MATEMATIČKI FAKULTET
Univerzitet u Beogradu

MASTER TEZA

Tema: Obrada elemenata logičkog izgleda teksta pod sistemom Unitex

Student: Saša Petalinkar
Broj indeksa: 1188/10

Sadržaj

Uvod

Korpus [3] u lingvistici je veliki i strukturisani skup tekstova. Sada su korpusi obično elektronski obrađeni i sačuvani. Oni se koriste za statističku analizu, proveru hipoteza, proveravanje pojava ili potvrđivanje jezičkih pravila u specifičnim univerzumima.

Korpus može da sadrži tekstove na jednom jeziku (jednojezični korpus) ili tekstualne podatke na više jezika (višejezični korpus). Višejezični korpus koji je specijalno oblikovan za uporedno poređenje naziva se paralelni korpus.

Da bi korpus bio korisniji za lingvistička istraživanja, on često biva anotiran. Uopšteno anotacija je proces dodavanja beležaka tokom čitanja teksta. U lingvistici ona uključuje komentare i metapodatke[1]. Jedan primer obeležavanja korpusa je označavanje vrsta reči, takozvano POS-označavanje[2], u kome se gramatičke informacije o svakoj reči (glagol, imenica, pridev, itd.) dodaju korpusu u obliku oznaka. Drugi primer dodaje lemu, osnovni oblik, svakoj reči[3]. Kada je potrebno omogućiti da korpus koriste lingvisti koji ne vladaju jezikom u kome je taj korpus napisan, koristi se međuredno tumačenje (*interlinear glossing*) da bi komentar postao dvojezičan.

Na neke korpuse su primenjeni dalji nivoi analize. Konkretno, manji korpus može biti u potpunosti sintaksički i sematički obrađen. Takav korpus se obično naziva banka drveta (*treebank*) ili parsirani korpus. Usled teškoće da se obezbedi da čitav korpus ima potpune i dosledne komentare obično sledi da su ti korpusi manji, sa veličinom od nekoliko miliona reči. Moguća je analiza na ostalim jezičkim nivoima, uključujući i beleške za morfologiju, semantiku i pragmatiku.

Korpusi su glavna baza znanja u oblasti korpusne lingvistike. Analiza i obrada različitih tipova korpusa su takođe predmet mnogih radova u oblasti računarske lingvistike, prepoznavanja govora i mašinskog prevođenja, gde se često koriste za kreiranje skrivenih Markovljevih modela za POS-označavanje i druge svrhe. Korpusi i liste frekvencija izvedene iz njih su takođe korisne i za učenje jezika.

Korpusi se takođe koriste u proučavanju istorijskih dokumenata, na primer, u pokušaju da se dešifruju drevni rukopisi.

Konkordanca [4] je azbučni spisak reči koje se koriste u knjizi ili korpusu, sa njihovim neposrednim kontekstom. Zbog vremena, teškoća i troškova koji su potrebni da bi se napravila konkordanca tokom pre-računarske ere su samo za tekstove od posebnog značaja, kao što su Vede, Biblija, Kuran i dela Šekspira, pravljene konkordance.

Čak i uz korišćenje računara, proizvodnja konkordance (bilo na papiru ili u računaru) može da zahteva mnogo ručnog rada, jer se često uključuje dodatni materijal, na primer komentar ili definicije indeksiranih reči i tematsko unakrsno indeksiranje koje još uvek nije moguće automatizavati.

Konkordanca se često koristi u lingvistici, tokom proučavanja tekstova, na primer: poređenja različitih upotreba jedne reči, analiza ključnih reči, analiza frekvencija reči, pronalaženje i analiza fraza i idioma, kreiranje indeksa i liste reči itd.

[1] "podaci o podacima"
[2] ili morfološko ili gramatičko označavanje
[3] Na primer, za reč *senci*, POS oznaka je *N(*imenica*)*, a lema je *senka*.

Unitex [1][2] je skup programa koji su razvijeni za analizu tekstova prirodnog jezika pomoću lingvističkih resursa. Ovi resursi se sastoje od elektronskih rečnika, gramatika i leksikon-gramatičkih tabela. Programi koji čine Unitex mogu se podeliti na dve celine: grafički interfejs koji je napisan u programskom jeziku Java, i konzolni deo koji je urađen na programskom jeziku C++. Koncept ovog softvera je rođen u LADL (*Laboratoire d'Automatique Documentaire et Linguistique*), pod vođstvom profesora Morisa Grosa.

Elektronski rečnici, gramatike i leksikon-gramatike koje se koriste u Unitex-u napravljeni su u okviru RELEX mreže [5], internacionalne grupe koja se bavi proučavanjem prirodnih jezika.

Elektronski rečnici i gramatike su najvažnije komponente aplikacija koje se bave obradom prirodnih jezika.

Pravljenje rečnika je složeni problem koji se rešava u više koraka. Prvo se pravi rečnik jednočlanih tekstuelnih reči[4]. Ovi rečnici su slični komercijanim rečnicima. Da bi bilo moguće da se porede leme iz rečnika sa rečima koje se nalaze u tekstu koji se obrađuje, svakoj reči su pridodati svi njeni oblici. Pošto je potrebno da ovakav rečnik prepoznaje sve reči iz teksta koje obrađuje, u takve rečnike se dodaju i izvedene reči, brojevi i prebrojavanja i lična imena. Sledeći korak je pravljenje rečnika višečlanih tekstuelnih reči[5]. Na kraju se u rečnike dodaju sematičke oznake[6].

Dakle elektronski rečnici sastoje se od liste jednostavnih i složenih reči jezika zajedno sa njihovim lemama i skupom gramatičkih (semantičkih i flektivnih) kodova. Rečnici su predstavljeni u DELA formalizmu i konstruisali su ih timovi lingvista, za više jezika (srpski, francuski, engleski, grčki, italijanski, španski, nemački, tajlandski, korejski, poljski, norveški, portugalski, itd).

Gramatike koja se ovde koriste su predstavljene na osnovu rekurzivnih mreža prelaska, formalizmom koji je tesno vezan za konačne automate. Brojne studije su pokazale adekvatnosti automata za rešavanje jezičkih problema na različitim nivoima. Gramatike koje su napravljene za Unitex koriste formalizam moćniji od automata. Ove gramatike su predstavljene kao grafovi koje korisnik može lako da pravi i ažurira. Gramatike se koriste i za pronalaženje novih termina, naročito složenih reči iz datih tekstova.

Leksikon-gramatike su matrice koje opisuju osobine nekih reči. Iskustvo je pokazalo da svaka reč ima kvazi-jedinstveno ponašanje, a ove tabele su način da se predstave sintaksičke osobine svakog elementa u leksikonu, otuda i ime leksikon-gramatika za ovu jezičku teoriju. Za Unitex leksikon-gramatike su napravljene za predvidljive elemente kao što su glagoli, imenice i pridevi za svaki od jezika koji su podržani.

U ovom sistemu programa tekstovi koji se obrađuju su pedstavljeni kao jednojezični korpusi sa anotacijima koje se sastoje samo od strogo definisanih metapodataka. Oni su sačuvani u datotekama sa ekstenzijom SNT. Svakoj od tih datoteka se pridružuje po jedan katalog, u daljem radu SNT katalog, u kome se čuvaju dodatni podaci o tekstu, kao na primer:

- Lokalni rečnici, to jest elektronski rečnici koji se satoje samo od reči koje nalaze u datom korpusu. Oni se nalaze u tri datoteke:
    - `dlf` u kojoj je sačuvan rečnik jednočlanih tekstuelnih reči;
    - `dlc` u kojoj je sačuvan rečnik višečlanih tekstuelnih reči;

[4] Primer: *senka, pratiti, ka*
[5] Primer : *sumporna kiselina, zajedno sa, dva miliona*
[6] Primer: *Hum* (čovek), *Col*(boja)

    - `err` u kojoj se nalaze greške[7].
- Podaci o tokenima, koji se nalaze u pet datoteka:
    - `tokens.txt` koja sadrži niz tokena redosledom kojim se pojavljuju u koprusu;
    - `text.cod` binarnsa datoteka koje korpusa u kome su tokeni zamenjeni indeksima kojim se pojavljuju u prethodnoj datoteci;
    - `tok_by_freq.txt` lista tokena uređena po frefenciji;
    - `tok_by_alph.txt` alfabetski urežena lista tokena;
    - `stats.n` datoteka koja sadrži statistike.
- Odstupanja

Takođe su u ovom katalogu sačuvani rezultati pretraga.

Etikete [1] (*tag*) su način na koji se opisuju metapodaci u tekstovima koji se obrađuju u Unitex-u. Etikete se nalaze između vitičastih zagrada i samo etikete se nalaze između vitičastih zagrada. Unitex podržava tri vrste etiketa: etiketa za razdvajanje rečenica, etiketa za zaustavljnje i DELA etikete rečnika. One se koriste za označavanje kraja rečenica, zaustavljanje automata i anotaciju rečnikom.

U Unitex-u možemo vršiti automatske pretrage izraza, koje opisujemo gramatikama. Unitex takođe omogućuje da se pretraga opiše i regularnim izrazom. To je omogućeno programom koji prevodi regularni izraz u grafove koji opisuju gramatike.

Rezultati tih pretraga su sačuvani u datotekama `concord.ind` i `concord.n` koje se nalaze u SNT katalogu korpusa koji se pretražuje. Radi preglednosti Unitex omogućava da se od rezultata pretraga konstruišu konkordance u više formata po želji korisnika.

Radi efikasnosti pretraga Unitex vrši prethodnu obradu teksta. Tako se tekstovi u formatima koje prihvata Unitex konvertuju u Unitex korpuse[8] prethodno opisane u ovom radu.

Prethodna obrada teksta se se sastoji od sledećih šest procesa:

- Normalizacija separatora, u daljem tekstu normalizacija;
- Podela teksta na rečenice;
- Svođenje skračenica na normalni oblik;
- Podela teksta na tokene, u daljem tekstu tokenizacija;
- Primena elekroskih rečnika na tekst;
- Analiza složenih reči;

Kada korisnik učita neobrađeni tekst[9] Uniteks ponudi da se taj tekst prethodno obradi. Od tih procesa normalizacija i tokenizacija su neophodni za dalji rad.

Standarni separatori su blanko karakter, tabulator i karakter za obeležavanje novog reda. U tekstovima može da se desi da se više separatora nađu jedan za drugim ali to nije važno za lingvističku analizu te se normalizacija vrši po sledećim pravilima:

1. Ukoliko niz separatora sadrži karakter za obeležavanje novog reda, taj niz se zameni jednim karakterom za obeležavanje novog reda;
2. Ukoliko niz separatora ne sadrži karakter za obeležavanje novog reda, taj niz se zameni jednim blanko karakterom;

[7] Reči koje nisu prepoznate u okviru sadašnjeg rečnika. Ovo je metod kojim se dodaju nove reči u rečnike.
[8] Datoteke sa ekstenzijom SNT
[9] Tekst koji nije u datoteci sa ekstenzijom SNT.

3. Ukoliko tekst između vitičastih zagrada nije pravilno definisana etiketa, vitičaste zagrade se zamenjuju uglastim;

Rezultat normalizacije je datoteka sa ekstenzijom SNT, a program koji izvršava taj proces pravi i SNT katalog.

Podela teksta na rečenice je važan deo prethodne obrade teksta pošto to pomaže u određivanju jedinica za lingvističku obradu. Unitex obeležava kraj rečenice etiketom predviđenom za tu svrhu. Te etikete se dodaju u tekst tako što se na SNT datoteku primeni odgovarajuća gramatika programom `Fst2Txt` u režimu za spajanje (*MERGE mode*). Grafovi koji opisuje te gramatike napravljeni su u okviru RELEX mreže za svaki jezik koji Unitex podržava.

Skraćenice koje se pojavljuju u korpusima je korisno svesti na nornalni oblik. To se radi tako što se svaka skraćenica u tekstu zameni izrazom koji joj odgovara[10]. Zamene se vrše u tekstu tako što se na SNT datoteku primeni odgovarajuća gramatika programom `Fst2Txt` u režimu za zamenu (*REPLACE mode*). Kao i grafovi za razdvajanje rečenica, grafovi koji opisuje te gramatike su napravljeni u okviru RELEX mreže za svaki jezik koji Unitex podržava.

Pošto su u nekim jezicima reči podeljene na nestardan način, Unitex deli tekst na tokene. Na srpskom jeziku tokeni su: etikete, niz slova iz definisanog alfabeta za taj jezik i jedan karakter van tog alfabeta. Rezulat tokenizacije su datoteke za opisivanje tokena u SNT katalogu korpusa koji je bio tokenizovan. Tokom tokenizacije se karakteri za obelažavanje novog reda zamenjuju blanko karakterom, a informacije o krajevima reda čuvaju u posebnoj datoteci.

Pod primenom rečnika na korpus podrazumevamo konstrukciju lokalnih rečnika koji su ranije bili opisani u radu. Tokom prethodne obrade teksta primenjuje se rečnik koji odgovara jeziku koji je odabran za rad sa tim tekstom. Moguće je primeniti rečnike na korpus van prethodne obrade teksta, i to ne samo standardne rečnike jezika u kome Unitex trenutno radi.

U nekim jezicima moguće je napraviti novu reč spajanjem dveju poznatih reči. Unitex omogućava da se ovakve reči uklone sa liste nepoznatih reči, tokom procesa analize složenih reči.

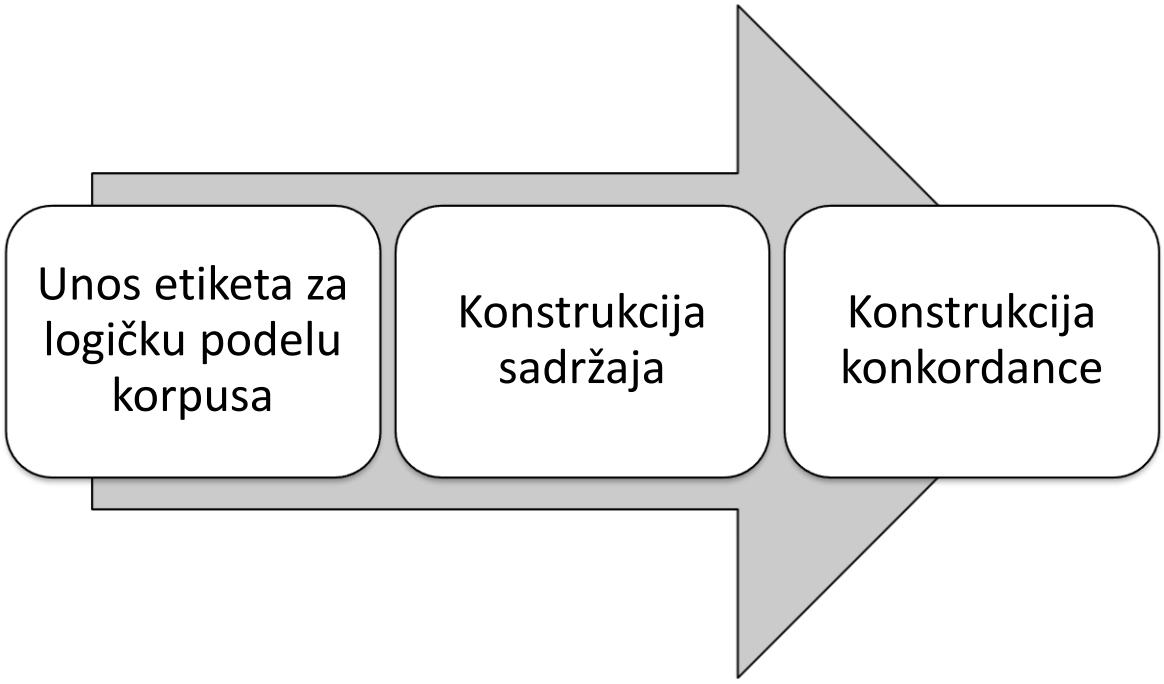
Za pretrage na velikim tekstovima bilo bi veoma korisno da se neki na način prikaže u kom delu teksta se nalaze rezultati tih pretraga. Takođe bi bilo korisno da se omogući analiza više tekstova istovremeno. Ovo Unitex trenutno ne obezbeđuje.

Cilj ovog rada je da se reši taj problem, i to dodavanjem novih funkcionalnosti sistemu Unitex koji treba da obezbede pretragu velikih tekstova ili više tekstova istovremeno, tako da u rezultatu pretrage bude prisutna informacija iz koje logičke celine teksta ili kog teksta je svaka dobijena konkordanca.

To je obezbeđeno podelom korpusa na logičke celine. Da bi se korpus podelio na logičke celine dodaju se novi tipovi metapodataka koji opisiju te logičke celine, i uvodi pojam sadržaja, to jest podataka koji opisuju logičke celine korpusa. Takođe je unapređen i proces konstrukcije konkordanci tako da se u konkordancama prikazuju traženi podaci.

U ovom rešenju rad sa više tekstova je omogućen tako što se ti tekstovi spajaju u jedan veći tekst, pravilno podeljen na logičke celine.

[10] Primer: *tj.* zamenjujemo sa *tako jest, itd.* zamenjujemo sa *i tako dalje*



Primer 1. Proces logičke podele teksta

Na primeru 1 se vidi rešenje problema logičke podele teksta koje je obrađeno u ovom radu. Dodate su etikete za logičku podelu teksta. To su sledeće tri etikete: etiketa za početak naslova prvog nivoa, etiketa za početak naslova drugog nivoa i etiketa za kraj naslova. Pošto je ovo rešenje predviđeno za rad sa velikim korpusima unos tih etiketa u korpuse je donekle automatizovan.

Nakon što su ove etikete dodate u tekstu, one se dalje koriste za konstrukciju „sadržaja“. Njega beležimo u dve datoteke: contains.html koja služi za prikazivanje sadržaja i contains.txt koja se dalje interno koristi tokom konstruisanja konkordanci.

Korisnost logičke podele teksta koji se obrađuje je najvidljivija tokom konstruisanja konkordanci iz rezultata pretraga. Sada je omogućeno da se prilikom konstrukcije konkordanci iz rezultata pretraga takođe koristi i prethodno napravljeni sadržaj. Na taj način se na početku svakog reda konkordance dodaje hiperveza, pošto su konkordance u HTML skriptu, koja pokazuje na naslov kojim počinje logička celina u kojoj se reč iz pretrage nalazi.

Ovo rešenje takođe sadrži i jedan manji program koji automatski obavlja analizu frekvencija reči.

Nove funkcionalnosti Unitex-a

Unitex već deli tekst na rečenice pomoću odgovarujuće etikete[11]. Sličan metod se ovde koristi da bi se korpus podelio na logičke celine. Omogućeno je da programi u Unitex-u prepoznaju etikete za logičku podelu[12].

Da bi se te etikete koristile, konzoli Unitex-a su dodate dve nove naredbe a naredbi za konstrukciju konkordanci je dodata nova opcija za rad sa sadržajem. Nove naredbe su:

- `AddFile` koja spaja tekstove iz više datoteka u jedan korpus;
- `MakeContains` koja koristi etikete za logičku podelu teksta da konstruiše datoteke sadržaja.

Nevezano sa logičkom podelom korpusa takođe je dodata i naredba za prebrojavanje lema, `LemmataCounter`.

Grafički interfejs Uniteksa je prilagođen tako da je moguće iz njega koristiti ove nove i izmenjene naredbe i da se u njemu prikazuju rezultati tih naredbi. Takođe je tu omogućena automatizacija unosa etiketa za logičku podelu teksta na poglavlja pomoću grafa.

Prethodnoj obradi teksta dodata su tri nova procesa:

- Podela teksta na poglavlja;
- Konstrukcija sadržaja;
- Prebrojavanje lema.

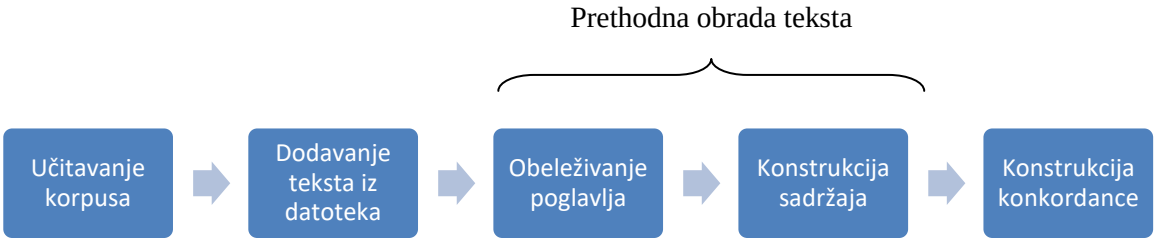
Podela teksta na poglavlja je implementirana na sličan način kao i podela teksta na rečenice, to jest na datoteku korpusa se primenjuje odgovarajući graf u modu za spajanje.

Konstrukcija sadržaja se izvršava pomoću gore opisane naredbe[13], i tokom te faze prethodne obrade teksta se konstruišu datoteke sadržaja koje bivaju sačuvane u SNT katalog teksta koji se obrađuje.

Tokom prebrojavanja lema se konstruišu nizovi uređenih parova koje sadrže leme tokena iz teksta uparene brojem pojavljivanja tokena koji odgovaruju lemama. Oni su sačuvani u dve datoteke u SNT katalogu odgovarajućeg koprusa.

Svi novi procesi su opcionalni, mada da bi se konstruisala konkordanca sa sadržajem potrebno je da prethodno postoje datoteke za opis sadržaja.

[11] {S}
[12] {EOF}, {TB} i {TE}
[13] `MakeContains`



Primer 2. Proces obrade jednog korpusa koji se nalazi u više datoteka.

Na primeru je pokazano kako se obrađuje jedan korpus koji se nalazi u više datoteka. Prvo se učita datoteka sa delom koji treba da bude na početku korpusa. Zatim se u željenom redosledu dodaju jedan po jedan tekstovi iz ostalih datoteka. Sledeća dva koraka se izvršavaju tokom prethodne obrade teksta. Bira se odgovarajući graf za podelu korpusa na poglavlja i zatim se vrši konstrukcija sadžaja. Na tako pripremljenom tekstu se konstruišu konkordance u kojima je svaka reč konkordance povezana sa logičkom celonom u kojoj se ona nalazi.

Ubacivanje Dodatnih Etiketa

U Unitex-u sve etikete se nalaze između vitičastih zagrada. Etikete u Unitex-u se sastoje od etiketa za kraj rečenica, etiketa za zaustavljanje i anotacija rečnikom. Ako tokom normalizacije Unitex ne prepozna etiketu kao jednu od etiketa definisanih u programu, vitičaste zagrade se zamenjuju uglastim [1].

Kao što je navedeno u uvodu za uvođenje logičke podele teksta, potrebno je dodati tri nove etikete: etiketu za početak naslova prvog nivoa {EOF}, etiketu za početak naslova drugog nivoa {TB} i etiketu za kraj naslova {TE}.

Pod dodavanjem etiketa podrazumevamo modifkovanje određenih programa iz Unitex-a, naročito programe za normalizaciju, tokenizaciju i pretragu odnosno primenu gramatike, tako da ti programi prepoznaju nove tri etikete kao legitimne.

Da bi se dodale ove etikete programu za normalizaciju promenjena je datoteka `NormalizeAsRoutine.cpp`, preciznije funkcija `normalize`, koja vrši normalizaciju teksta, u njoj. Kao što se vidi sa primera gde je zasenčeni deo nov kod, u mestu za indetifikaciju oznaka samo je omogućena indentifikacija novih oznaka. Ovako je omogućeno da se normalizuje tekst sa novim oznakama.

```
      if (!u_strcmp(tmp, "{S}") || !u_strcmp(tmp, "{STOP}")
         ||!u_strcmp(tmp, "{EOF}") || !u_strcmp(tmp, "{TB}")
                                   ||!u_strcmp(tmp, "{TE}")
                                   || check_tag_token(tmp))
```

Primer 3. Kod iz datoteke NormalizeAsRoutine.cpp

Kako bi se izbeglo javljanje greške tokom tokenizacije teksta sa unetim novim oznakama, modifikovana je funkcija `tokenization` u datoteci `Tokenize.cpp` na sličan način.

U daljoj obradi teksta Unitex pretpostavlja ako etiketa nije etiketa za podelu teksta (etiketa za razdvajanje rečenica {S} ili etiketa za zaustavljnje {STOP}) da je ta etiketa komentar rečnika, i pokušava da je parsira kao takvu. Ako je to jedna od novih etiketa to dovodi do zaustavljanja programa i prosleđivanja poruke o grešci. Da bi se taj problem otklonio, u kodu koji prepoznaje etikete {S} i {STOP} dodat je kod za prepoznavanje etiketa {EOF}, {TB} i {TE}. Taj kod se nalazi u sledeće četiri datoteke `Text_Tokens.cpp`, `LocatePettern.cpp`, `MorphologicalLocate.cpp` i `MorphologicalFilters.cpp` i veoma je sličan kodu sa primera 3.

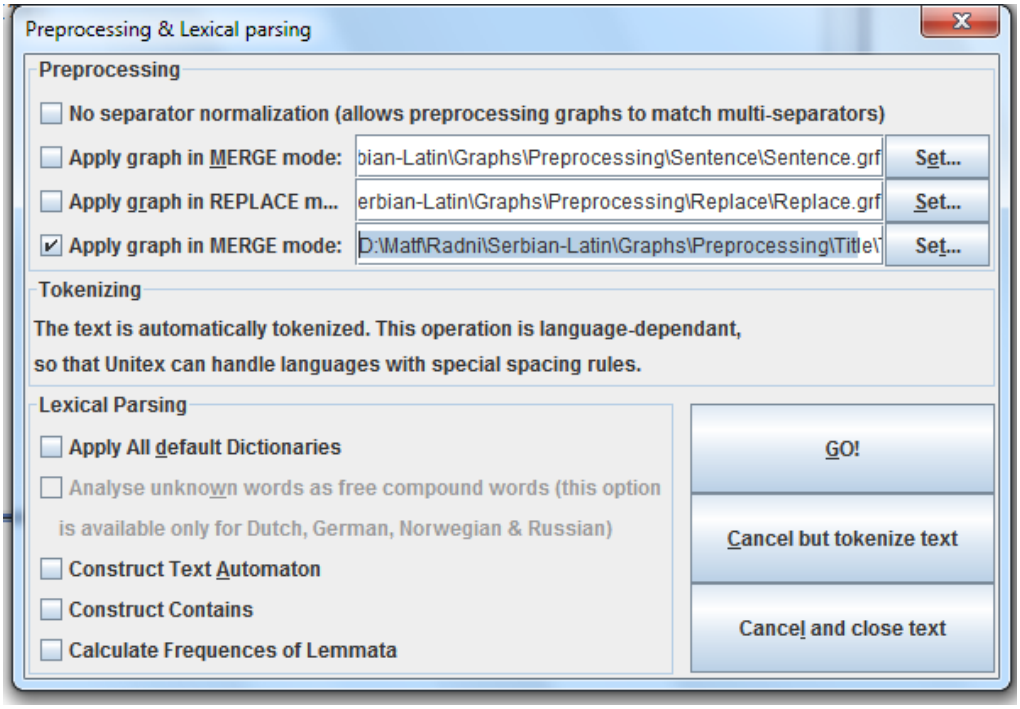
U datoteci `LocatePattern.cpp`, koja sadrži kod za pretrage, promenjene su funkcije `get_control_byte`, koja vraća kontrolni bit koji opisuje karakteristike datog tokena i `check_patterns_for_tag_tokens`, koja prepoznaje DELA oznake rečnika i proverava da li odgovaraju postavljenim uslovima pretrage.

U datoteci `Text_Tokens.cpp`, koja opisuje metode za rad sa tekstualnim tokenima, na naveden način izmenjena je procedura `extract_semantic_codes_from_tokens`, koja prepoznaje koji su od datih tokena DELA etikete[14] i iz onih koji su DELA etikete izdvaja semantičke kodove.

U datoteci `MorphologicalFilters.cpp`, koja opisuje metode za kreiranje i rad sa morfolškim filterima, je promenjena funkcija `new_FilterMatchIndex`, koja obezbeđuje mesto u memoriji i pravi strukturu koje nam za svaki token govori sa kojim od datih filtera se on slaže.

U datoteci `MorphologicalLocate.cpp` promenjena je funkcija `get_jamo_longest_prefix`, koja vrši poređenje tokena oznaka sa korejskim džamo (*Jamo*) slovima.

Od promena na grafičkom interfejsu ovde će biti opisana automatizacija podele teksta na poglavlja naslovima drugog reda, to jest ubacivanje parova oznaka {TB} i {TE} na odgovarajuća mesta. O automatizaciji naslova prvog reda će biti više reči u odeljku koji opisuje rad sa više datoteka.



Primer 4. Dijalog za prethodnu obradu teksta u kome se označeno polje za potvrdu za ubacivanje etiketa za označavanje naslova

Podela teksta na poglavlja je urađena na sličan način na koji Unitex obrađuje razdvanje rečenica etiketom {S}. To znači da se na tekst primenjuje gramatika opisana sa grafom koji opisuje naslove poglavlja sa ubacivanjem oznake {TB} na početku i oznake {TE} na kraju naslova, u režimu za spajanje (*MERGE mode*). Konstrukcija ovog grafa je problematična, pošto za razliku od grafa koji opisuje rečenice on ne zavisi samo od jezika već i od načina na koji su naslovi poglavlja predstavljeni[15], što varira od teksta do teksta. To znači da je praktično potrebno napraviti poseban graf za svaki tip naslova na svakom jeziku, ili praktičnije da korisnik napravi nov graf kada god naiđe na tekst čije naslove poglavlja nije moguće prepoznati grafovima koje već ima. U ovom radu dat je samo jedan primer grafa pod srpkim jezikom (latinicom).

[14] Radi izbegavanja višeznačnosti Unitex omogućava da se umesto nekih tokena unesu etikete koje predstavljaju redove iz rečnika u DELA formatu.
[15] Na primer : I.; Glava prva; 1. Uvod itd...

Primena tih grafova je omogućena, kao što se vidi sa primera 4, tako što je u dijalogu za prethodnu obradu teksta dodat panel, koji sadrži polje za potvrdu, polje za unos teksta i dugme koje poziva dijalog za izbor datoteka, koji izgleda isto kao i panel koji služi za podelu rečenica, ispod panela za zamenu skraćenica. Na primeru je polje za potvrdu dodatog panela označeno. Graf za obradu naslova je specifičan i po jeziku i po vrsti teksta koji obrađuje. U primeru je učitan graf koji obeležava naslove poglavlja romana koji počinju sa tekstom "Glava" zatim redni broj do devedesed devet, tekst naslova i na kraju kraj reda.

U tu svrhu je promenjena klasa `PreprocessDialog` iz paketa `fr.umlv.unitex.frames`. Njoj su dodate promenljive: `titleCheck`, koja predstavlja polje za potvrdu, i `titleName`, koja predstavlja polje za unos teksta, i metoda `titleGraph`, koja kreira konzolnu naredbu na osnovu vrednosti unesene u polje za unos teksta. Takođe su promenjene metode `preprocess` i `constructProcessingPanel`.

Rad sa više datoteka

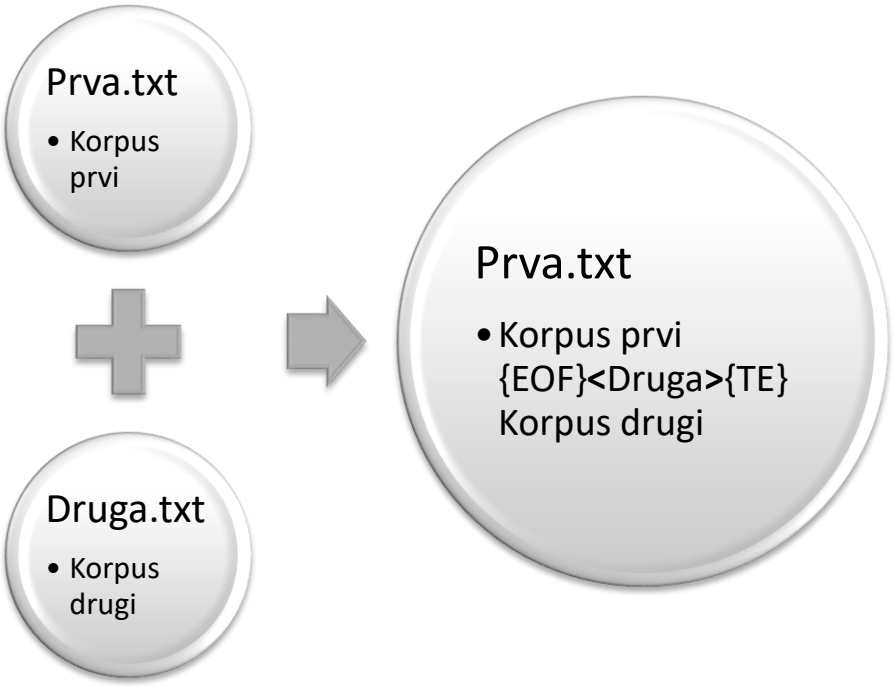
Naslovi prvog nivoa su predviđeni za veće logičke celine, koje će se nalaziti u posebnim datotekama. U ovom poglavlju je opisan način na koji se od više datoteka može da napravi jedan korpus, koji je podeljen na logičke celine naslovima prvog reda prema sadržaju datoteka od kojih je napravljen.

U konzoli Unitex-a dodata je nova naredba:

AddFile [Opcije] datoteka datoteka1 [datoteka2...]

Ova naredba poziva program koji spaja niz datoteka u jednu datoteku. Ove datoteke mogu biti neformatirani tekst (datoteke sa ekstenzijom TXT), tekst već obrađen u Unitex-u (datoteka sa ekstenzijom SNT) i hipertekstovi (datoteke sa ekstenzijama XML, HTM i HTML). Potrebno je da korisnik navede bar dve datoteke. Podržan je unos do pedeset datoteka.

Ovaj program prvo prevodi prvu datoteku u neformatirani tekst, ako već nije u tom formatu, zatim pravi kopiju te datoteke sa imenom `old_<ime prve datoteke>` (ako već postoji datoteka sa tim imenom onda se koristi `old01_<ime prve datoteke>`, `old02_<ime prve datoteke>`...) Svaka sledeća datoteku iz niza se prvo prevodi u neformatirani tekst, ako već nije u tom formatu, zatim se na kraj prve datoteke dodaje linija teksta kojom označavamo naslov prvog reda ({EOF}<ime datoteke iz niza koju obrađujemo>{TE}) i na kraju dodajemo sadržaj te datoteke prvoj datoteci. Na primeru 5 je pokazan taj proces na datotekama `Prva.txt` i `Druga.txt`.



Primer 5. Spajanje dve datoteke

Opcije koje se mogu koristiti pri pozivanju ovog programa su:

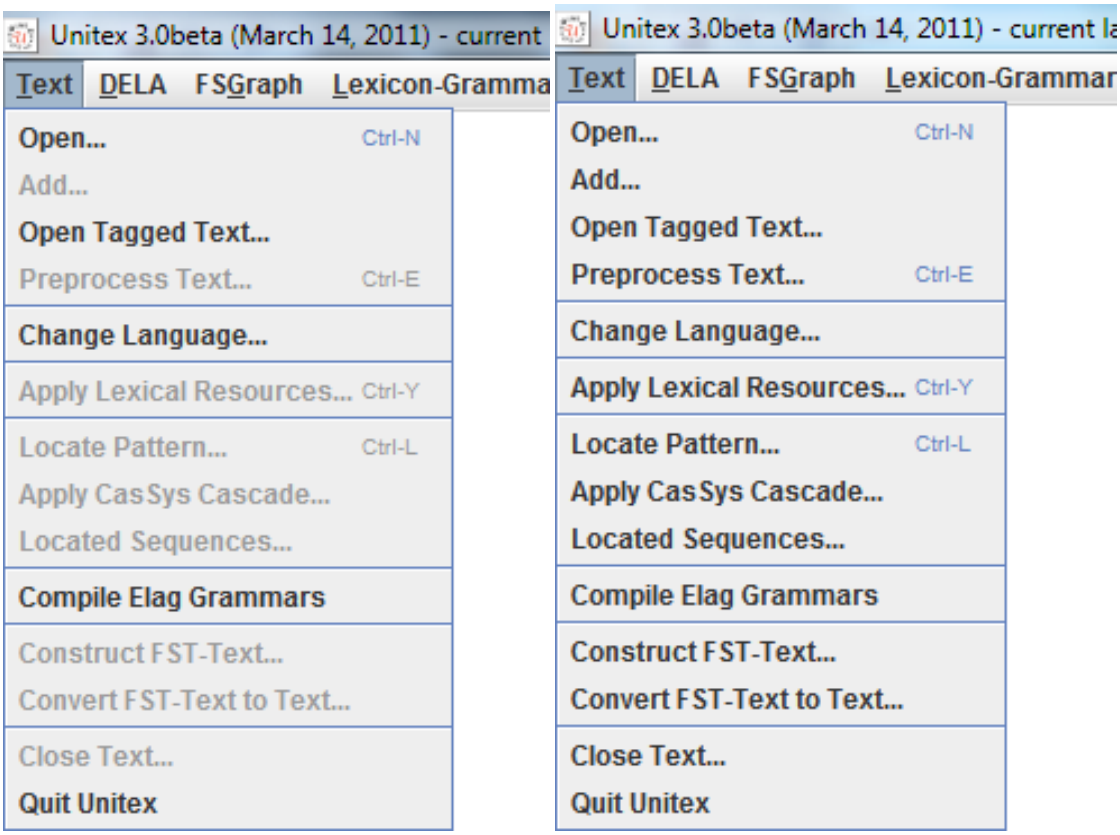
- `-h/--help`: koja poziva upustvo za korišćenje ovog programa;
- `-o <ime datoteke>/-old_text=<ime datoteke>`: ova opcija omogućava da korisnik zada ime kopije prve datoteke pre nego što se toj datoteci doda novi sadržaj.

U datoteci `UnitexTool.cpp u` nizu `utility_array` su dodate informacije koje omogućavaju da naredba `AddFile` pokrene odgovarajući program. Takođe je u toj datoteci dodata predprocersorska direktiva da se uključi datoteka `AddFile.h`.

Program koji omogućava rad sa više datoteka se nalazi u datotekama `AddFile.h i AddFile.cpp.` Komponente ovog programa su:

- `main_AddFile`: glavna funkcija koja se poziva naredbom `AddFile`;
- `optstring_AddFile` i `lopts_AddFile`: promenljiva i niz kojim opisujemo moguće opcije. One takođe služe za parsiranje opcija koje se navedene uz naredbu;
- `usage_AddFile`: promenljiva koja sadrži tekst upustva;
- `MAX_ADDED_FILES`: konstanta koja opisuje najveći broj datoteka sa kojim ovaj program radi. Njena vrednost je pedeset;
- `extension_is`: funkcija koja određuje tip datoteke prema ekstenziji. Podržava ekstenzije txt, snt, xml, htm i html;
- `snt2txt`: procedura koja za datu datoteku obrađenu u Unitex-u nalazi originalni tekst i ako taj tekst više ne postoji pravi se kopija obrađene datoteke sa ekstenzijom txt;
- `xml2txt`: funkcija koja prevodi hipertekst u neformatirani tekst. Ona to radi na način već opisan u datotekama `Xml.h` i `Xml.cpp`.

U datoteci `AddFile.h` su deklarisane sledeće komponente `main_AddFile`, `usage_AddFile`, `optstring_AddFile` i `lopts_AddFile` i definisana konstanta `MAX_ADDED_FILES`.



Primer 6. Padajući meni `Text sa učitanim korpusom (desno) i bez korpusa (levo)`

Kao što se vidi sa primera 6, u grafičkom interfejsu, u padajućem meniju `Text` dodat je element `Add`, koji poziva naredbu za spajanje datoteka, ispod elementa `Open`, koji služi za učitavanje korpusa. Element `Add` je aktivan samo kada je učitan korpus.

Kada korisnik pozove taj element, on poziva dijalog za izbor datoteka korpusa, što je u stvari podešen dijalog za izbor datoteka[16], i zatim kada korisnik izabere korpus koji se dodaje aktivnom korpusu konzoli se šalje sledeća naredba:

`AddFile <ime datoteke aktivnog korpusa> <ime datoteke korpusa koji je korisnik izabrao dijalogom>.`

Pošto konzolni program napravi novi korpus, on se učitava kao korpus za obrađivanje na standardan način za učitavanje neformatiranog teksta.

Na primeru 7 vidimo kod Java klase `AddFileCommand`, koja pravi naredbu za spajanje datoteka, iz paketa `fr.umlv.unitex.process.commands.` Ova klasa ima samo metodu `text` koja komandi dodaje novu datoteku za spajanje.

```
public class AddFileCommand extends CommandBuilder {

	public AddFileCommand() {
		super("AddFile");
	}

	public AddFileCommand text(File s) {
		protectElement(s.getAbsolutePath());
		return this;
	}

}
```

Primer 7. Java klasa za pravljenje komande `AddFile`

Objekt ove klase nalazimo u metodi `addText` klase `UnitexFrame` iz paketa `fr.umlv.unitex.frames`, koji poziva dialog za izbor datoteke korpusa, koristi objekt klase `AddFileCommand` da napravi ranije navedenu naredbu za spajenje korpusa, izvrši je i na kraju učita dobijeni korpus za obradu.

Ovaj metod poziva objekat `addText` iz iste klase.

U metodi `buildTextMenu` koja gradi padajući meni `Text` objekat `addText se` dodaje u taj meni.

[16] Definisani su filteri za izbor datoteka po ekstenziji

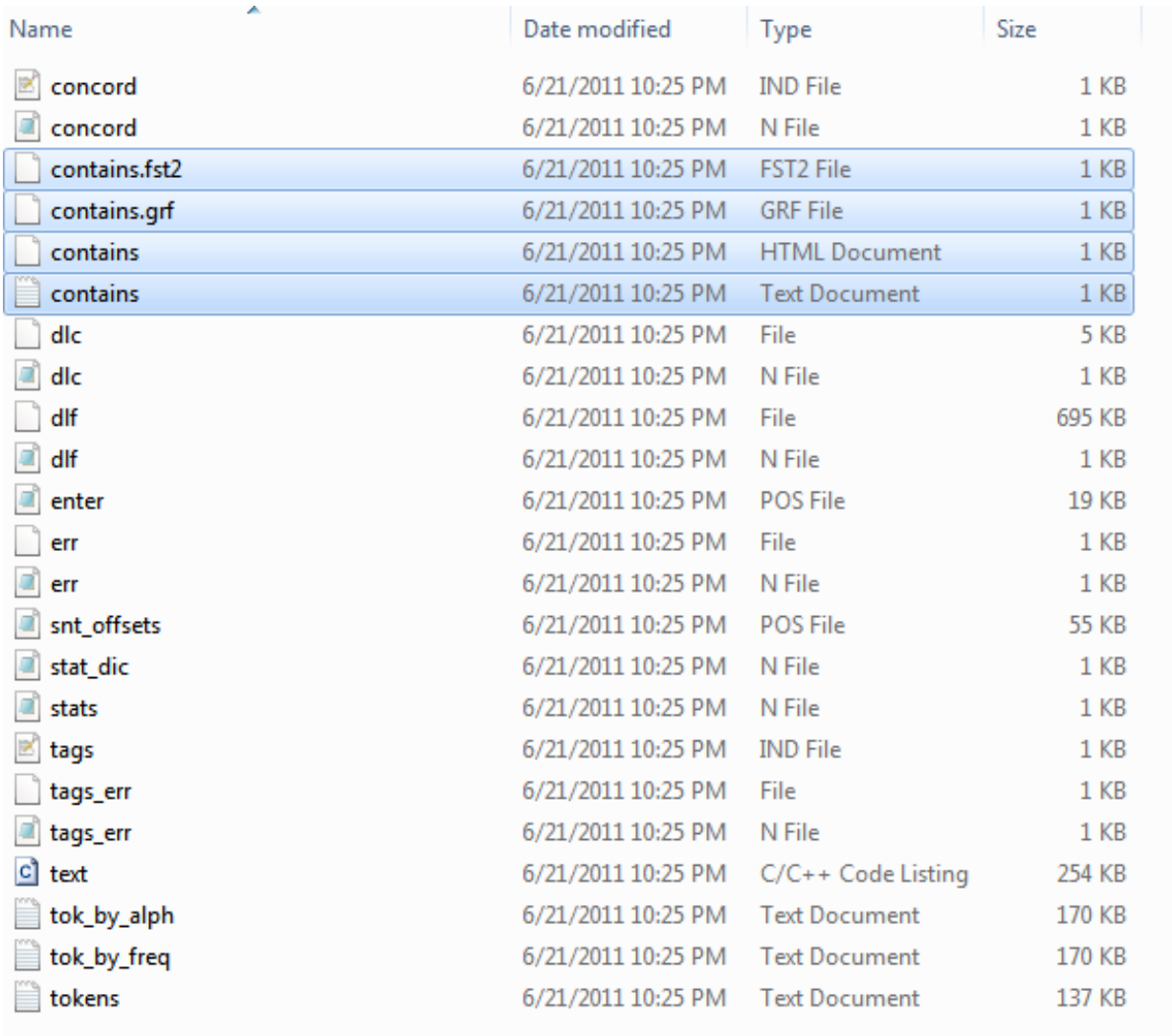
Pravljenje Sadržaja

U ovom poglavlju je opisan program koji pravi datoteke sadržaja i promene na grafičkom interfejsu koje omogućavaju da sa grafičkog interfejsa pokrenemo[17] taj program.

Program pozivamo naredbom:

MakeContains [Opcije] <datoteka korpusa>

Ovaj program u korpusu nalazi oznake za logičku podelu i na osnovu njih pravi datoteke koje opisuju sadržaj u SNT katalogu koji Unitex pravi uz svaki korpus tokom procesa normalizacije. Na primeru je prikazan jedan takav katalog u kome su zasenčene datoteke koje ovaj program pravi.



Name	Date modified	Type	Size
concord	6/21/2011 10:25 PM	IND File	1 KB
concord	6/21/2011 10:25 PM	N File	1 KB
contains.fst2	6/21/2011 10:25 PM	FST2 File	1 KB
contains.grf	6/21/2011 10:25 PM	GRF File	1 KB
contains	6/21/2011 10:25 PM	HTML Document	1 KB
contains	6/21/2011 10:25 PM	Text Document	1 KB
dlc	6/21/2011 10:25 PM	File	5 KB
dlc	6/21/2011 10:25 PM	N File	1 KB
dlf	6/21/2011 10:25 PM	File	695 KB
dlf	6/21/2011 10:25 PM	N File	1 KB
enter	6/21/2011 10:25 PM	POS File	19 KB
err	6/21/2011 10:25 PM	File	1 KB
err	6/21/2011 10:25 PM	N File	1 KB
snt_offsets	6/21/2011 10:25 PM	POS File	55 KB
stat_dic	6/21/2011 10:25 PM	N File	1 KB
stats	6/21/2011 10:25 PM	N File	1 KB
tags	6/21/2011 10:25 PM	IND File	1 KB
tags_err	6/21/2011 10:25 PM	File	1 KB
tags_err	6/21/2011 10:25 PM	N File	1 KB
text	6/21/2011 10:25 PM	C/C++ Code Listing	254 KB
tok_by_alph	6/21/2011 10:25 PM	Text Document	170 KB
tok_by_freq	6/21/2011 10:25 PM	Text Document	170 KB
tokens	6/21/2011 10:25 PM	Text Document	137 KB

Primer 8. Jedan SNT katalog sa sadržajem

Opcije koje se mogu koristiti pri pozivanju ovog programa su:

- `-h/--help`: koja poziva uputsvo za korišćenje ovog programa;

[17] Tokom predodne obrade teksta

- `-f FONT`: postavlja stil karaktera u hipertekstu;
- `-s VELIČINA`: postavlja veličinu karaktera u hipertekstu.

Datoteke koje opisuju sadržaj su `contains.txt` i `contains.html` o kojima će biti više rečeno u sledećem odeljku.

U korpusu etikete za logičku podelu su predviđene da se nalaze u uređenim parovima, prvo etiketa za početak naslova[18] zatim etiketa za kraj naslova[19], i tekst između njih se smatra naslovom logičke celine. Ukoliko u korpusu imamo dve etikete za početak naslova program pretpostavlja da nedostaje etiketa za kraj naslova između njih, šalje poruku da nedostaje etiketa za kraj naslova i pravi jednu logičku celinu između njih sa naslovom od dva tokena posle prve etikete. Ukoliko u korpusu program nađe etiketu za kraj naslova a da pre toga nije našao neuparenu etiketu za početak naslova on šalje poruku da je nađena neuparena etiketa i onda tu etiketu ignoriše. Dužina naslova logičkih celina je ograničena. Ako je tekst između etiketa veći od dozvoljenog, on biva skraćen na dozvoljenu dužinu.

U datoteci `UnitexTool.cpp u` nizu `utility_array` su dodate informacije koje omogućavaju da naredba `MakeContains` pokrene odgovarajući program. Takođe u toj datoteci je dodata predprocersorska direktiva da se uključi datoteka `MakeContains.h`.

Program koji omogućava pravljenje sadržaja se nalazi u datotekama `MakeContains.h i MakeContains.cpp.` Komponente ovog programa su:

- `main_MakeContains`: glavna funkcija koje se poziva naredbom `MakeContains`. O toj funkciji ćemo više reči u daljem tekstu;
- `optstring_MakeContains` i `lopts_MakeContains`: promenljiva i niz kojim opisujemo moguće opcije. One takođe služe za parsiranje opcija koje su navedene uz naredbu;
- `usage_MakeContains`: promenljiva koja sadrži tekst uputsva;
- `tag_is`: Funkcija čiji je argument token i vraća indikator da li je taj token oznaka za logičku podelu i ako jeste koja je to oznaka;
- `extract_title`: funkcija koja ako su data pozicije početnog i krajnjeg tokena naslova vraća taj naslov, i ako taj naslov prelazi dozvoljenu dužinu skraćuje ga.

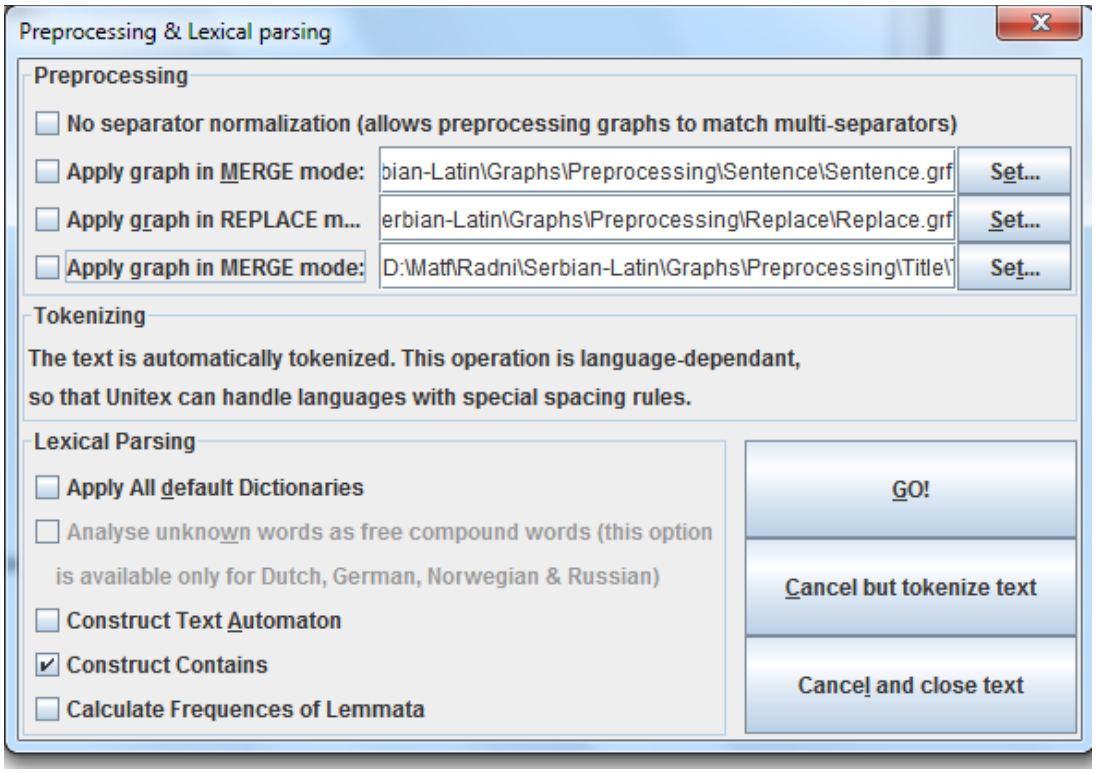
Glavna funkcija `main_MakeContains` regularni izraz `"{EOF}+{TB}+{TE}"` prevodi prvo u graf `contains.grf` a onda graf priprema za pretragu kao `contains.fst2`. Zatim ta funkcija poziva funkciju `locate_pattern` iz datoteke `LocatePattern.cpp` koja rezultate pretrage upisuje u datoteku `concord.ind`[20].

U sledećem koraku izvršavanja glavne funkcije rezultati pretrage se iz te datoteke učitavaju u listu specijalno konstruisanu u tu svrhu. Ta lista se naziva `match_list`. Ona i funkcije za rad sa njom su opisane u datotekama `LoacateMatches.h i LoacateMatches.cpp.` Funkcija inicijalizuje strukture za rad sa sadržajem, a te strukture i funkcije za rad sa njima će biti opisane u sledećem poglavlju, i zatim postavlja ime datoteke korpusa koja se obrađuje kao prvi naslov prvog reda. Zatim

[18] {EOF} ili {TB}
[19] {TE}
[20] Napomena: prilikom pravljenja sadržaja gube se rezultati prethodne pretrage ako ona postoji. To je značajno pošto Unitex koriti preatrage da napravi indeks reči za pravljenje konkordanci.

ulazi u petlju koja ispituje svaku pronađenu oznaku funkcijom `tag_is`. Ako su to oznake za početak, prvo proverava da li je proces trenutno u naslovu i ako jeste obrađuje grešku, zatim funkcija sačuva početak naslova i pređe u mod „u naslovu“. Ako je to oznaka za kraj naslova, prvo proverava da li je proces trenutno u u naslovu i ako jeste poziva funkciju `extract_title` i zatim koristi funkcije za rad sa sadržajem da ubaci novu stavku sadržaja sa dobijenim podacima. U suprotnom slučaju javlja grešku. Van te petlje funkcija poziva metode za snimanje sadržaja u datoteke `contains.txt` i `contains.html`.

Od promena na grafičkom interfejsu ovde će biti opisano pozivanje ove naredbe iz grafičkog interfejsa. Pošto se pravljenje sadržaja vrši uobičajno jedanput, i pre bilo koje pretrage korpusa, naredba za pravljenje sadržaja se poziva tokom prethodne obrade korpusa.



Primer 9. Dijalog za prethodnu obradu teksta sa izabranim poljem za pravljenje sadržaja

U tu svrhu u dijalog za prethodnu obradu teksta, u panelu za leksičko parsinje, kao što se vidi na primeru, dodato je polje za potvrđivanje nazvano `Contruct Contains`. Ako je ono aktivirano kada se klikne na dugme `GO!` konzoli se šalje sledeća komanda:

`MakeContains -f<stil karaktera konkordance> -s<veličina karaktera konkordance> <ime datoteke aktivnog korpusa>`

To je omogućeno promenama u klasi `PreprocessDialog` iz paketa `fr.umlv.unitex.frames`. Njoj je dodat objekt `containsCheck`, koji predstavlja polje za potvrdu. Takođe su promenjene metode `preprocess`, koja izdaje naredbe kada korisnik klikne na dugme `GO!`, i `construcLexicalParsingPanel`, koja konstruiše panel za leksičko parsiranje. U metodi `construcLexicalParsingPanel` je panelu za leksičko parsiranje dodato polje za izbor da li pravi sadržaj, a metodi `preprocess` dodato je da se pomoću objekta klase `MakeContainsCommand` doda

naredba za pravljenje sadržaja naredbama koje izvršavaju, ako je aktivirano polje za izbor koje odgovara objektu `containsCheck.`

Klasa `MakeContainsCommand`, koja pravi naredbu za pravljenje sadržaja, nalazi se u paketu `fr.umlv.unitex.process.commands.` Ova klasa ima tri metode koje komandi dodaje novu datoteku za spajanje:

- metodu `text` koja u naredbi postavlja ime datoteke korpusa čiji se sadržaj pravi;
- metodu `font` koja u naredbi postavlja stil karaktera u hipertekst datoteci sadržaja;
- metodu `fontSize` koja u naredbi postavlja veličinu karaktera u hipertekst datoteci sadržaja.

U konstruktoru se pozivaju metode `font` i `fontSize` koje usaglašavaju stil i veličinu karaktera sadržaja sa konkordancom.

Sadržaj

U ovom poglavlju je opisano kako se sadržaj interno opisuje u okviru sistema programa Unitex, i kako se on prikazuje u grafičkom interfejsu.

Kao što je rečeno u prethodnom poglavlju kada Unitex-u naredimo da napravi sadržaj on pravi datoteke `contains.txt` i `contains.html`. U datoteci `contains.txt` se nalaze sve informacije neophodne da se napravi interna reprezentacija sadržaja. Ova datoteka može se koristiti prilikom pravljenja konkordanci. Datoteka `contains.html` služi za prikazivanje sadržaja u grafičkom interfejsu.

```
Test primer
{TB}Prvi Naslov{TE}
Jedna rečenica. Još jedna rečenica.
{TB}Drugi Naslov{TE}
Ponovo jedna rečenica. Još jedna rečenica. {TB}Treći Naslov{TE}
{EOF}Druga_Datoteka{TE}
{TB}Naslov u drugoj datoteci{TE}
Rečenica u drugoj datoteci.
{TB}Drugi Naslov U Drugoj Datoteci{TE}
```

Primer 10. Jedan korpus sačuvan u datoteci Primer.txt

1.
- Test primer

1.1
- {TB}Prvi Naslov{TE}
- Jedna rečenica. Još jedna rečenica.

1.2
- {TB}Drugi Naslov{TE}
- Ponovo jedna rečenica. Još jedna rečenica.

1.3
- {TB}Treći Naslov{TE}

2.
- {EOF}Druga_Datoteka{TE}

2.1
- {TB}Naslov u drugoj datoteci{TE}
- Rečenica u drugoj datoteci.

2.2
- {TB}Drugi Naslov U Drugoj Datoteci{TE}

Primer 11. Korpus iz primera 10 podeljen na logičke celine naslovima

Na primeru 10 se vidi korpus koji se koristi za izgradnju i korišćenje sadržaja. Na sledećem primeru je pokazano kako naslovi dele taj korpus u logičke celine. Kao što se vidi, logičke celine su ograničene naslovima. Dakle, jedna logička celina drugog reda je tekst od početka datoteke do prvog naslova (ma kog reda), sledeće logičke celine sadrže tekst naslova i tekst do početka sledećeg naslova, i na kraju poslednja logička celina sadrži tekst poslednjeg naslova i sav tekst do kraja datoteke.

Strukture koje opisuju sadržaj i metode za rad sa njim opisane su u datotekama `Contains.cpp` i `Contains.h`. One su:

- `Contains`: struktura koja opisuje sadržaj. To je lista stavki sadržaja, gde se u svakoj stavki nalazi naslov, serijski broj, pozicija u tokenima, pozicija naslova, i indiakator nivoa podele;
- `Contains_options`: struktura u kojoj se nalaze opcije za rad sa datotekema sadržaja, uključujući i neophodne podatke za generaciju hiperteksta(stil i veličinu karaktera u njemu kao i naslov);
- `newContains_options`: inicijalizuje promenljivu tipa `Contains_options` i postavlja stil karartera hiperteksta na `Ariel` i veličinu na deset;
- `freeContains_options`: oslobađa memoruju od promenljive tipa `Contains_options;`
- `addToContains`: dodaje element na kraj liste tipa `Contains`, i ako ta lista ne postoji inicijalizuje je. Pretpostavlja se da se elementi unose onim redom kojim se pojavljuju u tekstu;
- `findInContains`: nalazi element liste u kome se nalazi dati token ako je data njegova pozicija u tokenima. Svaki element liste, kao što je već rečeno, sadrži poziciju u tokenima gde ta logička celina počinje. To znači da ova funkcija nalazi takav element liste da data pozicija bude veća od pozicije u elementu liste a manja od pozicije sledećeg elementa ili ako takav element ne postoji poslednji element liste;
- `freeContains`: rekurzivno briše listu koja opisuje sadržaj i oslobađa memoriju;
- `createContainsHTM`: pravi hipertekst datoteku koja služi za prikazivanje sadržaja, prema podacima iz promenljive tipa `Contains_options`;
- `saveContains`: snima listu koja opisuje sadržaj u datoteku čije je ime dato u promenljivoj tipa `Contains_options`;
- `loadContains`: učitava listu koja opisuje sadržaj iz datoteke čije je ime dato u promenljivoj tipa `Contains_options`.

1.	Primer	0	1	1 1 1
1.1.	Prvi Naslov	4	0	17 28 1
1.2.	Drugi Naslov	23	0	78 90 2
1.3.	Treći Naslov	44	0	146 158 3
2.	Druga_Datoteka	50	1	169 183 3
2.1.	Naslov u drugoj datoteci	56	0	193 217 3
2.2.	Drugi Naslov U Drugoj Datoteci	75	0	256 286 3

Primer 12. Sadržaj korpusa sačuvanog u datoteci Primer.txt

Na primeru 12 se vidi interna reprezentacija sadržaja korpusa iz primera 10. To je tekst iz datoteke napravljenje metodom `saveContains`. Ime te datoteke je određeno u opcijama za rad sa datotekama sadržaja.

Svaki red u ovoj datoteci pretstavlja jedan element liste koja opisuje sadržaj. Ovi podaci su sačuvani u sledećem formatu:

`<redni broj> <separator> <naslov> <separator> <pozicija u tokenima> <separator> <indikator nivoa podele> <separator> <pozicija naslova>`

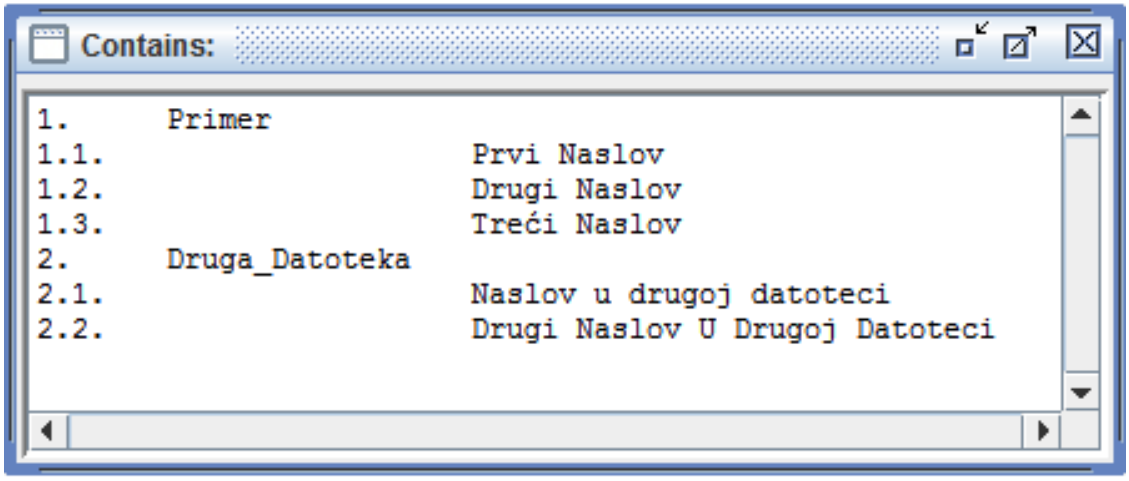
Redni broj naslova drugog nivoa sadrži prvo redni broj naslova prvog nivoa u čijoj se logičkoj celini taj naslov nalazi, a zatim broj koji označava koji je po redu u toj logičkoj celini taj naslov. Pozicija u tokenima određuje pozicuju u tokenima na kojoj počinje logička celina datog naslova. Ta logička celina se završava početkom sledeće logičke celine ili kraja datoteke ako je ta logička celina poslednji element sadržaja. Indikator nivoa podele ima vrednost 1 ako određuje naslov prvog nivoa i 0 ako određuje naslov drugog nivoa. Pozicija naslova služi da bi grafički interfejs selektovao naslov. Ona se sastoji od pozicije početka naslova u karakterima, pozicije kraja naslova u karakterima i broja rečenice u kojoj se naslov nalazi razdvojene blanko karakterom.

Na primeru 13. se vidi izvorni kod hiperteksta koji prikazuje sadržaj korpusa iz primera 10.

```
<html lang=en>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Contains of Primer</title>
</head>
<body>
<table border="0" cellpadding="0" width="100%" style="font-family: 'Courier new'; font-size: 12">
<tr><td nowrap><a href="1 1 1">1.</a></td><td nowrap>Primer</td><td nowrap></td></tr>
<tr><td nowrap><a href="17 28 1">1.1.</a></td><td nowrap></td><td nowrap>Prvi Naslov</td></tr>
<tr><td nowrap><a href="78 90 2">1.2.</a></td><td nowrap></td><td nowrap>Drugi Naslov</td></tr>
<tr><td nowrap><a href="146 158 3">1.3.</a></td><td nowrap></td><td nowrap>Treći
Naslov</td></tr>
<tr><td nowrap><a href="169 183 3">2.</a></td><td nowrap>Druga_Datoteka</td><td
nowrap></td></tr>
<tr><td nowrap><a href="193 217 3">2.1.</a></td><td nowrap></td><td nowrap>Naslov u drugoj
datoteci</td></tr>
<tr><td nowrap><a href="256 286 3">2.2.</a></td><td nowrap></td><td nowrap>Drugi Naslov U
Drugoj Datoteci</td></tr>
</table></body>
</html>
```

Primer 13. Izvorni kod hiperteksta koji prikazuje sadržaj korpus primera

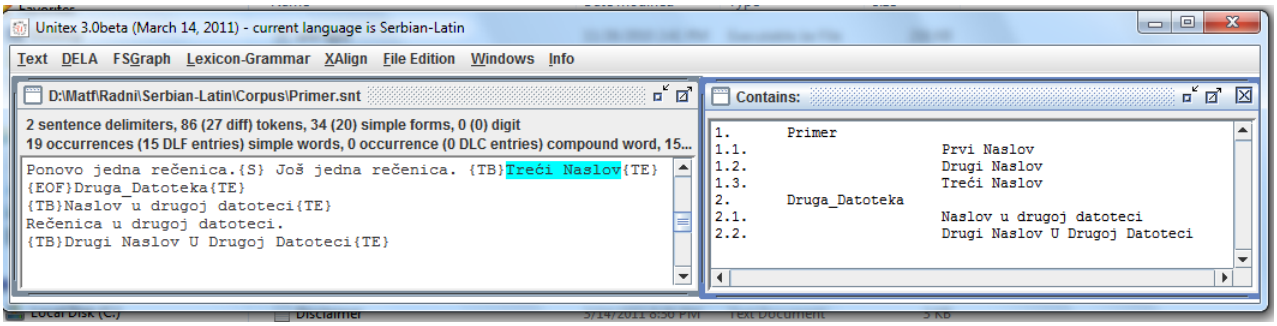
Sadžaj je prikazan kao tabela sa tri kolone. U prvoj koloni se nalazi redni broj kao hiperlink koji pokazuje na sam naslov preko pozicije naslova. Kod naslova prvog reda naslov se nalazi u drugoj koloni dok je treća kolona prazna. Kod naslova dugog reda naslov se nalazi u trećoj koloni dok je druga kolona prazna. Naslov, stil i veličina karaktera su podešeni prema unetim opcijama. Ova datoteka dalje koristi prikaz sadržaja kao što se vidi na primeru ispod.



Primer 14. Sadržaj prikazan u grafičkom interfejsu

Ovaj prozor se pojavljuje posle učitavanja korpusa i nestaje kada se on zatvori, bez obzira da li ta datoteka sadržaja postoji.

Objekat koji opisuje ovaj prozor je klase `ContainsFrame.` Panel koji prikazuje sadržaj je opisan objektom klase `JeditorPane,` koja podržava rad sa hiperteksom[21]. Na primeru 15 vidimo šta će se desiti kada se klikne na jednu od hiperveza u okviru sadržaja.



Primer 15. Rezultat izbora elemenata sadržaja

Objekat ove klase se kreira pomoću objekta klase `ContainsFrameFactory`, koji je pod imenom `containsFrameFactory` polje klase `InternalFrameManager`, koja služi za rad sa prozorima u grafičkom interfejsu Unitex-a. Takođe je promenjena klasa `UnitexFrame`, tako da se pravi prozor za prikazivanje sadržaja kada god je učitan korpus.

Sve gore navedene Java klase se nalaze u paketu `fr.umlv.unitex.frames.`

[21] Za razliku od panela kojim se prikazuju konkordance

Korišćenje sadržaja u konkordancama

Konkordance u Unitex-u služe da se prikažu rezultati pretraga.

U ovom poglavlju opisuje se kako se koristi logička podela korpusa prilikom pravljenja konkordanci. Konkordance u Unitex-u su opisane kao liste redova. Svaki red se sastoji od teksta određene dužine sa leve strane izraza, izraza[22], i teksta određene dužine sa desne strane tog izraza. Izraz koje određuje red konkordance će se u daljem tekstu nazivati značajna reč. Konkordanca u Unitex-u može biti napravljena u više formata. Podrazumevani format je hipertekst, i u tom slučaju značajna reč je hiperveza[23] koja pokazuje na mesto gde se ta reč nalazi u korpusu.

U velikim korpusima, naročito u onim koji se sastoje od više dokumenata je veoma korisno da u konkordanci uz položaj značajne reči takođe na neki način označimo u kojoj logičkoj celini tog korpusa se svaka značajna reč nalazi.

U Unitex-u prvo se pravi indeks rezultata pretraga i potom se taj indeks koristi za pravljenje konkordance naredbom [1]:

Concord [OPCIJE] <indeks>

Da bi se koristio sadržaj prilikom pravljenja konkordance opcijama je dodata još jedna opcija:

- `-c:` ova opcija nema argumenata i program pretpostavlja da se sadržaj nalazi u istom katalogu kao i indeks značajnih reči.

U daljem tekstu ako je rečeno da je konkordanca pravljena sa korišćenjem sadržaja to znači da je prilikom konstrukcije konkordance korišćena ova opcija.

U primeru 16. se vidi konkordanca korpusa iz primera 10 navedenog u prethodnom poglavlju, sa korišćenjem sadržajem takođe prikazanim u prethodnom poglavlju. Indeks reči korišćen za pravljenje ove konkordance je napravljen unosom teksta „rečenica“ u program za pretragu.

```
1.1.    Test primer {TB}Prvi Naslov{TE} Jedna rečenica.{S} Još jedna rečenica. {TB}Drugi Naslov{TE} P
1.1.  Naslov{TE} Jedna rečenica.{S} Još jedna rečenica. {TB}Drugi Naslov{TE} Ponovo jedna rečenica.{S
1.2.  nica. {TB}Drugi Naslov{TE} Ponovo jedna rečenica.{S} Još jedna rečenica. {TB}Treći Naslov{TE} {
1.2.  TE} Ponovo jedna rečenica.{S} Još jedna rečenica. {TB}Treći Naslov{TE} {EOF}Druga_Datoteka{TE}
2.1.  ka{TE} {TB}Naslov u drugoj datoteci{TE} Rečenica u drugoj datoteci. {TB}Drugi Naslov U Drugoj D
```

Primer 16. Hipertekst konkordanca napravljena korišćenjem sadržaja

Kao što vidimo na primeru tabela ima dve kolone, za razliku od tabele u konkordanci koja je napravljena bez korišćenja naslova koja ima samo jednu kolonu. Dodata je prva kolona koja sadrži podatke koji nam govore u kojoj logičkoj celini se nalazi odgovarajuća značajna reč. Ti podaci su prikazani kao hiperveza, gde korisnik vidi redni broj. Ta hiperveza pokazuje na naslov logičke celine (koristeći format koji je opisan u prethodnom poglavlju), i ima atribut naslova podešen na naslov logičke celine. Ovo se može detaljnije videti na sledećem primeru koji nam pokazuje izvorni kod tog hiperteksta. Na njemu su podebljani vidljivi delovi hiperveza kao i pozicije naslova logičke celine i značajne reči.

[22] Ovo je u Unitex-u rezultat pretrage. Taj rezultat je izraz pošto Unitex vrši pretrage po izrazima.
[23] Format te hiperveze je nestadaran i ona se može koristiti samo u Unitex-ovom prikazivaču konkordanci.

Pošto grafički inderfejs koristi liste za prikazivanje konkordanci, atribut naslova nije vidljiv sem ako se konkordanca ne prikaže u nekom od programa za prikazivanje hiperteksta, što je omogućeno u Unetx-u[24].

```
<html lang=en>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>5 matches</title>
</head>
<body>
<table border="0" cellpadding="0" width="100%" style="font-family: 'Courier new'; font-size:
12">
<tr><td nowrap><a href="17 28 1" Title="Prvi Naslov">1.1.</a></td><td
nowrap>  Test primer {TB}Prvi Naslov{TE} Jedna <a href="40 48
1">rečenica</a>.{S} Još jedna rečenica. {TB}Drugi Naslov{TE} P </td></tr>
<tr><td nowrap><a href="17 28 1" Title="Prvi Naslov">1.1.</a></td><td nowrap>Naslov{TE}
Jedna rečenica.{S} Još jedna <a href="63 71 2">rečenica</a>. {TB}Drugi Naslov{TE}
Ponovo jedna rečenica.{S </td></tr>
<tr><td nowrap><a href="78 90 2" Title="Drugi Naslov">1.2.</a></td><td nowrap>nica.
{TB}Drugi Naslov{TE} Ponovo jedna <a href="109 117 2">rečenica</a>.{S} Još jedna
rečenica. {TB}Treći Naslov{TE} { </td></tr>
<tr><td nowrap><a href="78 90 2" Title="Drugi Naslov">1.2.</a></td><td nowrap>TE} Ponovo
jedna rečenica.{S} Još jedna <a href="132 140 3">rečenica</a>. {TB}Treći Naslov{TE}
{EOF}Druga_Datoteka{TE}  </td></tr>
<tr><td nowrap><a href="193 217 3" Title="Naslov u drugoj datoteci">2.1.</a></td><td
nowrap>ka{TE} {TB}Naslov u drugoj datoteci{TE} <a href="223 231
3">Rečenica</a> u drugoj datoteci. {TB}Drugi Naslov U Drugoj D </td></tr>
</table></body>
</html>
```

Primer 17. Izvorni kod konkordance napravljene korišćenjem sadržaja

Da bi se naredbi za pravljenje konkordanci dodala opcija za korišćenje sadržaja promenjene su sledeće tri datoteke `Concord.cpp, Concordance.h, Concordance.cpp`.

U datoteci `Concordance.h` je promenjena definicija strukture `conc_opt`, koja opisuje opcije za pravljenje konkordance. Toj strukturi je dodata lista koja opisuje sadržaj.

Datoteka `Concord.cpp` sadrži funkciju `main_Concord` koja se poziva naredbom, parsira opcije, stavlja dobijene podatke u promenljivu `options` tipa `conc_opt` i poziva metodu koja kreira konkordancu sa datim opcijama. U toj datoteci je omogućeno da prihvati opciju za korišćenje sadržaja i ako je ta opcija uneta onda se u promenljivu `options` u listu sadržaja učitavaju podaci iz datoteke `contains.txt` iz istog kataloga u kome se nalazi indeks značajnih reči.

U datoteci `Concordance.cpp` se nalazi funkcija `create_concordance` koja pravi datoteku konkordance. Ta funkcija prvo poziva funkciju `create_raw_text_concordance`, koja pomoću indeksa sakuplja podatke potrebne za kreiranje konkordance i snima ih u neformatiranu tekst

[24] Napomena: u tom slučaju nije moguće koristiti hiperveze

datoteku, zatim učitava podatke iz te datoteke i koristi ih da napravi konkordancu prema zadatim opcijama.

Funkcija `create_raw_text_concordance je` modifikovana tako da ako u promenljivoj koja opisuje opcije je učitana lista sadržaja na kraju svakog reda teksta se dodaje redni broj, naslov i pozicija naslova logičke celine u kojoj se nalazi značajna reč koja odgovara tom redu.

Funkcija `create_concordance je` modifikovana tako da ako u promenljivoj koja opisuje opcije je učitana lista sadržaja, tada se sa kraja svakog reda teksta učitava redni broj, naslov i pozicija naslova logičke celine u kojoj se nalazi značajna reč koja odgovara tom redu, u promenljive koje su dodate u tu svrhu, i one se koriste prema zadatim opcijama prilikom pravljenja datoteke konkordance. U slučaju da se datoteka konkordance pravi u hipertekst formatu dodaje se još jedna kolona tabeli konkordance, a u tekst formatu datoteci se dodaje samo redni broj na početku svakog reda.

Takođe u ovoj datoteci je modifikovana funkcija koja vrši inicijalizaciju opcija za pravljenje konkordance tako da se lista sadržaja postavi na početku na praznu listu. Kao što je već navedeno ukoliko je lista sadržaja promenljive opcija za pravljenje konkordance prazna program kreira konkordancu bez korišćenja logičke podele korpusa, dok ako ta lista nije prazna kreirana konkordanca je indeksirana prema logičkim celinama opisanim tom listom sadržaja.

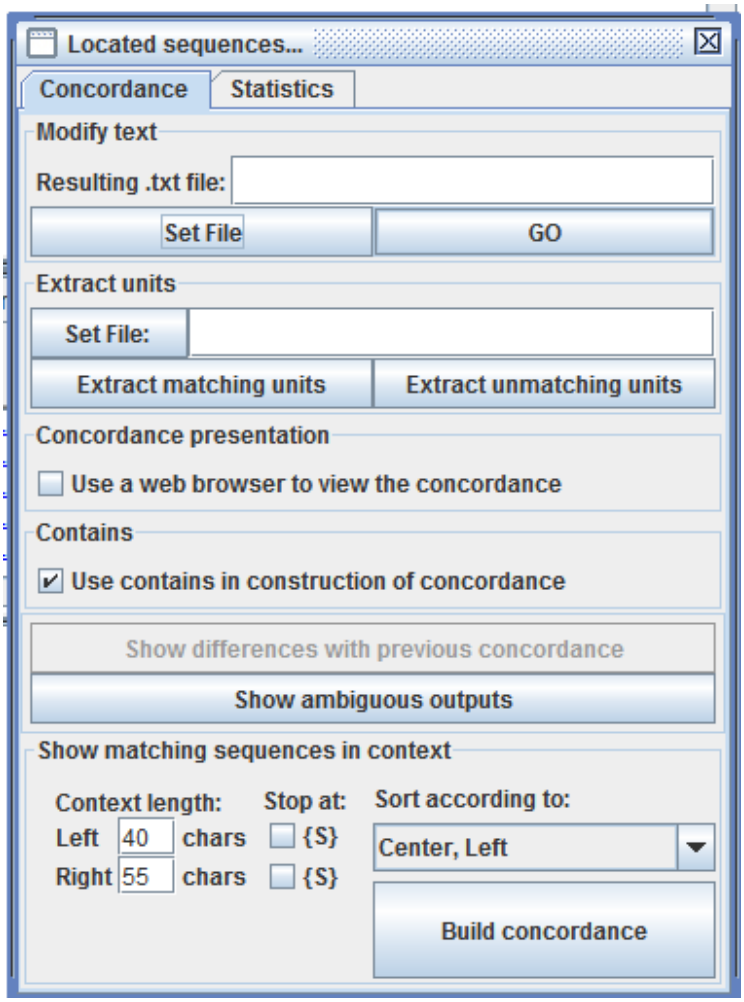
```
public ConcordCommand useContains(boolean cnt) {
	if (cnt)
		element("-c");
	return this;
}
```

Primer 18 Funkcija koja dodaje –c među opcije naredbe za pravljenje konkordance

Potrebno je omogućiti da se ova opcija može koristiti kada se konkordance prave iz grafičkog interfejsa. To je urađeno tako što je klasi čiji objekti kreiraju naredbe za pravljenje konkordanci dodata funkcija sa primera 18.

Klasa čiji objekti kreiraju naredbe za pravljenje je klasa `ConcordCommand` iz paketa `fr.umlv.unitex.process.commands.` Dodata funkcija se zove `useContains`, i ona ima logičku promenljivu kao argument. Ako se pozove ta funkcija i vrednost njenog agrumenta je tačno tada kada se formuliše naredba za pravljenje konkordance objektom klase `ConcordCommand` u naredbu se dodaje opcija za korišćenje sadržaja. Na taj način tu funkciju vezujemo za polje za potvrđivanje.

Prozor za pravljenje konkordanci, koji je pokazan na primeru 14, je objekat klase `ConcordanceParameterFrame` iz paketa `fr.umlv.unitex.frames.`



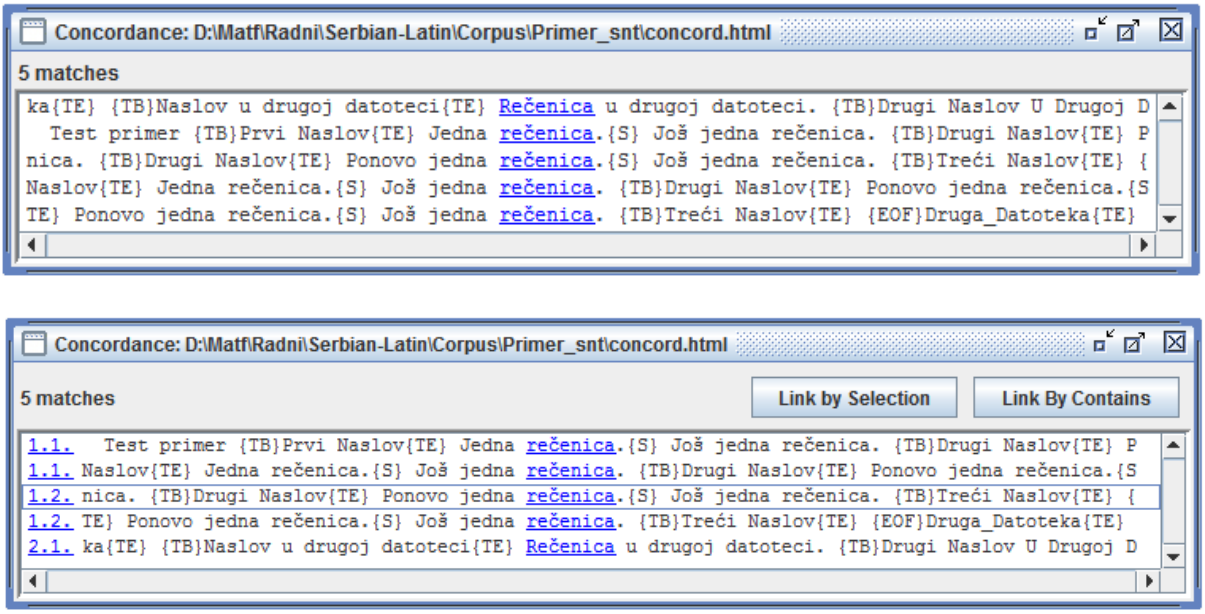
Primer 19. Prozor za pravljenje konkordanci

Ovom prozoru je dodat panel za rad sa sadržajem. Taj panel u sebi ima samo jedno polje za potvrdu. Kada je to polje odabrano, i dugme za pravljenje konkordance bude aktivirano, tada se pravi konkordanca indeksirana logičkim celinama opisanim sadržajem napravljenim tokom prethodne obrade korpusa, kao što je opisano u odeljku o pravljenju sadržaja.

U tu svrhu u klasu `ConcordanceParameterFrame` je dodata funkcija `constructContainsPanel` koja konstruiše panel za rad sa sadržajem. Promenjene su funkcije `constructConcordancePanel i buildConcordance.`

Funkciji `constructConcordancePanel` koja pravi prozor sa primera 14. je dodat kod koji poziva funkciju `constructContainsPanel` i dodaje rezultirajući panel prozoru.

U funkciju `buildConcordance` u kodu koji pravi komandu za pravljenje konkordance se povezuje polje za izbor sa prethodno opisanog panela sa objektom tipa `ConcordCommand` preko funkcije sa primera 18.



Primer 20 Prozor za prikaz konkordanci sa sadržajem (dole) i bez sadržaja (gore)

Pošto se izvrši komanda za pravljenje konkordanci grafički inderfejs Unitex-a automatski pravi prozor koji prikazuje tu konkordancu. Konkordance se u grafičkom inderfejsu Unitex-a prikazuju kao liste sa omogućenim skrolovanjem. Svaki element te liste je jedan red konkordance, to jest jedan red hipertekst tabele iz hipertekst dokumenta konkordance[25].

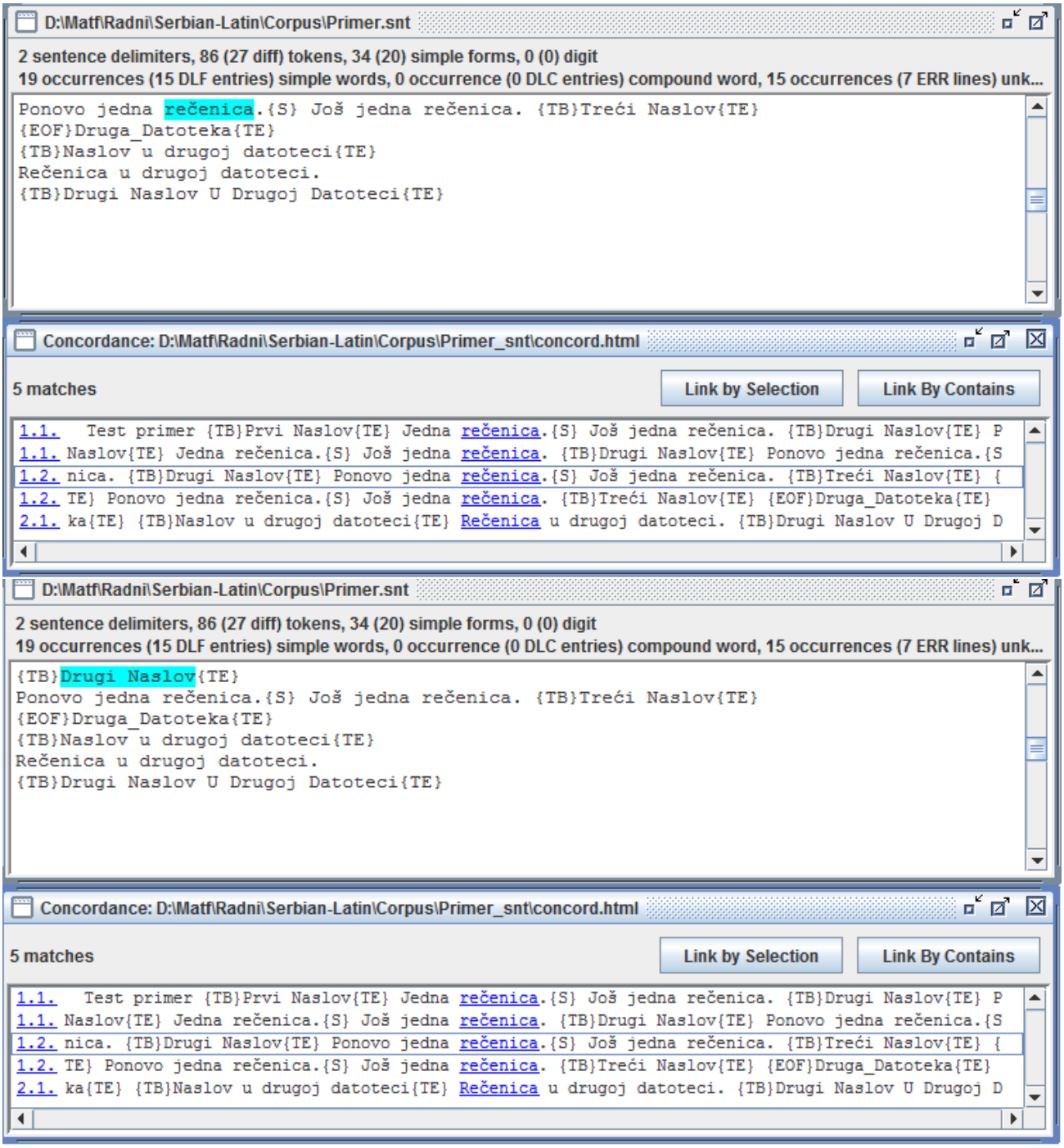
Pošto postoji značajna razlika[26] između konkordanci koje su pravljene sa primenom sadržaja i konkordanci koje su pravljene bez primene sadržaja potrebno je da postoje dve vrste prozora za prikazivanje tih konkordanci. To se jasno vidi na primeru 15. Kao što se vidi iz tog primera, vidljiva razlika između tih prozora je to da u prozoru za prikazivanje konkordanci koje su napravljene sa primenom sadržaja postoje dodatna dva dugmeta u gornjem desnom uglu. Ova dva dugmeta određuju kako se ta lista ponaša kada korisnik izabere element iz nje[27].

Kada korisnik klikne na jedan element liste koja prikazuje konkordancu koja ja napravljenja bez primene sadržaja tada Unitex pozove prozor za prikaz korpusa i u njemu zasenči značajnu reč tog reda konkordance. Pozicija značajne reči se nalazi na mestu adrese odredišta hiperveze i to u već napomenutom formatu[28], znači pozicija početka značajne reči u karakterima, pozicija kraja značajne reči u karakterima i redni broj rečenice u kojoj se ta značajna reč nalazi.

Problem je da kod konkordanci koje su napravljene sa primenom sadržaja u svakom redu liste koje je predstavlja imamo dva podatka koji se koriste: položaj značajne reči i položaj naslova logičke celine. Pošto su konkordance prikazane kao liste, prilikom selekcije reda iz njih se obavlja samo jedna akcija. Da bi bilo moguće koristiti novounesene podatke treba omogućiti izbor koji je od ta dva podatka trenutno značajan.

Na primeru 21. vidimo kako dva dugmeta dodata prozoru za prikazivanje konkordanci, koje su napravljenje sa primenom sadržaja, rešavaju taj problem.

[25] Iako to izgleda kao hipertekst, to je lista redova koji se mogu izabrati
[26] Konkordanca sa sadržajem ima kolonu više
[27] Pošto nije moguće da se direktno klikne na hiperveze
[28] Kod formata pozicije naslova logičke celine



Primer 21. Rezultat izbora elemenata liste koja prikazuje konkordancu kada je aktiviran mod za povezivanje po značajnoj reči (gore) i kada je aktivaran mod za povezivanje po sadržaju (dole)

Prvo dugme, gledajući sa leva u desno, postavlja listu u režim rada koji vrši povezivanje po značajnoj reči. U tom režimu rada kada korisnik izabere element liste tada Unitex pozove prozor za prikaz korpusa i u njemu zasenči značajnu reč tog reda konkordance.

Drugo dugme, gledajući sa leva u desno, postavlja listu u režim rada koji vrši povezivanje po sadržaju. U tom režimu rada kada korisnik izabare element liste tada Unitex pozove prozor za prikaz korpusa i u njemu zasenči naslov logičke celine u kojoj se nalazi značajna reč tog reda konkordance.

Prozori za prikazivanje konkordanci, koji je prikazan na primeru 15, su objekti klase `ConcordanceFrame` iz paketa `fr.umlv.unitex.frames.`

Da bi se omogućilo kreiranje prozora za prikaz konkordanci sa sadržajem konstruktoru te klase dodat je argument `cont`. Ovaj argument određuje da li se pravi prozor za prikaz konkordanci sa ili bez sadržaja. Takođe u konstruktor je dodata if-petlja koja kreira gore opisana dugmeta, ako taj argument ima vrednost tačno.

U klasu je dodata i promenljiva koje određuje da li se za izbor u prozoru korpusa koriste pozicije iz prve ili druge hiperveze. Ta promenljiva se menja prilikom klika na gore opisana dugmeta.

Prebrojavanje Lema

Tokom procesa tokenizacije Unitex pravi listu tokena sa njihovim frekvencijama pojavljivanja u korpusu. Program koji je opisan u ovom odeljku dodatno sređuje tu listu. On grupiše tokene prema njihovim lemama i tako pravi listu lema koje se poljavjuju u tekstu pri čemu se svakoj lemi dodaje broj pojavljivanja reči određenom tom lemom u korpusu. Time se vrši jedna analiza frekvencija reči.

Program pozivamo naredbom:

LemmataCounter [Opcije] <datoteka korpusa>
Uz ovaj progam postoji samo jedna opcija i to je:

- `-h/--help:` koja poziva uputsvo za korišćenje ovog programa

Kada se pokrene ovaj program isti koristi datoteke elektronskog rečnika korpusa da formira listu lema sa tokenima koji odgovaraju toj lemi, i zatim iskoristi tu listu da pomoću liste tokena izračuna sumu broja pojavljivanja svake leme i svakog tokena koji odgovara toj lemi, uredi tu listu prema alfabetskom redu i prema broju pojavljivanja i snimi te liste u odgovaruće datoteke. Pošto ovaj program koristi listu tokena i lokalni rečnik korpusa, na taj korpus treba primeniti proces tokenizacije i odgovarajuće elektronske rečnike pre nego što je moguće koristiti ovaj program[29].

Name	Date modified	Type	Size
dlc	6/17/2011 3:29 PM	File	1 KB
dlc	6/17/2011 3:29 PM	N File	1 KB
dlf	6/17/2011 3:29 PM	File	1 KB
dlf	6/17/2011 3:29 PM	N File	1 KB
enter	6/17/2011 3:29 PM	POS File	1 KB
err	6/17/2011 3:29 PM	File	1 KB
err	6/17/2011 3:29 PM	N File	1 KB
snt_offsets	6/17/2011 3:29 PM	POS File	1 KB
stat_dic	6/17/2011 3:29 PM	N File	1 KB
stats	6/17/2011 3:29 PM	N File	1 KB
tags	6/17/2011 3:29 PM	IND File	1 KB
tags_err	6/17/2011 3:29 PM	File	1 KB
tags_err	6/17/2011 3:29 PM	N File	1 KB
text	6/17/2011 3:29 PM	C/C++ Code Listing	1 KB
tok_by_alph	6/17/2011 3:29 PM	Text Document	1 KB
tok_by_freq	6/17/2011 3:29 PM	Text Document	1 KB
tokens	6/17/2011 3:29 PM	Text Document	1 KB
wrd_by_alph	6/17/2011 3:29 PM	Text Document	1 KB
wrd_by_freq	6/17/2011 3:29 PM	Text Document	1 KB

Primer 22. Jedan SNT katalog u kome su prebrojane leme

[29] Ukoliko nedostaje ili lista tokena (`tok_by_alph.txt`) ili lokalni rečnik korpusa (`dlf`) u SNT katalogu korpusa program će javiti grešku.

Na primeru 22. vidimo kako katalog korpusa iz ranijih primera izgleda posle primene ovog programa. Datoteke u koje upisuju liste lema su `wrd_by_alph.txt` i `wrd_by_freq.txt` .

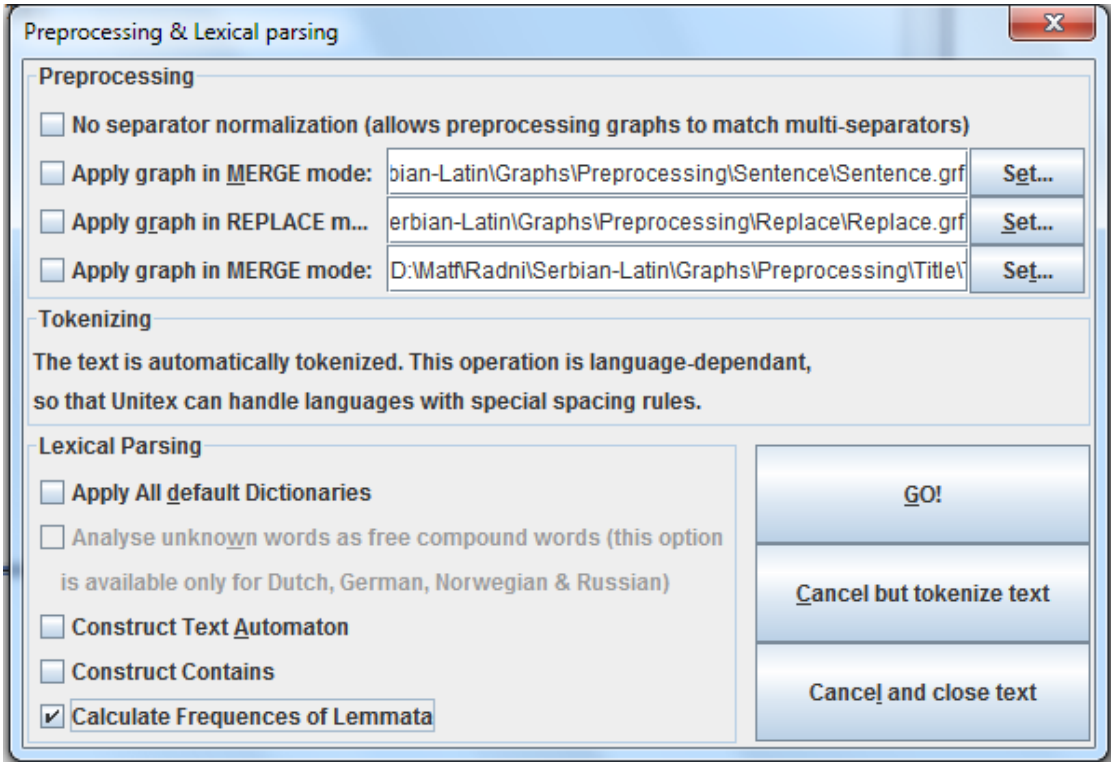
U datoteci `UnitexTool.cpp` u nizu `utility_array` su dodate informacije koje omogućavaju da naredba `LemmataCounter` pokrene odgovarajući program. Takođe je u toj datoteci dodata predprocersorska direktiva koja uključuje datoteku `LemmataCounter.h`.

Program koji prebrojava leme se nalazi u datotekama `LemmataCounter.h` i `LemmataCounter.cpp`. Komponente ovog programa su:

- `main_LemmataCounter`: glavna funkcija koje se poziva naredbom `LemmataCounter`. Ona otvara datoteku `dlf` SNT katalogu korpusa unetog naredbom i koristi je u dole opisanoj funkciji `read_dlf` da napravi listu lema. Potom otvara datoteku `tok_by_alph.txt` i koristi je u dole opisanoj funkciji `add_frq` da se listi lema dodaju frekfencije. Na kraju koristi funkciju `sort_and_save_by_frequence` da uredi i snimi listu lema u datoteku `wrd_by_freq.txt` i funkciju `sort_and_save_by_alph_order` da uredi i snimi listu lema u datoteku `wrd_by_alph.txt;`
- `optstring_LemmataCounter` i `lopts_LemmataCounter`: promenljiva i niz kojim opisujemo moguće opcije. One takođe služe za parsiranje opcija koje su navedene uz naredbu;
- `usage_LemmataCounter`: promenljiva koja sadrži tekst uputsva;
- `Word`: struktura koja u sebi sadrži lemu, listu tokena koji joj odgovaraju i broj pojavljivanja tih tokena u korpusu;
- `new_Word`: funkcija koja inicijalizuje promenljivu tipa `Word`. Ona kao argument ima lemu i jedan od tokena koji joj odgovaraju (to može biti i sama lema);
- `free_Word`:funkcija koja oslobađa memoriju rezervisanu za promenljivu tipa `Word`;
- `add_alias_to_Word`: funkcija koja dodaje token promenljivoj tipa `Word`, u listu tokena u toj promenljivoj ako se taj token već ne nalazi u toj listi. Ako se nalazi funkcija vraća 1, u suprotnom 0;
- `is_lemma`: funkcija koja određuje da li je dati token lema date promenljive tipa `Word`;
- `get_lemma`: funkcija koja vraća lemu date promenljive tipa `Word`;
- `get_frq`: funkcija koja vraća frekvencu date promenljive tipa `Word`;
- `is_alias`: funkcija koja proverava da li dati token pripada listi tokena date promenljive tipa `Word`;
- `sort_and_save_by_frequence`: funkcija koja uređuje niz promenljivih tipa `Word` prema broju pojavljivanja lema i snima uređenu listu u datu datoteku;
- `sort_and_save_by_alph_order`: funkcija koja uređuje niz promenljivih tipa `Word` prema lemama u azbučnom redu i snima uređenu listu u datu datoteku;
- `read_dlf`: funkcija koja iz datoteke lokalnog rečnika korpusa (datoteka `dlf` u katalogu sa primera) pravi niz promenljivih tipa `Word` od koji svaki element sadrži lemu i listu tokena koji je pripadaju;
- `add_frq`: funkcija koja koristeći datoteku liste tokena sa frekvencama datom nizu promenljivih tipa `Word` pridodaje svakom elementu tog niza sumu frekvenci svih tokena koje se nalaze u listi tokena tog elementa;

- `find_lemma`: funkcija koja iz niza promenljivih tipa `Word` nalazi onaj element tog niza čija lema odgovara datoj;
- `extract_lemma_and_alias`: funkcija koja iz jednog elementa elektronskog rečnika pravi par leme i token koji odgovara toj lemi;
- `extract_tok_and_frq`: funkcija koja iz reda datoteke koja opisuje listu tokena sa frekvencama pojavljivanja parsira token i njegov broj pojavljivanja u korpusu.

Pošto je omogućeno prebrojavanje lema u konzoli Unitex-a, potrebno je omogućiti da se naredba pozove iz grafičkog interfejsa kao i da se prikažu dobijeni rezultati. Pošto se prebrojavanje vrši uobičajno jedanput, i to posle tokenizacije i primene elektronskih rečnika na korpus, ta naredba se poziva tokom prethodne obrade korpusa, na kraju leksičkog parsiranja.



Primer 23. Dijalog za prethodnu obradu teksta u kome je označena polje za prebrojanje lema

U tu svrhu u dijalog za prethodnu obradu teksta, u panelu za leksičko parsiranje, kao što se vidi na primeru 23, dodato je polje za potvrđivanje nazvano `Calculate Frequneces of Lemmata`. Ako je ono aktivirano kada se klikne na dugme `GO!` konzoli se, među ostalima, šalje i sledeća komanda:

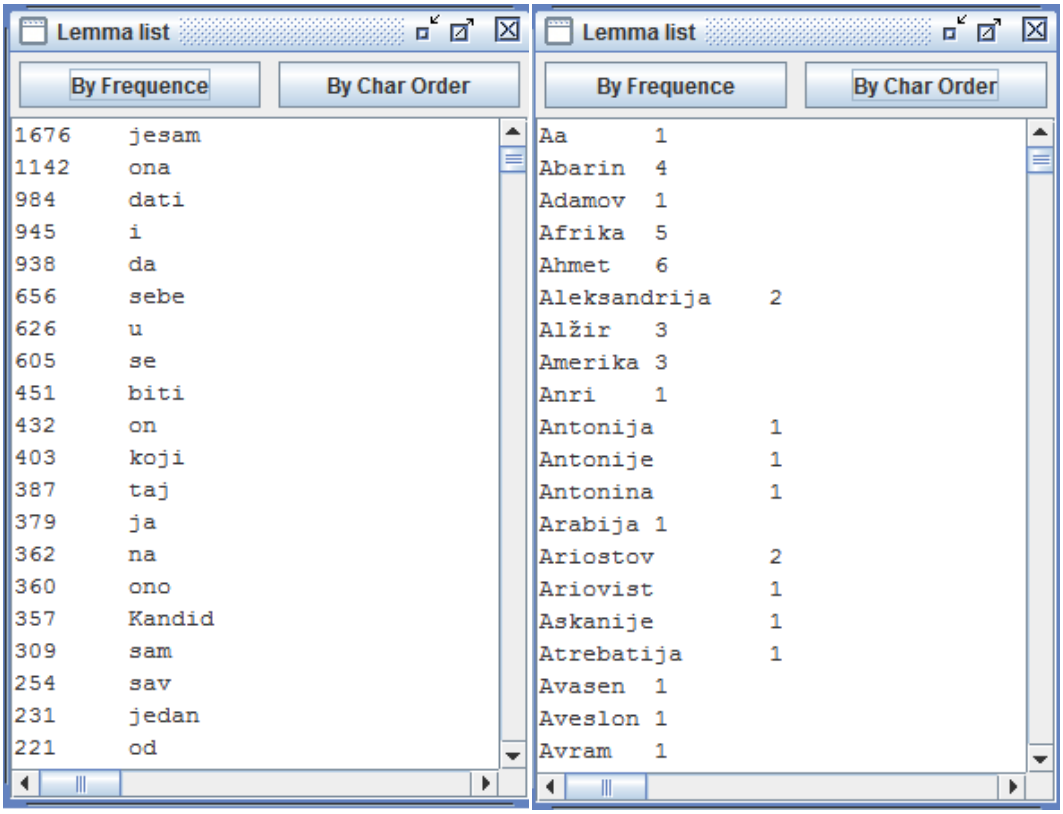
```
LemmataCounter <ime datoteke aktivnog korpusa>
```

U tu svrhu je promenjena klasa `PreprocessDialog` iz paketa `fr.umlv.unitex.frames`. Njoj je dodat objekt `lemmataCheck`, koji predstavlja polje za potvrdu. Takođe su promenjene metode `preprocess`, koja izdaje naredbe kada korisnik klikne na dugme `GO!`, i `construcLexicalParsingPanel`, koja konstruiše panel za leksičko parsiranje. U metodi `construcLexicalParsingPanel` je panelu za leksičko parsiranje dodato polje za izbor da li se prebrojavaju leme, a metodi `preprocess` dodato je da se pomoću objekta klase `LemmataCountCommand`, sa primera 23, doda naredba za prebrojavanje lema naredbama koje izvršavaju, ako je aktivirano polje za izbor koje odgovara objektu `lemmataCheck.`

```
public class LemmataCounterCommand extends CommandBuilder{
	public LemmataCounterCommand() {
		super("LemmataCounter");}
	public LemmataCounterCommand text(File s) {
		protectElement(s.getAbsolutePath());
		return this;}}
```

Primer 24. Java klasa za pravljenje naredbe `LemmataCountCommand`

Java klasa `LemmataCountCommand`, koja pravi naredbu za prebrojavanje lema, se nalazi u paketu `fr.umlv.unitex.process.commands.` Ova klasa ima samo metodu `text` koja u komandi postavlja datoteku korpusa koji se obrađuje.



Primer 25. Prikaz uređenje liste lema, po frekfenci (levo) i alfabetu (desno)

Na primeru 25 vidimo prozor koji služi za prikazivanje lista lema. Ovaj prozor se pojavljuje posle učitavanja korpusa i nestaje kada se on zatvori, bez obzira da li datoteke lista lema postoje u katalogu tog korpusa.

Kao što se vidi na primeru ovaj prozor na vrhu ima dva dugmeta koja određuju da li se prikazuje lista uređena po alfabetu (dugme desno) ili po frekvenci (dugme levo).

Ovaj prozor je objekat klase `LemmataFrame` iz paketa `fr.umlv.unitex.frames`. Ova klasa je veoma slična klasi koja prikazuje liste tokena, što je i logično sa obzirom da su liste lema obrađene liste tokena. Jedina značajna razlika su akcije koje se pokreću klikom na dugmeta iz ove klase.

Važniji objekti u ovaj klasi su lista `text` koja se definiše istom klasom kao i kod prozora za prikazivanje tokena, dugmeta koje određuju koja se lista pokazuje, i panel za prikazivanje teksta. Akcija koje se pokreće dugmetom koje uređuje listu po alfabetu je učitavanje datoteke

`wrd_by_alph.txt` u listu teksta za prikazivanje. Drugo dugme kada se klikne na njega pokreće akciju koja učitava datoteku `wrd_by_freq.txt` u listu teksta za prikazivanje.

Objekt ove klase kreira pomoću objekta klase `LemmataFrameFactory`, koji pod imenom `containsFrameFactory` polje klase `InternalFrameManager`, koje služi za rad okvirima u grafičkom interfejsu Unitex-a. Takođe je promenjena klasa `UnitexFrame`, tako da kada god bude učitan korpus pravi se prozor za prikazivanje lista lema koji odmah učitava datoteku `wrd_by_alph.txt` iz SNT kataloga tog korpusa, i da se taj prozor zatvara sa korpusom.

Zaključak

U radu su opisane promene u sistemu programa Unitex koje u tom sistemu dodaju elemente logičkog izgleda teksta.

C++ deo koda sistema programa Unitex je izmenjen tako da omogućava korišćenje dodatnih etiketa za podelu korpusa na logičke celine. U poglavlju "Ubacivanje Dodatnih Etiketa" pokazano je tačno koje delove koda treba promeniti da bi Unitex prihvatio etikete kao legitimne. Ako pri daljem usavršavanju ovog sistama programa bude bilo potrebno da se uvede još dodatnih etiketa sa tim opisom lako se može naći deo koda koji treba promeniti u tu svrhu. Etikete trenutno dele korpus pomoću naslova prvog i drugog nivoa. Omogućeno je takođe u grafičkom intefejsu i da se primenom grafova na korpusu izvrši njegova logička podela. Pošto su ti grafovi zavisni od korpusa, kao primer je dat graf koji deli roman na poglavlja.

Sistemu programa Unitex dodat je program koji spaja više korpusa u jedan, podeljen na celine koje odgovaraju korpusima pre spajanja. Podela je urađena korišćenjem etiketa predviđenih u tu svrhu, čije je korišćenje prethodno omogućeno. U grafičkom interfejsu je omogućeno da se korišćenjem ovog programa trenutno otvorenom korpusu doda jedan korpus. U daljem radu bi možda bilo korisno da se naprave programi koji uklanjaju, kopiraju i pomeraju ovakve logičke celine u korpusu, i da se pomoću tih programa u grafičkom interfejsu napravi bolji interfejs za rad sa više datoteka.

Time je omogućena podela korpusa na logičke celine, korišćenjem etiketa. Dalje je napravljen program koji od korpusa podeljenog oznakama pravi dve datoteke sadržaja, jednu za prikazivanje i drugu koja se koristi prilikom pravljenja konkordanci. Korišćenje tog programa je omogućeno u grafičkom interfejsu, tokom prethodne obrade korpusa, kao i prikazivanje sadržaja korpusa.

Ovim je obezbeđena logička podela korpusa u sistemu programa Unitex. Time je unapređena obrada velikih korpusa, i omogućen rad sa više datoteka istovremeno.

Programu koji od indeksa reči pravi konkordance dodata je opcija koja omogućava korišćenje sadržaja tokom pravljenja konkordanci. Sadržaj se koristi da bi se značajne reči indeksirale prema logičkim celinama. U grafičkom interfejsu je dijalogu za kreiranje konkordanci dodata ta opcija. Pošto se konkordance napravljenje sa tom opcijom razlikuju od onih bez te opcije, promenjen je i prozor za prikazivanje konkordanci.

Promenom programa za konstrukciju konkordanci je omogućeno se obave pretrage velikih tekstova podeljenih na poglavlja, i više tekstova istovremo. U rezultatima tih pretraga su prisutne informacije iz kog poglavlja teksta ili kog teksta je svaki dobijeni red konkordance. Time je ispunjen cilj ovog rada.

Pored toga ugrađen je program za prebrojavanje lema, koji nalazi sve leme iz korpusa i prebrojava koliko tokena koji odgovaraju tim lemema se nalaze u korpusu.

Sve napred probrojane promene su urađenje radi poboljšanja i lakšeg korišćenja sistema programa Unitex za analizu tekstova u različite svrhe.

Literatura

[1] Sébastien Paumier, *Unitex User Manuel* English translation of version 1.2 by Wolfgang Flury, Franz Guenthner, Friederike Malchok, Clemens Marschner, Sebastian Nagel, Johannes Stiehler

[2] Unitex Corpus Processor Home Page, http://www-igm.univ-mlv.fr/~unitex/

[3] Text corpus from Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Text_corpus

[4] Concordance (publishing) from Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Concordance_(publishing)

[5] The Relex Network, http://infolingu.univ-mlv.fr/english/Relex/Relex.html