Ročníkový projekt – špecifikácia

Názov: Prevod vnútorných formátov pražských korpusov

Autor: Michal Vachna
Vedúci projektu: Barbora Vidová Hladká

Programovací jazyk: C++

1. Motivácia

Na prevod medzi formátmi CSTS a PML sa používá procedúra, ktorá je realizována ako perlovské makro v prostredí anotačného editoru [b]TrEd. Pre niektoré aplikácie, ktorých súčasťou je prevod spomenutých formátov, je nutnosť nainštalovania TrEdu "na obtiaž". Prevodná procedúra, ktorá nevyžaduje "tretie" prostredie, by bola vhodnejšia.

2. Zadanie

Implementácia automatickej procedúry na prevod medzi formátmi, ktoré sa používajú pre vnútornú reprezentáciu tzv. pražských korpusov (banky textov obohatenej o jazykovedné informácie). Konkrétne se jedná o formát založený na XML, tzv. PML a o formát založený na SGML, tzv. CSTS.

3. Úvod do problematiky

Formát PML

Formát *Prague Markup Language* (PML) je založený na XML a navrhnutý pre bohatú reprezentáciu lingvistickej anotácie textov. Každá rovina anotácie je popísaná v soubore *PML schéma*, ktorý je akousi formalizáciou abstraktnej anotačnej schémy pre tú ktorú rovinu anotácie. Rozlišujú sa tri roviny anotácie: ***w***, ***m***, ***a*** . Rovina ***w*** obsahuje slovné jednotky. Rovina ***m*** obsahuje morfologickú anotáciu a hranice viet. Rovina ***a*** anotáciu stromov závislosti.

Každý PML súbor začína hlavičkou odkazujúcou na PML schéma súboru. V hlavičke sú uvedené všetky externé zdroje. Zbytok súboru obsahuje vlastnú anotáciu.

Anotácia je vyjádrená pomocou XML elementov a atributov pomenovaných a použitých v súlade s príslušnou PML schémou. XML elementy všetkých súborov patria do vyhradeného menného priestoru http://ufal.mff.cuni.cz/pdt/pml/.

Stručný popis rovín

W-rovina dat pozostáva s elementov ***head***, ***meta*** a ***doc***. Element ***doc*** pozostáva z hlavičky (element ***docmeta***) a odstavcov (element ***para***). Element ***para*** pozostáva z postupnosti elementov ***w*** obsahujúcich mimo iné dva povinné elementy: ***id*** a ***token***.

M-rovina dat pozostáva s elementov ***head***, ***meta*** a ***s***. Element ***s*** (veta) pozostáva z elementov ***m*** (slovné jednotky).

A-rovina dat pozostáva s elementov ***head***, ***meta*** a ***trees***. Element ***trees*** popisuje závislostnú štruktúru vety štruktúrou vnorovaných elementov. Synovské uzly sú obalené elementom ***children***. Každý uzol je ďalej obalený elementem ***LM***, ktorého atributom je identifikátor tohoto uzlu. Elementy ***s.rf*** a ***m.rf*** odkazujú na príslušný prvok nižšej roviny, ktorý obsahuje konkrétnu slovnú formu. Element ***ord*** obsahuje poradie uzlu v stromu zľava doprava, ktoré je zhodné s poradím slova vo vete.

Formát CSTS

Formát je založený na formáte SGML. Je lepšie čitateľný pre človeka a je možné ho spracovať jednoduchými nástrojmi. Vo formáte CSTS sú všetky roviny anotácie uchované v jednom súbore.

Súbor CSTS začíná (nepovinnou) hlavičkou (element ***h***) a ďalej obsahuje aspoň jeden element ***doc***. Element ***doc*** pozostáva z hlavičky (element ***a***) a obsahu (element ***c***). Element ***c*** potom pozostáva z postupnosti odstavcov (element ***p***) a viet v týchto odstavcoch (element ***s***).

Každá slovná jednotka vety je na samostatnom riadku súboru (element ***f***, resp. ***d*** pre interpunkciu), ďalej na tomto riadku nasleduje anotácia tejto slovnej jednotky na všetkých rovinách.

3.1. Princíp prevodu formátov

Najzložitejšou časťou prevodu je previesť slovnú jednotku z jednej reprezentácie do druhej a naopak. Ostatné časti prevodu sú viac-menej triviálne.

Pri prevode z CSTS je nutné načítať a rozparsovať jednotlivé slovné jednotky. Následne z týchto dat vytvárať postupne súbory w-roviny, m-roviny a a-roviny v PML.

Pri prevode z PML je tiež nutné načítať a naparsovať jednotlivé slovné jednotky do troch previazaných datových štruktúr (pre každú rovinu jedna), uspôsobené

na rýchle prehľadávanie a prístup. Kedže každá slovná jednotka je jednoznačne identifikovaná v rámci roviny pomocou id, môže tento údaj slúžiť aj na vyhľadávanie a odkazovanie. Vytváranie súborov v CSTS by spočívalo v skladaní údajov o každej slovnej jednotke prechodom od najvyššej vrstvy k najnižšej.

3.2. Algoritmus prevodu formátov

Ide v podstate o zhrnutie predchádzajúcej časti Princíp prevodu formátov.

Program po tom, čo spracuje vstupné parametre a prejde do jedného z dvoch módov prevodu (CSTStoPML, PMLtoCSTS), bude pokračovat v parsovaní vstupných súborov na nasledovnom princípe. Ak prečíta kľúčove slovo (názvy elementov, tagov), predá riadenie obslužnej funkcií, ktorá spracuje potrebnú časť vstupu a vráti riadenie. Načítané a spracované data sa budú ukladať do pripravených datových štruktúr.

Po spracovaní vstupu sa datové štruktúry predajú danej prevodnej funkcii (CSTStoPML, PMLtoCSTS).

CSTStoPML

V súbore v CSTS sa pracuje s elementami ***a*** a ***c*** elementu ***doc***. Element ***a*** sa prevedie na data v ***othermeta*** v súbore w-roviny. Z elementu ***c*** sa elementy ***p***, ***s***, ***f*** a ***d*** prevedú na data elementov do príslušných súborov rovín.

PMLtoCSTS

Zo súborov w-roviny v PML sa elementy ***meta*** a ***docmeta*** použijú na vytvorenie elementu ***a*** v elemente ***doc***. Element ***p*** sa vytvorí z elementov ***para*** súborov w-roviny, element ***s*** z elementu ***s*** súboru m-roviny a elementy ***f*** a ***d*** zložením elementov ***w***, ***m*** a ***LM*** jednotlivých rovín.

4. Implementácia

Algoritmy použité v programe sú spomenuté v časti Úvod do problematiky.

4.1. Návrh štruktúry programu:

Program by sa mal skladať minimálne zo 4 modulov:

1. Logika programu - mal by obsluhovať ovládanie programu (načítanie parametrov z príkazovej riadky, ich overovanie a zpracovanie, IO

funkce). Program tohoto typu nepotrebuje GUI a stačí teda načítanie parametrov z príkazovej riadky.
2. Modul obsahujúci parsovací algoritmus – všetko potrebné pre parsovanie (algoritmus, vytváranie príslušných datových štruktúr).
3. Modul obsahujúci prevodný algoritmus – všetko potrebné pre prevod (algoritmus, prístup k datovým štruktúram).
4. Modul obsahujúci triedy datových štruktúr.

4.2. Datové štruktúry

V oboch formatoch je potrebné uchovávať data o slovných jednotkách a tiež úvodné hlavičky.

Súbor v CSTS bude reprezentovaný a uložený ako stromová štruktúra tried. Koreňovou triedou bude trieda ***csts.*** Jej synovské uzly budú triedy ***h*** a ***doc.*** Tieto triedy budú mať ďalej svoje synovské uzly. Tým bude pokrytá celá štruktúra CSTS. Triedy ***s***, ***f*** a ***d*** bude vhodné udržiavať ako usporiadané zoznamy podľa ***id*** v rodičovskom uzle. Tým sa dosiahne jednoduchý prístup a práca s vetami a slovnými jednotkami.

Súbor v PML bude (podobne ako v CSTS) reprezentovaný a uložený ako stromová štruktúra tried s koreňovými triedami ***wdata***, ***mdata*** a ***adata.*** Triedy, ktoré je vhodné udržiavať ako usporiadaný zoznam podľa ***id*** sú vo w-rovine trieda ***w***, v m-rovine trieda ***s*** a ***m***, v a-rovine trieda ***LM***.

5. Literatúra

Nástroj TrEd

- domovská stránka - http://ufal.mff.cuni.cz/~pajas/tred/index.html

Formát PML

- stručný popis – http://ufal.mff.cuni.cz/rest/CAC/doc-cac20/cac-guide/cz/html/ch3.html#data-format
- domovská stránka - http://ufal.mff.cuni.cz/jazz/pml/

Formát CSTS

- stručný popis – http://ufal.mff.cuni.cz/rest/CAC/doc-cac20/cac-guide/cz/html/ch3.html#data-format
- dokumentácia - http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/cz/html/ch03.html#a-data-formats-csts