Tomaž Erjavec
Institut Jožef Stefan, Odsek za inteligentne sisteme
tomaz.erjavec@ijs.si

UDK 811.163.6'42:004.5

OZNAČEVANJE KORPUSOV

Urejene računalniške zbirke besedil, korpusi, so uporabni v številnih vejah jezikoslovja. Prispevek se osredotoči na računalniški zapis korpusov, predvsem na njihovo označevanje, tj. dodajanje interpretativnih informacij besedilu v korpusu. Predstavljeno označevanje temelji na upoštevanju mednarodnih standardov s tega področja, kar doprinese k boljši dokumentiranosti in preverljivosti, enostavnejši uporabi programov za obdelavo in k večji izmenljivosti ter trajnosti. V prvem delu obravnavamo standarde za računalniški zapis besedil, predvsem XML (eXtended Markup Language) in TEI (Text Encoding Initiative), v drugem pa orišemo nekaj bolj zanimivih ravni jezikoslovnega označevanja korpusov.

1 Uvod

Korpus je zbirka besedil, ki so izbrana tako, da karakterizirajo stanje ali raznovrstnost nekega jezika. Uporaben je kot osnova, na kateri gradimo opise jezika, ali pa kot sredstvo za preverjanje hipotez o jeziku. Korpusi so dandanes že standardno shranjeni na računalnikih, saj ti po eni strani omogočajo kompaktno in poceni hranjenje ter razširjanje velike količine besedil, po drugi strani pa ta besedila lahko z njimi bolj učinkovito izkoriščamo.

V tem članku se osredotočimo na računalniški zapis korpusov, predvsem na njihovo označevanje, tj. dodajanje interpretativnih informacij besedilu v korpusu. Termin označevanje ima dva pomena: po eni strani lahko s tem mislimo na proces dodajanja oznak, ki je lahko ročni ali pa, vse pogosteje, avtomatski. Čeprav v pričujočem članku omenimo orodja, ki jih lahko uporabimo za označevanje slovenskih besedil, pa bomo več pozornosti posvetili označevanju v drugemu pomenu, tj. kako so te dodane informacije v korpusu zapisane, ne glede na to, kako smo do njih prišli.

V zvezi z obema pomenoma označevanja se bralcu mogoče zastavi vprašanje, zakaj sploh potrebujemo drugi pomen, tj. zakaj bi korpus sploh eksplicitno označevali z dodatnimi informacijami, posebej če do njih tako ali tako pridemo po povsem avtomatski poti. Ali ni enostavneje teh informacij, npr. oblikoslovnih oznak besed v korpusu, izračunati sproti, ko in če jih potrebujemo, namesto da obremenjujemo z njimi besedilo? Čeprav je ta scenarij v posameznih primerih upravičen, je zelo omejujoč. Z dodajanjem oznak v sam korpus namreč omogočimo večjo eksplicitnost in

preglednost, s tem pa tudi večjo prenosljivost podatkov. Oznake v korpusu lahko tudi ročno popravljamo, s čimer povečamo njihovo kvaliteto, obenem pa dobimo učno množico za programe, ki se avtomatsko učijo modele jezika. S tako naučenimi programi potem lahko označujemo nove korpuse, prvi nivo oznak pa nam tudi služi kot osnova za bolj kompleksna označevanja.

V tem članku bomo zagovarjali zapis in označevanje korpusov, ki temelji na upoštevanju mednarodnih standardov s tega področja. Kot bomo videli, upoštevanje standardov prinese s sabo vrsto prednosti, npr. boljšo dokumentiranost, preverljivo pravilnost zapisa, enostavnejšo uporabo programov za obdelavo, večjo izmenljivost in trajnost itd.

V drugem poglavju obravnavamo standarde za zapis korpusov in drugih besedil, tretje poglavje oriše nekaj bolj zanimivih ravni označevanja korpusov, četrto poglavje pa poda zaključke.

2 Standardi za zapis korpusov

Precej zgodaj so se že začele pojavljati pobude za standardizacijo zapisa jezikovnih podatkov, pri katerih se skuša predpisati javno dostopne in trajne načine označevanja. Tako zapisane jezikovne podatke potem uporabljajo orodja, ki implementirajo te standarde, bodisi neposredno ali pa tako, da lahko vanje podatke uvažajo oz. izvažajo. Najbolj vplivna pobuda s tega področja je bil leta 1986 izdani ISO standard SGML (Standard Generalised Markup Language), ki določa jezik za predstavitev dokumentov, nad katerimi bodo delovali programi za procesiranje besedil. Ta standard naj bi zagotovil način zapisa, ki je prenosljiv med računalniškimi platformami, odporen na tehnološke spremembe in ki omogoča uporabo dokumentov v različne namene. Velika prednost SGML pred drugimi zapisi je tudi, da je mogoče avtomatsko preveriti, ali je nek dokument zapisan v skladu s standardom.

Standard SGML je bil dobro premišljen, vendar pa zelo kompleksen, zato je svojo nišo našel predvsem v podjetjih, ki posedujejo velike količine dragocenih besedil in so bila pripravljena v standardizacijo njihovega zapisa tudi investirati veliko denarja. Zaradi kompleksnosti je bilo na voljo razmeroma malo programov, ki so implementirala standard; ti so bodisi delovali na platformi Unix, ki se je uporabljala predvsem v akademskih krogih, ali pa so bili izredno dragi. Zaradi teh problemov SGML nikoli ni dosegel zares široke popularnosti.

S prodorom svetovnega spleta pa se je pojavila potreba, da se mrežnim aplikacijam zagotovi standardizirani in prenosljivi način zapisa podatkov; to ni mogel biti HTML, saj ima ta zelo omejene izrazne možnosti, niti SGML, zaradi svoje kompleksnosti. Rešitev, ki jo je izdelal konzorcij za svetovni splet W3C (Word Wide Web Consortium), je bila podmnožica SGML, ki ohranja dobre lastnosti standarda, vendar pa izpusti elemente, ki so vnašali pretirano kompleksnost. Ta izvedenka SGML se imenuje XML <http://www.w3.org/XML/> (eXtended Markup Language); prva različica specifikacije je bila objavljena leta 1998, druga, trenutno zadnja, ki je popravila nekatere napake iz prve, pa leta 2000.

Za razliko od SGML je XML postal izjemno popularen in dejansko postaja univerzalni medij zapisa jezikovnih podatkov. K njegovi odmevnosti je pripomoglo tudi veliko število prosto dostopnih programov, ki standard implementirajo, kot tudi kopica pridruženih standardov, ki ga dodatno osmislijo.

V izdelavo korpusov, posebej ročno označenih, je potrebno vložiti ogromno dela, potencialno pa so nato uporabni v raznovrstne namene, zato so se tudi tu že zgodaj pojavile pobude za standardizacijo zapisa. Največ na tem področju je naredila iniciativa za zapis besedil TEI (Text Encoding Initiative) s svojimi Priporočili, ki definirajo konkretne oznake za opis besedil v namene znanstvene obravnave; TEI je najprej temeljil na SGML, zadnja različica, ki je izšla 2002, pa podpira tudi XML.

V nadaljevanju tega poglavja podamo osnove XML in razložimo strukturo priporočil TEI.

2.1 Strukture XML

Standard XML formalno definira računalniški zapis besedila in uvede načine, kako lahko to besedilo označimo in strukturiramo. V Sliki 1 vidimo primer dokumenta XML; kot prvo lastnost zapisa je pomembno izpostaviti, da je dokument berljiv tudi neposredno v obliki XML. Čeprav ni mišljeno, da bi dokumente XML brali »surove«, pa je vseeno koristno, da jih lahko tudi brez posebnih orodij beremo in popravljamo.

Dokument XML vsebuje elemente, od katerih se vsak začne z začetno oznako, npr. <l>, in zaključi s končno, ki vsebuje poleg lomljenih oklepajev in imena elementa še poševnico. Element XML je tako sestavljen iz treh delov: obeh oznak in vsebine. Izjema so prazni elementi (primer je <pb/> spodaj, ki naj bi označeval prelom strani), ki imajo za razliko od tistih z vsebino samo eno oznako, ta pa se končuje s poševnico.

Elementi lahko vsebujejo besedilo ali pa spet elemente oz. mešanico obeh, celoten dokument pa mora vsebovati natanko en vrhnji element; v Sliki 1 je to . Dokumenti XML so tako strukture z dobro poznanimi formalnimi lastnostmi, t. i. drevesa.

```
<div n="14" id="jenko.14">
  <head>Uvod.</head>
  <lg>
    <l>Dvigni se! ukaz mi reče.</l>
    <l>Srce pade mi v oblasti</l>
    <l>Silne, prej neznane strasti,</l>
    <l>Ki ko živi ogenj peče.</l>
  </lg>
  <pb/>
  <lg>
    <l>Čut se zlije mi v besede. -</l>
    <l>Preč so črne bolečine,</l>
    <l>Strast občutkov divjih mine,</l>
```

```
    <l>Jasen mir se v prsi vsede.</l>
  </lg>
</div>
```

Slika 1: *Primer dokumenta XML*

Drevesni model označevanja je intuitivni in enostaven za računalniško obdelavo, ni pa zadosten za vse vrste struktur. Predvsem je problematično, kadar bi dokument radi kategorizirali v več ortogonalnih hierarhij, npr. če hočemo zajeti tipografsko in retorično strukturo besedila, saj so navzkrižna gnezdenja, npr. ... <page> ... ... </page>, prepovedana.

Obstaja več načinov, kako obidemo takšne probleme; enega smo že videli v Sliki 1, kjer je <pb/> zapisan kot prazni element, ki tako označuje prelom, namesto da bi stran vseboval. Splošni način, kako se izogniti omejitvam neposrednega XML označevanja, pa je uvedba referenčih mehanizmov. V modelu posrednega označevanja (stand-off annotation) je originalno besedilo nedotaknjeno, oznake pa se nahajajo v ločenih dokumentih in na besedilo samo kažejo. V praksi se izkaže, da najboljše rezultate daje mešanica obeh pristopov, saj je bolj splošni model posrednega označevanja ustrezno kompleksnejši, težje preverljiv in zato okoren predvsem za gradnjo virov.

Poleg samih elementov definira XML tudi sredstvo za izražanje njihovih lastnosti, skozi t. i. atribute, ki jih lahko vsebujejo začetne značke elementov; kot vidimo v Sliki 1, sledi imenu atributa enačaj in vrednost v navednicah.

Kot zadnjo lastnost dokumentov XML moramo omeniti nabore znakov in entitete za njihov zapis. Če ni drugače določeno, se privzame, da je dokument XML zapisan v skladu s standardom Unicode, ki vsebuje večino svetovnih pismenk. Vendar pa je ta zapis 16-bitni in ga je dostikrat iz tehničnih in zgodovinskih razlogov še vedno potrebno prevesti v manjše in manj splošne nabore.

XML definira splošno uporaben način za zamenjavo nizov, tj. enostavno in strojno neodvisno metodo, ki omogoča določiti, da se mora pri obdelavi določen niz znakov v dokumentu XML zamenjati z nekim drugim nizom. Nize, definirane s to metodo za nadomeščanje, imenujemo entitete. K temu pojmu se še povrnemo kasneje, tu pa le omenimo, da omogoči npr. prenos poljubnih znakov v naboru ASCII: med znaka za začetek (&) in konec (;) entitete pri tem napišemo kodo znaka iz kodnega nabora Unicode, npr. krščansko, kjer # pomeni, da sledi numerična koda, x pa, da je zapisana v heksadecimalnem zapisu; 161 in 10D sta v Unicode naboru znakov kodi za znaka *š* in *č*.

Nekaj bolj uporabnih znakov pa je v XML tudi vnaprej definiranih z mnemoniki; od teh sta najbolj pomembna & za & in < za <, saj ju je, kadar sta del besedila, potrebno zapisati z entitetami.

2.2 Definicija tipa dokumentov XML

Če bi dokumenti XML lahko vsebovali kakršne koli elemente v poljubnih medsebojnih odnosih in za elemente ne bi vedeli, kaj naj sploh pomenijo, ti dokumenti ne bi bili preveč uporabni. Seveda je možno, da je znanje o nekem tipu dokumentov vgrajeno v samo aplikacijo; primer tega so splenti brkljalniki, saj ti na zaslon izpišejo zapis HTML, četudi jim ne podamo formalne specifikacije oznak, ki jih HTML uporablja.

Vendar je izredno koristno imeti možnost, da lahko formalno definiramo nabor elementov za določen tip dokumentov; pri SGML je celo veljalo, da mora vsak dokument vsebovati ali se vsaj sklicevati na tako formalno specifikacijo, t. i. definicijo tipa dokumentov (Document Type Definition, DTD). Ravno zaradi lažje komunikacije med procesi, to pri XML ni potrebno, je pa možno. XML je od SGML podedoval (poenostavljen) mehanizem DTD-jev, ki uporablja poseben jezik, da definira skladnjo za elemente določenega tipa dokumentov.

Dokumenti XML, ki vsebujejo ali se sklicujejo na DTD, so pravilni (valid), tisti, ki pa se ne, vendar pa so vseeno zapisani po pravilih XML, pa so dobro oblikovani (well-formed); primer dobro oblikovanega dokumenta XML je bil podan v Sliki 1.

```
<!ELEMENT    div      (head?, (lg | pb)+)          >
<!ELEMENT    head     (#PCDATA)                    >
<!ELEMENT    lg       (#PCDATA | l)*               >
<!ELEMENT    l        (#PCDATA)                    >
<!ELEMENT    pb       (EMPTY)                      >
<!ATTLIST    div
             n        CDATA      REQUIRED
             id       ID         #IMPLIED          >
```

Slika 2: *Primer DTD XML*

DTD-ji omogočajo definicijo še raznih drugih gnezdenj elementov, vrednosti atributov, vsebujejo pa tudi mehanizme za modularizacijo, ki pa jih tu ne bomo obravnavali. Vseeno pa je treba omeniti entitete, ki smo jih že uvedli v prejšnjem razdelku; DTD namreč omogoča definicijo entitet kot poljubnih nizov ali kar celotnih datotek, ki jih nato lahko uporabimo v pravilnem dokumentu XML.

Pravilen dokument XML vsebuje DTD ali pa se nanj sklicuje s posebnim ukazom DOCTYPE, ki mora biti prvi v dokumentu. Če bi se npr. dokument v Sliki 1 začel z vrstico <!DOCTYPE div SYSTEM 'pesem.dtd'> in bi datoteka pesem.dtd vsebovala DTD iz Slike 2, bi bil ta dokument pravilen in ne samo dobro zapisan.

2.3 Pridruženi standardi

Uspešnost XML lahko pripišemo tudi obstoju pridruženih standardov, ki XML dodatno osmislijo, in prosto dostopnim orodjem, ki te standarde implementirajo; v tem poglavju omenimo samo nekatere najpomembnejše.

XML je prvenstveno jezik za opisovanje lastnosti, ne pa videza dokumenta. To je v splošnem izredno dobra lastnost, saj je za uporabnost dokumentov precej bolj pomembno, kaj določen del besedila pomeni in ne kako izgleda. Kljub temu pa je nek dokument slej ko prej potrebno prikazati bodisi na zaslonu ali pa na papirju. Jezik XSLT definira način, kako preoblikovati dokumente XML v druge formate, bodisi XML ali drugačne, ki vsebujejo formate za stavljenje, vendar pa nanje niso omejeni. XSLT je tako uporaben ne samo kot prikazovalnik, pač pa tudi kot splošno orodje za konverzijo iz nekega XML formata (DTD) v poljuben format, ki ga potrebuje določena aplikacija, ki naj bi dokument procesirala. Posebej privlačna je konverzija iz enega XML DTD v drugega, saj odpira možnost poljubnega preoblikovanja, selekcije in združevanja dokumentov XML. Pri tem pa je jezik XSLT tudi sam zapisan v XML, s čimer lahko uporabimo orodja XML za pisanje XSLT transformacij, obstajajo pa tudi prosto dostopni programi, ki implementirajo XSLT konverzije.

Kot je bilo že omenjeno, mehanizem DTD omogoča preverjanje pravilnosti dokumentov, kar je izredno koristno pri izdelavi in označevanju novih jezikovnih virov. Vendar pa imajo DTD-ji tudi vrsto pomankljivosti, predvsem to, da so zapisani v svojem jeziku in ne v XML, ter da podpirajo samo relativno enostavne podatkovne modele. V DTD npr. ni mogoče določiti, da mora biti vsebina nekega elementa število ali da je vrednost atributa datum. Zato je bilo razvitih več shem, ki presežejo te omejitve – tu naj omenimo samo dve. Prva je RELAX NG, ki je predlog standarda ISO in je osnovana na regularnih izrazih, druga pa XML Schema, ki je predlog konzorcija W3C. Shemi sta komplementarni, saj ima vsaka svoje prednosti. Obstajajo tudi orodja, ki implementirajo preverjanje skladnosti dokumentov XML po specifikaciji v teh shemah in prevajanje DTD-jev v izraze iz teh dveh shem.

2.4 Iniciativa za zapis besedil TEI

Iniciativa za zapis besedil TEI (Text Encoding Initiative) je bila ustanovljena leta 1987 pod pokroviteljstvom več mednarodnih združenj, nastala pa je z namenom, da se standardizira zapis besedil, ki bi se uporabljala pretežno v znanstvene namene, oz. da se razvije skupni način označevanja kompleksnih struktur besedil. S tem bi se zmanjšalo razdrobljenost obstoječih načinov digitalnega zapisa, poenostavilo računalniško obdelavo in spodbudilo razširjanje in izmenjevanje elektronskih besedil. Kmalu pa je postalo očitno, da bo zadosti prožna shema zapisa lahko nudila rešitve za splošne probleme zapisa besedil.

TEI je prvi osnutek svojih priporočil (TEI P1) izdal leta 1990, drugega pa leta 1992. Medtem ko sta bila tako P1 kot P2 še osnutka, predstavlja leta 1994 izdan TEI P3 zaključek prve faze dela TEI. TEI je kot osnovo svojega zapisa vzel SGML. TEI P3 je nabor fragmentov definicij tipov dokumentov, ki za široko paleto zvrsti besedil določa konkretne oznake in njihovo strukturo. Skorajda bolj pomembnih pa je 1200 strani dokumentacije, ki podaja pomen posameznih oznak, opisuje posamezne podsklope ter izpelje način za njihovo kombiniranje ter nadgradnjo. TEI P3 kot njegovi nasledniki so sicer izšli v knjižni obliki, so pa tudi prosto dostopni prek sve-

tovnega spleta. Leta 1999 je izšla popravljena izdaja TEI P3, ki je odpravila nekaj tipografskih in drugih napak, 2002 pa je bil izdan TEI P4 (Sperberg-McQueen in Burnard 2002), ki nudi enakovredno podporo za SGML in XML in popravi razne napake iz P3, pri čemer pa ohranja skladnost s TEI P3.

Leta 2000 je bil ustanovljen konzorcij TEI <http://www.tei-c.org/>, ki naj bi skrbel za razvoj standarda TEI. V okviru konzorcija je bil tako izdan TEI P4, ob tem pa so bila identificirana področja, kjer bi TEI bil potreben temeljitejše prenove. Zato je bilo ustanovljenih več delovnih skupin, ki imajo nalogo revidirati P4, da bo lahko (predvidoma) leta 2004 izdana nova različica TEI P5.

Več kot 80 projektov, ki pokrivajo prek 30 jezikov, je do sedaj uporabljalo priporočila TEI za zapis raznovrstnih virov, npr. za srednjeveške rokopise, tekstnokritične izdaje poezije, slovarje, knjižnične kataloge ipd. TEI je bil vpliven tudi pri označevanju korpusov, kjer je mogoče najbolj znan primer Britanski nacionalni korpus, pri nas pa so bili v skladu s priporočili TEI zapisani, med drugim, korpusi MULTEXT-East (Erjavec 2001), IJS-ELAN (Erjavec 2002a, 2001b) in FIDA (Erjavec et al. 1998); primere iz teh korpusov bomo srečali v nadaljevanju članka.

S TEI skladen DTD ustvarimo za potrebe konkretnega projekta s kombinacijo naborov oznak, ki jih definira TEI. Središčne oznake (core tags) so obvezne v vsakem TEI DTD. Središčne oznake tako določajo elemente, ki so na voljo v vseh TEI dokumentih (npr. oznake za naslove in odstavke) ter glavo dokumenta, ki vsebuje bibliografske in druge podatke o dokumentu. Osnovni nabori oznak (base tag sets) opisujejo različne zvrsti besedil, ki so med seboj razmeroma dobro ločene; vsak DTD lahko vsebuje natanko en osnovni nabor znakov. TEI definira osnovne nabore za prozo, poezijo, gledališče, zapis govora, tiskane slovarje ter terminološke baze. Dodatni nabori oznak (additional tag sets) zajemajo raznovrstna dodatna označevanja, ki predstavljajo določeno interpretacijo besedila ali pa nebesedilne elemente besedil, kot so navzkrižne povezave (za stvarna kazala) ali pa slike. Takih naborov je vsega skupaj devet, med njimi so nabor za analitične mehanizme, npr. skladenjsko analizo, nabor za dokumentiranje uredniških posegov, nabor za imena in datume in končno tudi nabor za jezikovne korpuse. Tako lahko v TEI definiramo uporabniško določene oznake (user defined tagset), kjer dodamo lastne oznake ali spremenimo oznake, ki jih definira TEI. Kot primer je v Sliki 3 podana parametrizacija TEI, ki jo uporabljamo za korpus MULTEXT-East; ta definira, da uprabljamo XML različico TEI P4 z osnovnim naborom za prozo ter dodatnimi nabori za povezovanje, jezikoslovne analize, lastnostne strukture in korpuse.

```
<!DOCTYPE TEI.2 SYSTEM "http://www.tei-c.org/Guidelines/DTD/tei2.dtd"
 [<!ENTITY  %  TEI.XML            "INCLUDE">
  <!ENTITY  %  TEI.prose          "INCLUDE">
  <!ENTITY  %  TEI.linking        "INCLUDE">
  <!ENTITY  %  TEI.analysis       "INCLUDE">
  <!ENTITY  %  TEI.fs             "INCLUDE">
  <!ENTITY  %  TEI.corpus         "INCLUDE">
 ]>
```

Slika 3: *Primer TEI DTD*

2.5 Standard za zapis korpusov CES

TEI podaja med drugim tudi določila za označevanje korpusov, ki smo jih tudi sami uporabljali pri zapisu večine naših korpusov. Na osnovi teh priporočil je bil izdelan DTD in spremna dokumentacija, namenjena eksplicitno za zapis korpusov, ki so namenjeni razvoju jezikovnih tehnologij. Predlog, imenovan CES <http://www.cs.vassar.edu/CES/> (Corpus Encoding Standard), je nastal v okviru iniciative Evropske unije Eagles in projektov MULTEXT ter MULTEXT-East. CES je osnovan na SGML, novejša različica z imenom XCES pa na XML.

CES tudi določa osnovni zapis in obseg označevanja, ki ga mora korpus zadovoljiti, da ga še lahko pojmujemo kot standardiziranega. CES opredeli tri nivoje take standardizacije, kjer vsak višji nivo dodatno standardizira korpus.

CES je precej enostavnejši od TEI, saj se ukvarja samo s korpusi, vendar pa ima slabost, da v nekaterih podrobnostih ni skladen s TEI, predvsem pa nima vgrajenih mehanizmov za parametrizacijo, razen seveda z neposrednim spreminjanjem DTD-ja.

3 Ravni označevanja korpusov

V tem poglavju se osredotočimo na bolj pomembne, tj. uporabne vrste oznak, ki se lahko pojavijo v korpusu, opišemo s kakšnimi programi lahko takšne oznake dodajamo za slovenski jezik in kako izgledajo v priporočilih TEI.

3.1 Glava TEI

Vsak dokument TEI in s tem vsak korpus oz. korpusna komponenta morajo biti dokumentirani; element, ki vsebuje metapodatke o dokumentu, je <teiHeader>. Glava TEI je sestavljena iz štirih elementov, od katerih je samo prvi obvezen.

Opis datoteke <fileDesc> vsebuje poln bibliografski opis datoteke, vključno z bibliografsko informacijo njenega vira ali virov. Opis zapisa <encodingDesc> opisuje odnos med elektronskim besedilom in njegovim virom oz. viri. Omogoča podroben opis tega, kako (in če) je bilo besedilo normalizirano pri transkripciji, kako je označevalec razrešil dvoumnosti v viru, katere ravni analize so bile izvedene itd. Opis profila <profileDesc> vsebuje klasifikacijske in kontekstualne podatke o besedilu, npr. tematiko, osebe, ki jih opisuje oz. so sodelovale pri njegovem nastajanju itd. Ta opis je še posebej koristen pri strukturiranih sestavljenih besedilih, kot so korpusi, kjer je dostikrat zaželeno, da vzpostavimo kontroliran opisni besednjak, prek katerega lahko zajemamo besedila glede na zvrst ali izvor. Opis sprememb <revisionDesc> omogoča označevalcem, da zabeležijo zgodovino sprememb, ki so se vršile nad elektronskih besedilom med njegovim nastajanjem; podatek je pomemben za nadzor nad različicami dokumenta (version control) in za razreševanje vprašanj o zgodovini datoteke.

Za ilustracijo informacij, ki jih najdemo v glavi TEI, je v Sliki 4 podan opis vira <sourceDesc>, ki je del opisa datoteke ene od komponent korpusa IJS-ELAN.

```
<sourceDesc>
  <listBibl>
  <bibl lang="sl" default="YES">
    <title>Nacionalni program varstva okolja</title>
    <date value="1999-02-24">24-Feb-99</date>
    <publisher lang="sl">Republika Slovenija, Ministrstvo za okolje in
       prostor, Uprava RS za varstvo narave</publisher>
    <xptr url="http://www.gov.si/mop/vsebina/npvo.html"/>
  </bibl>
  <bibl lang="en" default="NO">
    <title>National Environmental Protection Programme</title>
    <date value="1999-03-10">30-Mar-99</date>
    <publisher>Republic of Slovenia, Ministry of the Environment and
       Physical Planning</publisher>
    <xptr url="http://www.gov.si/mop/vsebina/angl/okolje.html"/>
  </bibl>
 </listBibl>
</sourceDesc>
```

Slika 4: *Primer glave TEI*

3.2 Struktura korpusa

Tipično imajo dokumenti TEI kot korenski element <TEI.2>; ta vsebuje glavo TEI in samo besedilo <text>. Izjema v tej strukturi so ravno korpusi, saj imajo kot vrhnji element <teiCorpus.2>, ta pa nato vsebuje posamezne elemente korpusa, kot je ilustrirano v Sliki 5.

```
<TEIcorpus.2>
   <teiHeader type="corpus">
      *Glava korpusa*
   </teiHeader>
   <TEI.2>
      <teiHeader type="text">
        *Glava elementa*
      </teiHeader>
      <text>
        *Besedilo elementa*
      </text>
   </TEI.2>
  *Ostale komponente*
</TEIcorpus.2>
```

Slika 5: *Struktura korpusa TEI*

Besedilo je nato sestavljeno iz začetnega dela <front>, telesa <body> in končnega dela <back>, pri čemer prvi in zadnji nista obvezna. Besedilo je razdeljeno na razdelke , ki se lahko tudi gnezdijo, ti pa naprej iz odstavkov. V osnovnem naboru oznak za leposlovje imamo nato na voljo množico oznak, ki opisujejo strukturo besedila, kot so npr. glava <head>, poudarjeno <hi>, opomba <note>, povezava <ref> itd.

Pri pretvorbi iz izvirnega (elektronskega) zapisa se dostikrat zastavi vprašanje, koliko in katere oznake naj se ohranijo v korpusu; tipično ohranimo samo tiste, ki jih je enostavno avtomatsko prepoznati (npr. odstavek), ostale pa izpustimo, saj je pravilna konverzija iz raznovrstnih in slabo strukturiranih dokumentov težavna, za namene korpusnega jezikoslovja pa dostikrat tudi nepotrebna.

3.3 Besede in stavki

Čeprav so oznake v korpusu v veliki meri odvisne od namembnosti, pa sta prva koraka, ki sta vedno koristna, označitev besed in stavkov v besedilu, t. i. tokenizacija in segmentacija, saj z njima damo jezikovni vsebini osnovno strukturo. To na prvi pogled preprosto označevanje v sebi skriva kar nekaj pasti (Grefenstette in Tapanainen 1994), saj npr. pika ne označuje vedno konca stavka, npr. *Dr. Prešeren*, pa tudi besede imajo lahko kompleksno strukturo, posebej v tehničnih besedilih, npr. *B3.2 14,40+/-1,92*. Dodatna ovira kvalitetnemu razčlenjevanju so tudi napake in nekonsistentnosti v besedilih. Zato se še vedno dela na izdelavi kvalitetnih tokenizatorjev, ki za tipično delovanje potrebujejo jezikovno odvisna pravila in sezname, npr. okrajšav; sami smo jih razvijali za slovenski jezik v okviru projekta MULTEXT-East za orodja MULTEXT (Di Cristo 1996).

V TEI so besede, ločila in stavki del dodatnega nabora znakov za jezikoslovno analizo, TEI.analysis; primer tako označenega besedila iz korpusa IJS-ELAN je podan v Sliki 6.

```
<seg id="ecmr.sl.17">
<w>Tokrat</w> <w>v</w> <w>obliki</w> <w>poslab&scaron;anja</w>
<w>Euromoneyeve</w> <w>ocene</w> <w>ekonomskih</w><w>sprememb</w>
<w>v</w> <w>Sloveniji</w> <c type="open"></c><w>stran</w> <w
type="dig">6</w><c type="close"></c><c>.</c>
</seg>
```

Slika 6: *Primer besednih oznak TEI.analysis*

3.4 Oblikoslovno označevanje

Naslednja stopnja označevanja je pripis oblikoslovnih značilnosti besedam. Takšno označevanje je bilo najprej razvito za angleški jezik (part-of-speech tagging), kjer je veliko besed dvoumnih glede na besedno vrsto, naloga označevalnika pa je besedi glede na sobesedilo določiti pravilno besedno vrsto. V slovenščini je sicer takšnih dvoumnosti manj (pa še vedno precej), zato pa jih je toliko več v pregibnih lastnostih besed, predvsem številu in sklonu. Zaradi bistveno bogatejšega pregibanja je tudi nabor oblikoslovnih oznak za slovanske jezike bistveno večji kot za večino zahodnoevropskih jezikov; če je za angleščino oznak tipično okoli petdeset, jih je v slovenščini prek tisoč.

Ker je določanje oblikoslovnih lastnosti besed pomemben korak za nadaljnje obdelave, je v zadnjem desetletju postalo izjemno aktivno področje raziskav (van

Halteren 1999). Označevanje tipično poteka v dveh fazah. Za prvi korak potrebujemo slovar (ponavadi skupaj z oblikoslovnim analizatorjem), ki nam za vsako besedo vrne vse njene možne oblikoslovne oznake, npr. za *berači*, da je glagol v velelniku ali povedniku ali samostalnik v imenovalniku ali orodniku. Do te faze je npr. trenutno označen korpus FIDA. Drugi korak, ki predstavlja bistvo označevanja, pa nato določi pravilno oznako besede glede na njeno funkcijo v besedilu; dodatno funkcionalnost predstavlja določanje oznak tudi neznanim besedam, torej tistim, ki jih ni v slovarju.

Čeprav je pravila za razdvoumljanje mogoče pisati tudi ročno, je to izredno kompleksno in zamudno delo, zato je prevladujoča paradigma ta, da se označevalniki avtomatično naučijo statističnega modela jezika iz korpusa, ki je predhodno ročno označen. Ena bolj odmevnih metod uporablja t. i. skrite markovske verige, s katerimi določamo najbolj verjetno zaporedje oblikoslovnih oznak besed v stavku glede sosednje oznake.

Za slovenski jezik je največja ovira za kvalitetno označevanje oblikoslovnih lastnosti pomanjkanje velikega ročno označenega korpusa, ki bi lahko služil kot učna množica. Korpus MULTEXT-East (Erjavec 2001) je sicer javno dostopen v raziskovalne namene <http://nl.ijs.si/ME/>, je pa njegov označeni del majhen (100.000 besed) in vsebuje samo roman 1984. Vseeno pa smo na tem korpusu ovrednotili večje število dostopnih označevalnikov (Džeroski et al. 2000); najboljše rezultate je dosegel program TnT (Brants 2000), z natančnostjo 92 %. Ta označevalnik z naučenim modelom skupaj s še nekaj izboljšavami smo nato uporabili za avtomatsko označevanje slovenskega dela korpusa IJS-ELAN (Erjavec 2002b), ki je tudi prosto dostopen <http://nl.ijs.si/elan/>.

V TEI so oblikoslovne oznake tipično zapisane v atributu analize besed iz nabora znakov TEI.analysis; primer označenega stavka iz korpusa IJS-ELAN je podan v Sliki 7.

```
<seg id="kmet.sl.230" corresp="kmet.en.230">
<w ana="Ncmsn" lemma="prikaz">Prikaz</w>
<w ana="Afpmpg" lemma="odgovoren">odgovornih</w>
<w ana="Ncmpg" lemma="del">delov</w>
<w ana="Ncnsg" lemma="ministrstvo">ministrstva</w>
<w ana="Spsa" lemma="za">za</w>
<w ana="Ncnsa" lemma="izvajanje">izvajanje</w>
<w ana="Rgp" lemma="zgoraj">zgoraj</w>
<w ana="Afpfpg" lemma="naveden">navedenih</w>
<w ana="Ncfpg" lemma="naloga">nalog</w>
<c ctag=":">:</c>
</seg>
```

Slika 7: *Primer oblikoslovno označenega besedila*

3.5 Imena, besedne zveze in termini

Identificiranje imen (named entity extraction) se ima za svojo popularnost v veliki meri zahvaliti tekmovanjem, ki so od poznih 80. spremljala konference MUC (Message Understanding Conferences, financirane z ameriške DARPA), kjer je bila naloga s čim večjo točnostjo identificirati in kategorizirati zemljepisna imena, imena ljudi in podjetij, časovne in denarne izraze ipd. Takšna imena so mogoče bolj kot za klasično korpusno jezikoslovje pomembna predvsem za zajemanje informacij (Information Extraction), saj predstavljajo kritični prvi korak pri delovanju takšnih sistemov.

Za slovenski jezik na tem področju še ni bilo kaj dosti narejenega; še najbolj se tej problematiki približa Jakopin (2000), ki poskuša identificirati elektronske in spletne naslove v časopisu DELO.

Imena se v TEI označujejo z elementom <name>, ki je del središčnih oznak; le-te vsebujejo tudi vrsto sorodnih elementov, npr. <date>, <number> itd. Za podrobna označevanja imen pa ponuja TEI tudi dodaten nabor oznak TEI.names.dates; ta vsebuje podrobne elemente za imena in druge besedne zveze, ki opisujejo osebe, kraje, organizacije, pa tudi datume in čase.

Do neke mere podobna naloga kot identificiranje imen je identificiranje besednih zvez (phrase chunking); tu je naloga razdeliti stavek na neprekrivajoče se besedne zveze, tipično samostalniške. Označevanje besednih zvez torej predstavlja nekakšno plitko skladenjsko analizo, vendar je še vedno precej zahteven korak, predvsem zaradi kompleksnosti razreševanja dvoumnosti vezave posameznih konstituentov. Ti sistemi ponavadi pričakujejo besedilo, ki je že bilo oblikoslovno označeno; nekateri od njih nato uporabljajo ročno napisana pravila, večina pa se modela nauči iz vnaprej označenih besedil, kjer uporabljajo podobne (ali celo identične) metode kot pri oblikoslovnem označevanju.

Identificiranje predvsem samostalniških zvez ima več aplikacij (Abney 1991): uporablja se lahko, podobno kot oblikoslovno označevanje, kot predstopnja za polno skladenjsko analizo, saj že razreši del dvoumnosti v stavku in tako razbremeni skladenjski modul. Uporabno je tudi, podobno kot identifikacija imen, v sistemih za zajemanje informacij pa tudi v večjezičnih korpusih (oz. sistemih za ekstrakcijo leksikonov in avtomatsko prevajanje), saj se prevodne ustreznice dostikrat najde šele na ravni besedne zveze in ne besede.

Elemente za označevanje besednih zvez najdemo v modulu TEI.analysis; tu lahko izberemo med generičnim <seg>, ki ga po možnosti opremimo z atributom *type*, lahko pa uprabimo tudi namenski element <phr>.

Kot posebno vrsto besednih zvez lahko pojmujemo termine; ti so izrazito pomembni v strokovnem jeziku, definirani pa niso samo strukturno, temveč tudi pomensko oz. leksikalno, saj mora neka beseda ali besedna zveza biti specifična in pomembna za neko strokovno področje, da jo lahko poimenujemo termin. Obstaja vrsta metod, kako identificirati termine, in večina spet temelji na statističnih lastnostih besedila v korpusu. Ena bolj enostavnih primerja frekvenco pojavitev besed iz

splošnega korpusa s frekvencami pojavitev iz konkretnega besedila: besede ali besedne zveze, ki imajo v nekem besedilu precej višjo frekvenco od povprečja, so verjetno termini. Pri korpusih, ki so oblikoslovno označeni, je možno tudi izkoriščati vzorce z besednimi vrstami, ki omejijo najdene kolokacije. Identifikacija terminov, podobno kot samostalniških zvez, je še posebej zanimiva v kontekstu večjezičnih korpusov, saj odpira vrata za avtomatizirano izdelavo večjezičnih terminoloških slovarjev in pomagal. Pregled metod za luščenje terminov najdemo v Vintar (2002), ker je tudi opis razvoja in uporabe dveh metod za slovenski jezik, na primeru slovensko-angleškega korpusa TRANS.

3.6 Povezave

Do sedaj obravnavana označevanja imajo skupno lasnost, da kategorizirajo nek del besedila. Drug tip označevanja pa je povezovanje besedila z nekim drugim besedilom, bodisi da gre za prevod ali pa za variantno izdajo.

Vzporedni večjezični korpusi so tipični primer, ko je smiselno vzpostaviti povezave (vsaj) med stavki originala in prevodov. Obstaja več avtomatskih načinov, kako poravnati stavke; verjetno najbolj znan je statistični program, ki enostavno primerja dolžine stavkov (Gale in Church 1993); tega smo uporabili tudi mi pri poravnavi korpusov MULTEXT-East in (deloma) IJS-ELAN.

Poravnave se tipično izraža s pomočjo povezav, tj. atributov, ki kažejo na poravnane elemente. Primer tega lahko vidimo v Sliki 8, kjer atribut *corresp* identificira poravnani segment angleškega prevoda. Bolj splošen način ponuja CES s posrednim označevanjem, kjer so poravnave shranjene v posebnem dokumentu, ki samo kaže na besedilo. Slika 8 nam poda primer takšnega označevanja iz korpusa MULTEXT-East, kjer je prva poravnava 1-1, druga 2-1 in tretja 1-0.

```
<link xtargets="Osl.1.1 ; Oen.1.1">
<link xtargets="Osl.1.2 Osl.1.3 ; Oenl.1.2">
<link xtargets="Osl.1.4 ; ">
```

Slika 8: *Primer povezav CES*

Povezovanje se uporablja tudi pri enojezičnih korpusih oz. elektronskih izdajah, bodisi za povezave znotraj dokumenta, predvsem pa, kadar nas zanimajo različne izdaje istega dela. Tu velja omeniti dva modula TEI. TEI.transcr obravnava transkripcijo primarnih virov in je namenjen predvsem označevanju rokopisov, medtem ko modul za označevanje kritičnega aparata TEI.crit definira strukturo in oznake za pripombe in variantne poglede na besedilo. Ti moduli omogočajo označevanje in povezovanje več »branj« besedila (<rdg> reading) in definirajo tudi elemente za opis grafične podobe besedila.

4 Zaključki

V članku smo podali pregled standardov in ravni označevanja, ki se jih uporablja oziroma bi se jih lahko uporabljalo pri zapisu korpusov slovenskega jezika. Predlagane metode imajo prednost, da spodbujajo izmenjavo, trajnost in večnamensko uporabo korpusov.

XML ima toliko očitnih prednosti, da lahko v prihodnosti pričakujemo skoraj univerzalno uporabo tega zapisa za potrebe korpusov in drugih jezikovnih virov. Tako se XML npr. že uporablja v slovenskem slovaropisju: Krek (2003) opiše XML DTD, ki so ga naredili za slovenske dvojezične slovarje.

Navzočnost TEI v zapisu jezikovnih virov je bolj vprašljiva, saj imajo lastni DTD-ji prednost, da so tipično enostavnejši in bolj perskriptivni, kar olajša uporabo, lažje pa jih je tudi posloveniti. Ravno zaradi enostavosti je postal popularen tudi CES, ki je sicer neprimerno manj fleksibilen kot TEI. Vendar pa je večino kompleksnosti TEI možno skriti pred uporabnikom, pa tudi snovalcem in uporabnikom DTD-ja; spletni vmesnik TEI Pizza Chef <http://www.hcu.ox.ac.uk/TEI/pizza.html>, omogoča interaktivno definicijo definicije tipa dokumentov, izražene v eni datoteki.

Močan argument proti uporabi TEI bi lahko bila tudi anglocentričnost pobude: čeprav je besedilo v poljubnem jeziku, pa so imena vseh oznak angleška. In čeprav TEI P4 omogoča lokalizacijo imen elementov skozi uporabniško določene entitete, ne podpira prevajanja imen atributov in njihovih vrednosti. V zadnjem času se je tudi tu zgodil premik, saj je v teku pobuda (Bia et al. 2003) za izdelavo standardiziranih knjižnic večjezičnih imen vseh TEI elementov, atributov in njihovih vrednosti, s katerimi je možno oznake dokumentov TEI avtomatično z uporabo XSLT prevajati med jeziki.

Članek je podal tudi pregled ravni jezikovnega označevanja korpusov, pri tem pa izpostavil predvsem videnje označevanja korpusov, kjer le-te naprej zajamemo, nato pa tokeniziramo, segmentiramo, oblikoslovno označimo in pri vzporednih korpusih še čimbolje povežemo.

To je približno tudi domet jezikovnih tehnologij za slovenski jezik, saj te ravni z večjo ali manjšo natančnostjo še zmoremo avtomatsko označiti. Čeprav je tudi tu še polno odprtih vprašanj in možnosti za izboljšanje tehnologij, pa obstajajo tudi jezikoslovne ravni, ki se jih za slovenščino sploh nismo dotaknili. Tu velja izpostaviti predvsem skladenjsko označevanje, kjer je vsak stavek v korpusu označen s svojo strukturno ali funkcijsko analizo. Korpusi skladnje (treebanks) obstajajo za večje število jezikov, začenši z angleškim, kjer je bil izredno vpliven Penn Treebank <http://www.cis.upenn.edu/~treebank/>, ki vsebuje milijon besed, označenih s frazними strukturami. Za slovenski jezik je bolj zanimiv Prague Dependecy Treebank <http://ckl.mff. cuni.cz/>, ki vsebuje milijon besed, označenih z odvisnostnimi analizami. Čeprav je del PDT označen tudi z globinskimi funkcijskimi strukturami (tectogramatic level), pa se na prvi stopnji analitične strukture vsaki besedi oz. ločilu pripiše odvisnost od nadrejene besede.

Literatura

Abney, Steven, 1991: Parsing by Chunks. Berwick, Robert C., Abney, Steven P., in Carol Tenny (ur.): *Principle-Based Parsing: Computation and Psycholinguistics*, Dodrecht: Kluwer Academic Publishers. 257–278.

Bia, Alejandro, Sanchez-Quero, Manuel, Regis Deau, 2003: Multilingual Markup of Digital Library Texts Using XML, TEI and XSLT. *XML Europe 2003*. Alexandria, ZDA: IDEAlliance. 53–58.

Brants, Thorsten, 2000: TnT – A Statistical Part-of-Speech Tagger. *Sixth Applied Natural Language Processing Conference ANLP-2000.* Seattle, WA: ACL. 224–231.

Di Cristo, Philippe, 1996: MtSeg: *The MULTEXT Multilingual Segmenter Tools*. MULTEXT Deliverable MSG 1, Version 1.3.1, CNRS, Aix-en-Provence.

Džeroski, Sašo, Erjavec, Tomaž, Zavrel, Jakub, 2000: Morphosyntactic Tagging of Slovene: Evaluating PoS Taggers and Tagsets. *Second International Conference on Language Resources and Evaluation, LREC'00*. Paris: ELRA. 1099–1104.

Erjavec, Tomaž, Gorjanc, Vojko, Stabej, Marko, 1998: Korpus FIDA. *Konferenca Jezikovne tehnologije za slovenski jezik.* Ljubljana: Institut Jožef Stefan. 124–127.

Erjavec, Tomaž, 2001: Harmonised Morphosyntactic Tagging for Seven Languages and Orwell's 1984. *6th Natural Language Processing Pacific Rim Symposium, NLPRS'01.* Tokyo. 487–492.

Erjavec, Tomaž, 2002a: Compilation and Exploitation of the IJS-ELAN Parallel Corpus. *Zbornik B 5. Informacijska družba IS'2002: jezikovne tehnologije.* Ljubljana: Institut Jožef Stefan. 86–93.

Erjavec, Tomaž, 2002b. The IJS-ELAN Slovene-English Parallel Corpus. *International Journal of Corpus Linguistics* 7/1. 1–20.

Gale, William, 1993: A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, 19/1. 75–102.

Grefenstette, Greg, Tapanainen, Pasi, 1994: What is a Word, What is a Sentence? Problems of Tokenization. *The 3rd International Conference on Computational Lexicography, COMPLEX '94.* Budapest: Research Institute for Linguistics, Hungarian Academy of Sciences. 79–87.

Jakopin, Primož, Bizjak, Aleksandra, 1997: O strojno podprtem oblikoslovnem označevanju slovenskega besedila. *Slavistična Revija,* 45/3–4. 513–532.

Jakopin, Primož, 2000: Elektronski in spletni naslovi v časopisu DELO, 1998–2000. *Konferenca Jezikovne tehnologije, Informacijska družba, IS'2000.* Ljubljana: Institut Jožef Stefan. 116–119.

Krek, Simon, 2003: Sodobna dvojezična leksikografija. *Jezik in Slovstvo*, 48/1. 45–60.

Sperberg-McQueen, C, M., Burnard, Lou, 2002: *Guidelines for Electronic Text Encoding and Interchange, The XML Version*. The TEI Consortium.

van Halteren, Hans, 1999: *Syntactic Wordclass Tagging*. Dodrecht: Kluwer Academic Publishers.

Vintar, Špela, 2002: Avtomatsko luščenje izrazja iz slovensko-angleških vzporednih besedil. *Zbornik B 5. Informacijska družba IS'2002: jezikovne tehnologije.* Ljubljana: Institut Jožef Stefan. 78–85.

Seznam pomembnejših spletnih naslovov

http://www.w3.org/XML/
XML: eXtended Markup Language
http://www.tei-c.org/
TEI: Text Encoding Initiative
http://www.cs.vassar.edu/CES/
CES: Corpus Encoding Standard
http://www.telri.de/
Iniciativa TELRI, »Trans European Language Resources Infrastructure«
http://tractor.bham.ac.uk/
Mre ni arhiv TRACTOR: TELRI Research Archive of Computational Tools and Resources
http://nl.ijs.si/ME/
Korpusi in leksikoni projekta MULTEXT-East, »Multilingual TextTools and Corpora for Central and Eastern European Languages«
http://nl.ijs.si/elan/
Slovensko-angleški korpus IJS-ELAN
http://nl2.ijs.si/corpus/
Mre ni konkordančnik IJS
http://bos.zrc-sazu.si/
Korpus Nova beseda
http://www.fida.net/
Korpus FIDA
http://www.ijs.si/lit/leposl.html
Korpus SlovLit