

```
targets="bt-en-sr_s.xml#d1p2s2 bt-en-sr_s.xml#d1p2s1 "
type="linking" xml:id="l2"/>
      </linkGrp>
      <linkGrp type="noCorresp">
      </linkGrp>
      <linkGrp type="alignment">
         <link
targets="bt-en-sr_f.xml#d1p1s1 bt-en-sr_s.xml#d1p1s1"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p1s2 bt-en-sr_s.xml#d1p1s2"
type="alignment"/>
         <link
targets="#l1 bt-en-sr_s.xml#d1p1s3"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p2s1 bt-en-sr_s.xml#d1p1s4"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p2s2 #l2"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p2s3 bt-en-sr_s.xml#d1p2s3"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s1 bt-en-sr_s.xml#d1p3s1"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s2 bt-en-sr_s.xml#d1p3s2"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s3 bt-en-sr_s.xml#d1p3s3"
type="alignment"/>
```

```
         <link
targets="bt-en-sr_f.xml#d1p3s4 bt-en-sr_s.xml#d1p3s4"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s5 bt-en-sr_s.xml#d1p3s5"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s6 bt-en-sr_s.xml#d1p3s6"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s7 bt-en-sr_s.xml#d1p3s7"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s8 bt-en-sr_s.xml#d1p3s8"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s9 bt-en-sr_s.xml#d1p3s9"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p3s10 bt-en-sr_s.xml#d1p3s10"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p4s1 bt-en-sr_s.xml#d1p4s1"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p5s1 bt-en-sr_s.xml#d1p5s1"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p5s2 bt-en-sr_s.xml#d1p5s2"
type="alignment"/>
         <link
targets="bt-en-sr_f.xml#d1p6s1 bt-en-sr_s.xml#d1p6s1"
type="alignment"/>
```

```
            <link
      targets="bt-en-sr_f.xml#d1p6s2 bt-en-sr_s.xml#d1p6s2"
      type="alignment"/>
            <link
      targets="bt-en-sr_f.xml#d1p6s3 bt-en-sr_s.xml#d1p6s3"
      type="alignment"/>
          </linkGrp>
        </div>
      </body>
    </text>
</TEI>
```

D

TMX

D.1 DTD za TMX 1.4

```
<!-- TMX (Translation Memory eXchange)

Known as "-//LISA OSCAR:1998//DTD for Translation Memory eXchange//EN"
Use in TMX: <!DOCTYPE tmx SYSTEM "tmx14.dtd">

An SGML application conforming to:
- International Standard ISO 8879 Standard Generalized Markup Language,
- XML (Extensible Markup Language), W3C Recommendation

All TMX element and attribute names must be in lowercase.

-->

<!ENTITY % segtypes    "block|paragraph|sentence|phrase" >

<!-- Base Document Element -->
   <!ELEMENT tmx             (header, body) >
   <!ATTLIST tmx
```

```
          version             CDATA          #FIXED "1.4" >

<!-- Header -->
   <!ELEMENT header           (note|prop|ude)* >
   <!ATTLIST header
          creationtool        CDATA          #REQUIRED
          creationtoolversion CDATA          #REQUIRED
          segtype             (%segtypes;)   #REQUIRED
          o-tmf               CDATA          #REQUIRED
          adminlang           CDATA          #REQUIRED
          srclang             CDATA          #REQUIRED
          datatype            CDATA          #REQUIRED
          o-encoding          CDATA          #IMPLIED
          creationdate        CDATA          #IMPLIED
          creationid          CDATA          #IMPLIED
          changedate          CDATA          #IMPLIED
          changeid            CDATA          #IMPLIED >

<!-- Body -->
   <!ELEMENT body             (tu*) >
   <!-- No attributes             -->

<!-- Note -->
   <!ELEMENT note             (#PCDATA) >
   <!ATTLIST note
          o-encoding          CDATA          #IMPLIED
          xml:lang            CDATA          #IMPLIED
          lang                CDATA          #IMPLIED >
          <!-- lang is deprecated: use xml:lang -->

<!-- User-defined Encoding -->
   <!ELEMENT ude              (map+) >
   <!ATTLIST ude
```

```
         name                CDATA          #REQUIRED
         base                CDATA          #IMPLIED >
<!-- Note: the base attribute is required if one or more <map>
     elements in the <ude> contain a code attribute. -->

<!-- Character mapping -->
   <!ELEMENT map             EMPTY >
   <!ATTLIST map
         unicode             CDATA          #REQUIRED
         code                CDATA          #IMPLIED
         ent                 CDATA          #IMPLIED
         subst               CDATA          #IMPLIED >

<!-- Property -->
   <!ELEMENT prop            (#PCDATA) >
   <!ATTLIST prop
         type                CDATA          #REQUIRED
         xml:lang            CDATA          #IMPLIED
         o-encoding          CDATA          #IMPLIED
         lang                CDATA          #IMPLIED >
         <!-- lang is deprecated: use xml:lang -->


<!-- Translation Unit -->
   <!ELEMENT tu              ((note|prop)*, tuv+) >
   <!ATTLIST tu
         tuid                CDATA          #IMPLIED
         o-encoding          CDATA          #IMPLIED
         datatype            CDATA          #IMPLIED
         usagecount          CDATA          #IMPLIED
         lastusagedate       CDATA          #IMPLIED
         creationtool        CDATA          #IMPLIED
         creationtoolversion CDATA          #IMPLIED
```

```
          creationdate        CDATA          #IMPLIED
          creationid          CDATA          #IMPLIED
          changedate          CDATA          #IMPLIED
          segtype             (%segtypes;)   #IMPLIED
          changeid            CDATA          #IMPLIED
          o-tmf               CDATA          #IMPLIED
          srclang             CDATA          #IMPLIED >

<!-- Translation Unit Variant -->
   <!ELEMENT tuv              ((note|prop)*, seg) >
   <!ATTLIST tuv
          xml:lang            CDATA          #REQUIRED
          o-encoding          CDATA          #IMPLIED
          datatype            CDATA          #IMPLIED
          usagecount          CDATA          #IMPLIED
          lastusagedate       CDATA          #IMPLIED
          creationtool        CDATA          #IMPLIED
          creationtoolversion CDATA          #IMPLIED
          creationdate        CDATA          #IMPLIED
          creationid          CDATA          #IMPLIED
          changedate          CDATA          #IMPLIED
          o-tmf               CDATA          #IMPLIED
          changeid            CDATA          #IMPLIED
          lang                CDATA          #IMPLIED >
          <!-- lang is deprecated: use xml:lang -->

<!-- Text -->
   <!ELEMENT seg              (#PCDATA|bpt|ept|ph|it|hi|ut)* >


<!-- Content Markup ================================================= -->

   <!ELEMENT bpt              (#PCDATA|sub)* >
```

```
<!ATTLIST bpt
       i                  CDATA          #REQUIRED
       x                  CDATA          #IMPLIED
       type               CDATA          #IMPLIED >

<!ELEMENT ept             (#PCDATA|sub)* >
<!ATTLIST ept
       i                  CDATA          #REQUIRED >

<!ELEMENT sub             (#PCDATA|bpt|ept|it|ph|hi|ut)* >
<!ATTLIST sub
      datatype            CDATA          #IMPLIED
      type                CDATA          #IMPLIED >

<!ELEMENT it              (#PCDATA|sub)* >
<!ATTLIST it
       pos                (begin|end)    #REQUIRED
       x                  CDATA          #IMPLIED
       type               CDATA          #IMPLIED >

<!ELEMENT ph              (#PCDATA|sub)* >
<!ATTLIST ph
       x                  CDATA          #IMPLIED
       assoc              CDATA          #IMPLIED
       type               CDATA          #IMPLIED >

<!ELEMENT hi              (#PCDATA|bpt|ept|it|ph|hi|ut)* >
<!ATTLIST hi
       x                  CDATA          #IMPLIED
       type               CDATA          #IMPLIED >


<!-- The <ut> element is deprecated -->
```

```
   <!ELEMENT ut              (#PCDATA|sub)* >
   <!ATTLIST ut
          x                  CDATA          #IMPLIED >


<!-- End -->
```

D.2 XSL-transformacija formata TMX u format HTML

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
     xmlns:xsd="http://tempuri.org/Dataset1.xsd" version="1.0">
  <xsl:param name="lang1FullName"/>
  <xsl:param name="lang2FullName"/>
  <xsl:template match="/tmx">
    <html>
      <head>
        <meta http-equiv="Content-Type"
              content="text/html; charset=utf-8"/>
        <link rel="stylesheet" href="tmx.css"/>
        <title>ACIDE</title>
      </head>
      <body>
        <table>
          <tr>
            <td colspan="2">
                <xsl:value-of select="body/tu/prop"/>
            </td>
          </tr>
          <tr>
```

```
            <td>
                <xsl:value-of select="$lang1FullName"/>
                <xsl:value-of select="body/tu/tuv[1]/@xml:lang"/>)
            </td>
            <td>
                <xsl:value-of select="$lang2FullName"/>
               (<xsl:value-of select="body/tu/tuv[2]/@xml:lang"/>)
            </td>
          </tr>
          <xsl:for-each select="body/tu">
            <tr>
              <xsl:for-each select="tuv">
                <td>
                    <xsl:value-of select="attribute::creationid"/>:
                    <xsl:copy-of select="seg"/>
                </td>
              </xsl:for-each>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

D.3 Razlaganje TMX-datoteke na datoteke izvornog i ciljnog jezika

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output encoding="UTF-8" method="xml"
              omit-xml-declaration="no"/>
```

```
        <xsl:param name="langValue"/>

        <xsl:variable name="newline">
    <xsl:text>
</xsl:text>
  </xsl:variable>

  <xsl:template match="tmx">
    <text>
      <xsl:value-of select="$newline"/>
      <xsl:apply-templates select="body/tu"/>
    </text>
    <xsl:value-of select="$newline"/>
  </xsl:template>

  <xsl:template match="tu">
    <tu>
      <xsl:value-of select="$newline"/>
      <xsl:for-each select="tuv[@xml:lang=$langValue]">
        <xsl:apply-templates select="seg"/>
      </xsl:for-each>
    </tu>
    <xsl:value-of select="$newline"/>
  </xsl:template>

  <xsl:template match="seg">
    <seg>
      <xsl:value-of select="$newline"/>
      <xsl:value-of select="."/>
      <xsl:value-of select="$newline"/>
    </seg>
    <xsl:value-of select="$newline"/>
```

```
  </xsl:template>
</xsl:stylesheet>
```

E

Konverzije

E.1 Makro za konverziju kodnog rasporeda u programu Microsoft Word

```
Attribute VB_Name = "Module1"
Sub snimiCistTekst(novoIme As String, _
                   izlazniKodniRaspored As MsoEncoding)
  ActiveDocument.SaveAs FileName:=novoIme, _
                        Encoding:=izlazniKodniRaspored, _
                        Fileformat:=wdFormatText
End Sub

Sub paketnaObrada()
  Dim oDokument As Word.Document
  Dim sAdresaKataloga As String
  Dim sSpecifikacija As String
  Dim sImeDatoteke As String
  Dim sTipDatoteke As String
  Dim sAdresaDatoteke As String
  Dim sAdresaNoveDatoteke As String
  Dim sListaDatoteka() As String
```

```
Dim i As Integer
Dim sPoruka As String

' Unos adrese kataloga
sPoruka = "Uneti adresu kataloga cije datoteke treba obraditi"
sAdresaKataloga = InputBox(sPoruka)


If FileOrDirExists(sAdresaKataloga) Then
    ' Uklanjanje eventualnih belina iz adrese
    sAdresaKataloga = Trim(sAdresaKataloga)
    ' Poslednji karakter u adresi kataloga mora da bude '\'
    If Right$(sAdresaKataloga, 1) <> "\" Then
      sAdresaKataloga = sAdresaKataloga & "\"
    End If

    sPoruka = "Uneti specifikaciju tipa datoteka koje treba obraditi"
    sSpecifikacija = InputBox(sPoruka) '"*.txt" ili "*.doc"
    If sSpecifikacija <> "*.txt" Or sSpecifikacija <> "*.doc" Then
      MsgBox "Nedozvoljeni tip datoteke"
      End
    End If
    ' Dopisivanje odgovarajuceg tipa datoteke
    sSpecifikacija = sAdresaKataloga & sSpecifikacija

    ' Obrada svake datoteke kataloga koja zadovoljava specifikaciju
    sImeDatoteke = Dir$(sSpecifikacija)

    While sImeDatoteke <> ""
      sAdresaDatoteke = sAdresaKataloga & sImeDatoteke
      sTipDatoteke = Right(sImeDatoteke, 4)

      If sTipDatoteke = ".doc" Then
```

```
            sAdresaNoveDatoteke = sAdresaKataloga & "\" & Left(sImeDatoteke, Len(sImeDatote
        ElseIf sTipDatoteke = ".txt" Then
            sAdresaNoveDatoteke = sAdresaDatoteke
        End If

        Set oDokument = Word.Documents.Open(sAdresaDatoteke)
        snimiCistTekst novoIme:=sAdresaNoveDatoteke, _
                       izlazniKodniRaspored:=msoEncodingUTF8
        oDokument.Close (True)
        sImeDatoteke = Dir$
      Wend
  Else
    MsgBox "Zadati katalog ne postoji"
  End If
End Sub

' Funkcija FileOrDirExists je bez izmena preuzeta sa portala VBA Express
' http://www.vbaexpress.com/kb/getarticle.php?kb_id=559
Function FileOrDirExists(PathName As String) As Boolean
     'Macro Purpose: Function returns TRUE if the specified file
     '               or folder exists, false if not.
     'PathName     : Supports Windows mapped drives or UNC
     '             : Supports Macintosh paths
     'File usage   : Provide full file path and extension
     'Folder usage : Provide full folder path
     '               Accepts with/without trailing "\" (Windows)
     '               Accepts with/without trailing ":" (Macintosh)

    Dim iTemp As Integer

     'Ignore errors to allow for error evaluation
    On Error Resume Next
    iTemp = GetAttr(PathName)
```

```
     'Check if error exists and set response appropriately
    Select Case Err.Number
    Case Is = 0
        FileOrDirExists = True
    Case Else
        FileOrDirExists = False
    End Select

     'Resume error checking
    On Error GoTo 0
End Function
```

E.2 Izvod iz izvornog koda za konverziju kodnog rasporeda u programu CorpusPreprocessor

```
public abstract class EncodingConvertor
{
    protected String m_sFileIn;
    protected String m_sFileOut;
    public abstract void Convert(String sEncIn, String sEncOut);
    public abstract void Convert(Encoding encIn, Encoding encOut);
    protected EncodingConvertor(String sFileIn, String sFileOut)
    {
        m_sFileIn = sFileIn;
        m_sFileOut = sFileOut;
    }
}

public class TextEncodingConvertor: EncodingConvertor
{
    public TextEncodingConvertor(String sFilename, String sFileOut)
    :base(sFilename, sFileOut) {}
```

```
    override
    public void Convert(String sInEncoding, String sOutEncoding)
    {
        Encoding encIn = Encoding.GetEncoding(sInEncoding);
        Encoding encOut = Encoding.GetEncoding(sOutEncoding);
        Convert(encIn, encOut);
    }

    override
    public void Convert(Encoding encIn, Encoding encOut)
    {
        FileStream fsIn
            = new FileStream(m_sFileIn, FileMode.Open);
        FileStream fsOut
            = new FileStream(m_sFileOut, FileMode.Create);
        StreamReader sr = new StreamReader(fsIn, encIn);
        StreamWriter sw = new StreamWriter(fsOut, encOut);
        String sLineIn;

        while((sLineIn = sr.ReadLine()) != null)
        {
            byte [] inputEncBytes = encIn.GetBytes(sLineIn);
            byte [] outputEncBytes
                = Encoding.Convert(encIn, encOut, inputEncBytes);
            string sLineOut = encOut.GetString(outputEncBytes);
            sw.WriteLine(sLineOut);
        }

        sr.Close();
        sw.Close();
    }
}
```

F

Terminološki rečnik

F.1 Englesko-srpski

aligned corpora	paralelizovani korpusi
aligned text	paralelizovani tekst
balance	balansiranost
balanced corpus	balansirani korpus
batch processing	paketna obrada
bilingual corpora	dvojezični korpusi
bitext	bitekst
chunking	plitko parsiranje
collocation	kolokacija
comparable corpora	uporedni (komparativni) korpusi
computational linguistics	računarska lingvistika
contrastive corpora	kontrastni korpusi
corpus linguistics	korpusna lingvistika
concordance	konkordanca
corpus	korpus
corpus-based	zasnovan na korpusu
diachronic corpora	dijahroni korpusi

dynamic corpora	dinamički korpusi
electronic corpora	elektronski korpusi
empty string	prazna niska
encoding	kodni raspored
formal language theory	teorija formalnih jezika
frequency list	lista učestanosti
general corpora	opšti korpusi
genre	žanr
geographical corpora	geografski korpusi
history corpora	istorijski korpusi
lemma	lema
lemmatization	lematizacija
lexeme	leksema
linguistic competence	jezička sposobnost
linguistic performance	govorna delatnost
literal	obični karakter, literal
markup	obeležavanje, označavanje, anotacija
machine translation	automatsko (mašinsko) prevođenje
metacharacter	metakarakter
monitor corpora	monitor-korpusi
monolingual corpora	jednojezični korpusi
morphosyntactic description	morfosintaksički opis
multilingual corpora	višejezični korpusi
multimodal corpora	multimodalni korpusi
natural language processing	obrada prirodnog jezika
native speaker	izvorni govornik
parsing	parsiranje, sintaksička analiza
parallel corpora	paralelizovani korpusi
parallel text	paralelizovani tekst
part of speech	vrsta reči
pattern	obrazac

pre-electronic corpora	preelektronski korpusi
preprocessing	predobrada
register	registar
representative corpus	reprezentativni korpus
sample	uzorak
sample unit	jedinični uzorak, jedinica uzorka
sampling	uzorkovanje
sampling frame	okvir uzorkovanja, okvir uzorka
sentence boundary disambiguation	identifikacija kraja rečenice
sentence breaking	segmentacija na rečenice
shallow parsing	plitko parsiranje
simple random sampling	prost slučajan uzorak
specialized corpora	specijalizovani korpusi
spoken corpora	korpusi govornih tekstova
spoken text	govorni tekst
static corpora	statički korpusi
string	niska
synchronic corpora	sinhroni korpusi
tag	obeležje, etiketa
tagging	anotacija, obeležavanje, etiketiranje
text acquisition	prikupljanje tekstova
token	token
tokenization	tokenizacija
topic corpora	tematski korpusi
translation corpora	paralelizovani korpusi
translation memory	prevodilačka memorija
treebank	banka sintaksičkih drveta
type	tip
variation	varijetet
word class	vrsta reči
word form	formalna (tekstuelna) reč

word list	lista reči
word token	korpusna reč
word type	korpusni tip
written corpora	korpusi pisanih tekstova
written text	pisani tekst

Biografija autora

Miloš Utvić je rođen 29. januara 1976. godine u Prokuplju. Osnovnu školu i gimnaziju je završio u Kuršumliji kao đak generacije. Diplomirao je 29. maja 2001. godine na Matematičkom fakultetu u Beogradu na smeru računarstvo i informatika sa prosečnom ocenom 9,34 (devet i 34/100). Stepen magistra računarstva je stekao 18. februara 2008. godine odbranivši magistarski rad sa temom *Konačni automati u regularnoj imenskoj derivaciji* na Matematičkom fakultetu Univerziteta u Beogradu.

Od 3. decembra 2001. godine radi kao asistent-pripravnik na Katedri za bibliotekarstvo i informatiku na Filološkom fakultetu u Beogradu. U zvanje asistenta na Filološkom fakultetu Univerziteta u Beogradu je biran 15. oktobra 2008. godine i ponovo 12. oktobra 2011. godine. Sada je asistent za predmete *Informatika za bibliotekare 1–2*, *Informatički praktikum 1–4*, *Digitalni tekst 1–2*, *Baze podataka i bibliotečki informacioni sistemi*, *Pronalaženje informacija*, *Multimedijalni dokumenti* (osnovne studije) i *Elektronsko izdavaštvo i digitalne biblioteke* (master studije).

Član je Grupe za jezičke tehnologije Univerziteta u Beogradu, kao i Komisije za korpus Odbora za standardizaciju srpskog jezika SANU. Trenutno je angažovan na dva domaća i jednom stranom projektu. Od programskih rešenja je razvio desetak podsistema u okviru Korpusa savremenog srpskog jezika, a u saradnji sa dr Rankom Stanković i integrisano okruženje za pripremu paralelizovanih korpusa. Učestvovao je na nekoliko radionica, konferencija i simpozijuma i prezentovao više radova.

Njegove uže oblasti interesovanja su bibliotečka informatika i računarska lingvistika, posebno računarska leksikografija i korpusna lingvistika.

Od stranih jezika govori engleski jezik, a služi se i ruskim jezikom.

Прилог 1.

Изјава о ауторству

Потписани-а Милош В. Утвић

број индекса _____

Изјављујем

да је докторска дисертација под насловом

Изградња референтног корпуса савременог српског језика

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, 13.12.2013.

Утвић Милош

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Милош Утвић

Број индекса _____

Студијски програм _____

Наслов рада Изградња референтног корпуса савременог српског језика

Ментор др Цветана Крстев, ванредни професор

Потписани/а Милош Утвић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 13.12.2013.

Утвић Милош

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

ИЗГРАДЊА РЕФЕРЕНТНОГ КОРПУСА САВРЕМЕНОГ СРПСКОГ ЈЕЗИКА

која је моје ауторско дело.

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

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прераде

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, 13.12.2013.

Милош Јакимовић

1. Ауторство - Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство - некомерцијално – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство - некомерцијално – делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство - делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.