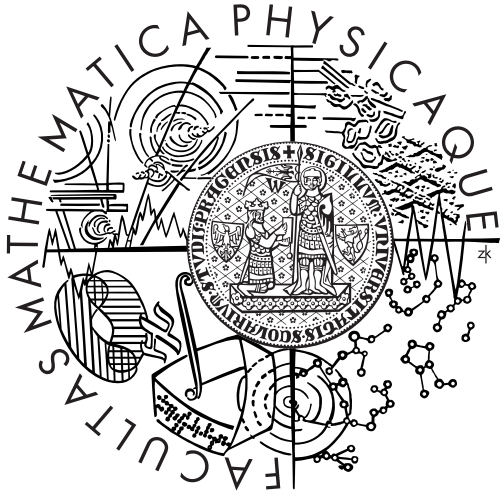
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Jan Dědek

Sémantická anotace dat z webovských zdrojů

Katedra softwarového inženýrství

Vedoucí diplomové práce: Prof. RNDr. Peter Vojtáš, DrSc.
Studijní program: Informatika, I2 - Softwarové systémy

2007

Chtěl bych poděkovat vedoucímu Prof. RNDr. Peterovi Vojtášovi, DrSc. za motivující a inspirující vedení a kontrolu průběhu vzniku diplomové práce. Dále děkuji Ing. Zdeněku Žabokrtskému, Ph.D., RNDr. Václavu Klimešovi, Ph.D., doc. PhDr. Karelu Palovi, CSc za poskytnutí softwaru a přínosné konzultace.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 10. 8. 2007 Jan Dědek

Název práce: Sémantická anotace dat z webovských zdrojů
Autor: Jan Dědek
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí diplomové práce: Prof. RNDr. Peter Vojtáš, DrSc.
e-mail vedoucího: Peter.Vojtas@mff.cuni.cz

Abstrakt: Tato práce se odráží od myšlenky sémantického webu. Stručně rozebírá možnosti formální reprezentace znalostí v deskripční logice a její paralelu v několika formalismech pro tvorbu ontologií. Ukazuje, jak lze využít ontologií při sémantické anotaci webovských zdrojů. Představuje sémantickou anotaci v praxi, v kontextu několika projektů z různých oblastí. V práci jsou rozebrány různé metody extrakce informací, které pomáhají sémantickou anotaci zautomatizovat. Podrobněji jsou v tomto ohledu popsány nástroje, které poskytuje současná česká počítačová lingvistika. Na teoretické úrovni se tato práce dotýká vztahu mezi lingvistickou anotací přirozeného jazyka a formální reprezentací znalostí v deskripční logice. V rámci této práce byl proveden experiment – zpracování českého přirozeného textu několika lingvistickými nástroji za účelem jeho sémantické anotace.

Klíčová slova: sémantický web, sémantická anotace, ontologie, zpracování přirozeného jazyka, extrakce informací

Title: Semantic annotation of data from web resources
Author: Jan Dědek
Department: Department of software engineering
Supervisor: Prof. RNDr. Peter Vojtáš, DrSc.
Supervisor's e-mail address: Peter.Vojtas@mff.cuni.cz

Abstract: This work starts with the idea of The Semantic Web. Then basic description logics is introduced with its parallel in a couple of formalisms for building of ontology. In this work, there is shown, how the ontologies are employed in the semantic annotation process and also there are described some projects that use semantic annotation in a practical way. Information extraction methods that help to automatize the semantic annotation process are mentioned. Tools for natural language processing of Czech language are described in detail. A practical experiment shows how these tools can help with extraction of information from plain text. This work also deals with the relationship of natural language processing and formal representation of knowledge in description logics.

Keywords: semantic web, semantic annotation, ontology, natural language processing, NLP, information extraction

Obsah

Kapitola 1

Úvod

Tato práce se zabývá sémantickou anotací webovských zdrojů. Slovo anotace znamená připojování poznámek či vysvětlivek. V sémantické anotaci, tedy v sémantických poznámkách či vysvětlivkách, se snažíme o zachycení významu (sémantiky) anotovaného zdroje. Tuto sémantiku chceme vyjádřit formálně, tak aby byla strojově (softwarově) zpracovatelná a přispěla tak k vytvoření takzvaného „sémantického webu“, který je podrobněji popsán v oddíle 1.3. Při této anotaci se zaměřujeme na zdroje webu – především web-stránky, nicméně ukážeme, že sémantická anotace má stejný smysl i v jiných oblastech. Příklad takové oblasti je možné najít hned v podkapitole 1.1.2.

Tato práce má několik cílů. Nejprve jsme se pokusili zmapovat oblast sémantické anotace jako takové. Zjistili jsme, že pole sémantické anotace je široké a že na tuto činnost můžeme nahlížet různě. Na nejpovrchnější úrovni zkoumání jsme sémantickou anotaci rozdělili na takzvanou *anotaci autorskou* a *anotaci dodatečnou*. Rozdíl mezi těmito pohledy na anotaci je velký, avšak oba přístupy mají mnoho společného a dá se říci, že se vzájemně doplňují.

Autorskou anotací budeme označovat proces, kdy autor vytváří web-stránku, případně jiný potenciální zdroj informací a tento zdroj obohatí o takzvaný „sémantický popis“. Tedy že formálně popíše informace, které jsou ve zdroji obsažené tak, aby byli využitelné pro strojové zpracování. Můžeme to také chápat tak, že autor, který tvoří web-stránku pro lidské návštěvníky v přirozeném jazyce, ji vytvoří ještě ve verzi pro softwarové návštěvníky ve formálním strojovém jazyce. Může se to zdát náročné, ale není. Strojový jazyk je stručný a přímočarý, nehledí na design ani stylistiku. Navíc autor může vybrat jen některé nejdůležitější informace, které formálně vyjádří. Přínos je potenciálně obrovský.

Dodatečnou anotací budeme označovat proces, kdy web-stránky (pří-

padně i jiné zdroje) už existují a my se je pokoušíme anotovat dodatečně. V tomto případě je většinou autor anotace jiný než majitel původní stránky. Hotové anotace se v původním zdroji neobjeví, nejsou určeny k přímé publikaci (která by se pravděpodobně musela právně ošetřit autorskou smlouvu). Primární cíl této anotace je jiný. Snažíme se ze zdroje získat co nejvíce sémantických dat, která jsou určená pro další zpracování. Například, můžeme chtít vytvořit speciální vyhledávač pracovních nabídek, který shromažďuje data o nabídkách práce z velkého množství různých web-stránek. Právě takový projekt v současné době vzniká na Slovensku pod názvem NAZOU[1]. Dodatečná sémantická anotace se velmi podobá extrakci informací z webu (web data mining, web content mining), největší důraz je zde kladen na automatizaci celého procesu a na robustnost extrakční metody k různým typům web-stránek. Navíc oproti obyčejné extrakci informací se zde snažíme využít pokročilé metody pro formální reprezentaci znalostí.

Poté, co jsme takto sémantickou anotaci rozdělili, zkoumáme obě oblasti víceméně odděleně. Autorskou anotací se zabýváme z hlediska možných standardů a formalismů pro reprezentaci znalostí. Podrobněji rozebíráme některé uznávané ontologie pro popis dat frekventovanějších domén.

Dodatečnou anotaci zkoumáme v kontextu mnoha existujících projektů a metod, které byly pro automatickou extrakci dat z webových zdrojů použity. Seznámili jsme se s množstvím různých přístupů k tomuto problému. Kromě metod strojového učení, je v projektech často využívána počítačová lingvistika.

Cílem praktické části práce byl pokus o realizaci některé automatické metody. Vybrali jsme lingvistický přístup v českém prostředí. Pokusili jsme se využít dostupné lingvistické nástroje pro analýzu českých textů, jmenovitě nástroje PDT (viz oddíl 4.2) pro automatickou lingvistickou anotaci Češtiny a český WordNet (kapitola 5). Jedním z hlavních důvodů pro volbu české lingvistiky byl fakt, že jsme nenarazily na žádnou, práci, ve které by tyto nástroje byly použity pro extrakci informací. Domníváme se, že zkušenosti z našeho experimentu mohou být přínosné nejen pro další sémantickou anotaci českých textů, případně pro extrakci informací z nich, ale i pro samotnou českou lingvistiku.

Hledali jsme vhodnou oblast pro aplikaci lingvistických metod. Jako zdroj anotace jsme nakonec použili data hasičské záchranné služby a databázi úpadců Ministerstva spravedlnosti České republiky. Tyto zdroje jsou popsány v podkapitole 6.2.

Zkušenosti, metodika !!!dopsat!!!

[1] http://nazou.fiit.stuba.sk

1.1 Motivace

1.1.1 Vylepšeme své stránky!

Překládáme nyní krátký motivační text, volně převzatý ze stránek Semantic Web - Annotation and Authoring[2].

> Tvoříte právě nové stránky? Přemýšleli jste už někdy o tom, jak by se stránky daly udělat „inteligentní“? Nemyslíte, že by bylo velkou výhodu, když by vašim stránkám rozuměly kromě lidských návštěvníků i softwarové programy? Odpověď bude pravděpodobně ano - bylo velmi přínosné, využít prostředky, které by něco takového umožnily i ve vašich stránkách.
>
> Představte si nové možnosti, které takové stránky přinesou. Každý by snadno nalezl co hledá, protože by mohl využít služeb softwarových agentů, pro které není problém vyznat se ve stránkách s rozšířeným „sématickým“ obsahem. Nakupování, vzdělávání se, hledání nejrůznějších dokumentů, získávání kontaktních informací a jakékoliv brouzdání po internetu se stane mnohem efektivnější. Tato budoucnost vůbec není nepravděpodobná. Jistě na ni chcete být připraveni!
>
> Co chybí vašim stránkám, aby se mohly stát součástí takového integrovaného souboru znalostí - sémantických informací, aby se mohly stát součástí Sémantického webu? Jsou to právě sémantické anotace!

Je několik možností, jak anotovat stránky pro sémantický web. I jimi se budeme v této práci zabývat. Doplňme ještě komentářem tvrzení, že každý majitel web-stránek by pravděpodobně ocenil jejich „sémantičnost“.

Softwarový agenti a reklama

V současné době tržní ekonomiky je reklama silný nástroj, který obchodníci používají aby přilákali více zákazníků a reklama na webových stránkách dnes není ničím výjimečným. Je jasné, že softwarový agenti psychologickému tlaku reklamy nepodléhají, a tak vzniká otázka, zda se majitelé komerčních stránek nebudou sémantickým technologiím bránit. Ztrátu, kterou by obchodníci utrpěli snížením efektivity web-reklamy, bude pravděpodobně nutné kompenzovat. Jak by taková kompenzace mohla vypadat, případně jak

[2] http://annotation.semanticweb.org/

bychom mohli obchodníky motivovat k používání sémantických technologií na jejich stránkách, nechme zatím stranou. Je ale možné, že jednou přijde doba, kdy si lidé budou moci snadno vybrat pro ně ideální dostupný výrobek či službu, aniž by museli odolávat psychologickému nátlaku reklamy. Sémantický web a sémantická anotace by k tomu mohly přispět.

1.1.2 Spojme naše znalosti!

Je mnoho oblastí výzkumu, kde se více lidí na různých místech zabývá podobným nebo dokonce stejným tématem. Bylo by jistě přínosné, když by tito lidé mohli snadno porovnat a spojit své výsledky. Takovou oblastí je například Data Mining. Často se stává že více lidí zkoumá stejná data a každý z nich v těchto datech objeví jiné závislosti a vztahy. Každý odborník na závěr svého datového výzkumu sepíše své výsledky do analytické zprávy. Pokud chceme výsledky všech prací nějak porovnat a shrnout, nezbude nám, než projít všechny analytické zprávy a srovnání provést ručně. Nebo se můžeme pokusit znalosti ve zprávách popisované reprezentovat nějakým strojově srozumitelným způsobem. Například tyto zprávy sémanticky anotovat.

Sémantická anotace by v tomto případě umožnila formálně přesně popsat zkoumanou oblast i získané výsledky. Například v projektu STULONG[3] [34] se zkoumají data z medicínského prostředí. V analytických zprávách vystupují stále stejné zkoumané veličiny (výška, váha, věk, BMI, krevní tlak, chorobopis pacienta atd). Podobně prezentované výsledky mají velmi přesnou matematickou interpretaci a tedy i formální sémantiku. Většinou se jedná přímo o hodnotu nějakého matematického vzorce – korelace, Fisherův nebo χ^2 test, aritmetický průměr, hodnoty naměřené u asociačních vztahů, jako je například fundovaná implikace (Když má pacient nadváhu, pak trpí vysokým krevním tlakem – naměřená konfidence 97%).

Tedy jak zkoumanou doménu, tak získané výsledky je možné poměrně přesně formálně popsat. Sémantická anotace analytické zprávy by navíc nemusela klást velké nároky na jejího autora. Formální popis zkoumané domény neboli doménovou ontologii by mohl vytvořit jeden odborník. Autoři zpráv by ji využili a už pouze označili své výsledky vhodnými „pojmy“ doménové ontologie. Sémantický popis konkrétních výsledků by zase mohli generovat přímo analytické nástroje, které autor zprávy použil při své analýze.

Takto anotované analytické zprávy by umožnily integraci znalostí nejen uvnitř jednoho projektu, ale potenciálně napříč celou doménou. Například by

[3] http://euromise.vse.cz/stulong/

bylo možné porovnat výsledky výzkumu aterosklerózy a výzkumu diabetu, u různých skupin pacientů, z různých nemocnic, nebo dokonce z různých zemí.

Zatím jsme mluvili pouze o Data Miningu v medicínské doméně. Nasnadě je však myšlenka, sémantické anotace a integrace přímo lékařských zpráv, které vznikají při každé naší návštěvě lékaře. Samozřejmě by se taková integrace musela provést nanejvýš citlivě vzhledem k osobním údajům pacientů. Realizace takového projektu je pravděpodobně spíš vize než reálná perspektiva několika let, ale přínos by byl jistě nemalý.

1.2 Přínosy práce

Komu? Čím? Srozumitelné pro nelingvistu....

1.2.1 Seznámení s lingvistikou a WordNetem

Tato práce se snaží přiblížit možnosti, jak využít dostupné lingvistické nástroje analyzující český text (sekce 4.4.3) především lidem, kteří se zabývají extrakcí informací z textu, ale nejen jim. Práce na čtenáře neklade žádné nároky co se týká lingvistického vzdělání a sama základní znalosti z lingvistiky podněcuje (kapitola 4). Těmito znalostmi se snaží pokrýt požadavky, které klade používání lingvistických nástrojů zde popisovaných. Zběžné znalosti zde poskytnuté jsou doplněny odkazy a referencemi na zdroje, kde se čtenář o dané problematice může dozvědět více.

V kapitole 5 se čtenář seznámí se sémantickým lexikonem WordNet a získá představu o možnostech jeho využití při analýze českých textů.

1.2.2 Souhrn

Práce poskytuje základní přehled v oblasti sémantické anotace a shrnuje myšlenky, které jsou v jejím pozadí (kapitola 2). Čtenář si může udělat představu o tom, kterými směry se sémantická anotace ubírá, jaké metody byly využity, s jakou úspěšností, ve kterých projektech (oddíl 3.5 a sekce 3.4.1).

V práci jsou shrnuta doporučení autorům webových stánek, jak postupovat při tvorbě stránek se sémantickým obsahem (oddíl 3.3).

1.2.3 Průzkum

Součástí práce je praktický experiment (kapitola 6) s lingvistickými nástroji: Tools for machine annotation z PDT (viz sekce 4.4.3), nástroj pro tektogramatickou analýzu češtiny od Václava Klimeše (sekce 4.4.4), český WordNet (oddíl 5.3) a některé další. Čtenáři jsou poskytnuty zkušenosti z praktického používání těchto nástrojů a z prací které s jejich použitím souvisely. Tyto zkušenosti se týkají především dostupnosti, zprovoznění, výkonnosti, přínosů a nedostatků těchto nástrojů. Postup experimentu se v jednotlivých fázích snaží kopírovat skutečné akce, které by bylo nutné provést v opravdovém projektu zaměřeném na sémantickou anotaci. V práci tak vzniká jednoduchá základní analýza tohoto typu projektů. Ve skutečném projektu pak bude možné ji přinejmenším jako inspiraci využít.

1.2.4 Návrhy

V práci je navržena metodika, pro extrakci informací z přirozeného textu pomocí zmíněných lingvistických nástrojů (sekce 6.1). Je zde navržený jednoduchý dotazovací jazyk pro lingvistické anotace (oddíl 7.1), stručný návrh indukce vzorů (oddíl 7.2) a zamyšlení nad možnostmi lingvistické anotace pro indexaci dokumentů (sekce 7.1.2).

1.2.5 Teorie

V oddíle 3.6 je proveden pokus o teoretický přínos v oblasti sémantické anotace. Oddíl 3.7 se zamýšlí mad možnými přínosy a vztahem mezi počítačovou lingvistikou a formální reprezentací znalostí.

1.3 Sémantický web

V roce 2001 napsal Tim Berners-Lee, tvůrce současného webu a ředitel Konsorcia W3C spolu s dalšími autory velmi známý článek The Semantic Web [1] (volný český překlad je k dispozici například v [4]). V tomto článku je popsána lákavá představa světa, kde všechny nepříjemné problémy spojené se zařizováním běžných životních problémů, jako je například návštěva lékaře, pomáhají vyřídit softwarový agenti. Pomáhají je vyřídit především tím, že naleznou a zkombinují všechny důležité relevantní informace, které jsou potřeba. Například najdou lékaře, který se specializuje na daný druh zdravotních potíží, adresu a otvírací dobu jeho ordinace, dopravní dostupnost tohoto místa atd.

Většina těchto informací je již dnes na webu dostupná, avšak i zkušeného uživatele internetu stojí jejich nalezení nezanedbatelný čas a energii. Navíc nalezení informací je teprve první část problému. To, jakým způsobem je zkombinovat a vyhodnotit, je část druhá. Avšak vyřešení tohoto druhého problému není pro současný software žádnou utopií. Například nalézt dopravní spojení na adresu lékařovy ordinace v dnešní době rozhodně nepovažujeme za programátorsky neřešitelný problém.

Softwaroví agenti pravděpodobně nebudou v dohledné době tak „chytří“, aby se vyznali ve webových stránkách současného internetu a dokázali z nich vytěžit informace, které potřebují. Proto se v souvislosti s myšlenkou sémantického webu snažíme tyto stránky softwarovým agentům přiblížit, udělat je srozumitelnější respektive přístupnější pro strojové získávání informací z nich. Do stránek se vkládají takzvaná metadata (*data o datech*), která co možná nejpřesněji formálně zachycují obsah stránek jinak srozumitelný jen pro člověka. Tomuto obohacování stránek o metadata budeme říkat *sémantická anotace*.

Kapitola 2

Reprezentace znalostí

Jedním ze základních kroků k vytvoření sémantického webu je konceptualizace dat dostupných na internetu [4]. Jedním z klíčových nástrojů konceptualizace jsou ontologie. Ontologie lze charakterizovat jako formalizované reprezentace znalostí určené k jejich sdílení a znovupoužití. Ontologie jsou často doménového (oborového) zaměření a bývají konstruovány jako pojmové (konceptuální) hierarchie nebo sítě.

Právě o konceptuální a formálně přesné vyjádření znalostí, které jsou obsaženy ve zdroji, se snažíme při sémantické anotaci.

2.1 Základní struktury

K modelování znalostí nejčastěji používáme tři základní prvky:

- Třídy (koncepty, kategorie nebo pojmy)
- Role (vztahy, relace, případně vlastnosti)
- Individua (instance tříd, objekty)

Budeme je podrobněji rozebírat v celé této kapitole.

2.2 Deskripční logika

Formálními prostředky pro zachycení znalostí disponuje též deskripční logika. Její vyjadřovací prostředky jsou přehledné, jejich interpretace je přesně definovaná. Deskripční logika se často používá jako základní terminologie

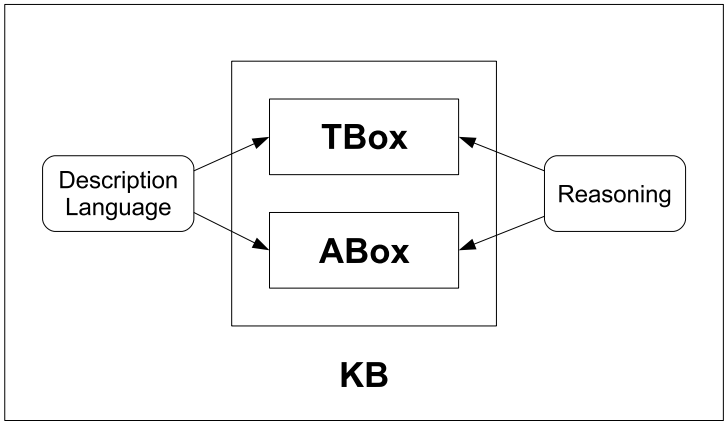
formálního popisu znalostí. Podívejme se na ni nyní trochu podrobněji, pojmy zavedená v deskripční logice pak budeme moci používat dále.

V tomto oddíle se budeme opírat o článek [3], který poskytuje základní úvod do deskripční logiky. Příklady zde uvedené pocházejí z tohoto článku.

Hned na začátku poznamenejme, že pojem *deskripční logika* (DL) je nepřesný. Správně bychom měli mluvit o deskripčních logikách, případně o jazycích deskripční logiky. Jazyků deskripční logiky je mnoho, každý používá trochu jiné konstrukce a má i jinou vyjadřovací sílu. Přesné vymezení jednotlivých deskripčních jazyků je základním předpokladem pro formální studium algoritmů pro odvozování znalostí (reasoning, inference). V této práci se zajímáme spíš o dostupné konstrukce jazyků DL, proto vystačíme s jedním jazykem formálně označovaným jako $\mathcal{ALCN}$, který podrobněji popíšeme později v 2.2.2.

2.2.1 Znalostní báze - knowledge base

Systémy pro reprezentaci znalostí založené na DL tyto znalosti uchovávají v takzvané „znalostí bázi“ (knowledge base). Znalostní báze je také základnou pro odvozování znalostí (reasoning). Reasoning se v těchto systémech používá pro většinu operací se znalostmi: zjišťování subsumpce, ekvivalence či disjunkce tříd, testování konzistence modelu, tj. splnitelnosti všech logických axiomů, zjišťování příslušnosti individuí ke třídám. Tyto operace umožňují vyhodnocování dotazů na znalosti uložené v bázi. A to nejen na znalosti explicitně uvedené, ale i na znalosti, které z nich implicitně plynou, a které se z nich dají pomocí reasoning-u odvodit.



Obrázek 2.1: Knowledge base systému pro reprezentaci znalostí [3]

Obrázek 2.1 ukazuje rozdělení znalostní báze na *TBox* (terminologii) a *ABox* (využití terminologie k popisu světa).

2.2.2 Syntax deskripční logiky

Uvedeme zde syntax destrukčního jazyka $\mathcal{ALCN}$. Jazyk $\mathcal{ALCN}$ vznikne ze základního jazyka deskripční logiky $\mathcal{AL}$ (attributive language) [3] přidáním sjednocení, obecné negace, plného existenčního kvantifikátoru a restrikcí kardinality. Relace povolujeme pouze binární.

Třídy v DL chápeme jako množiny individuí. Třídy se v DL velmi často označují jako *koncepty*. Role reprezentují (binární) relace mezi individui. Základním stavebním kamenem jsou *atomické třídy* a *atomické role*, ostatní koncepty se z nich podle pravidel (konstruktorů) na obrázku 2.2. Písmena C a D zastupují obecné třídy, písmeno A atomickou třídu, písmeno R atomickou relaci.

$C, D \rightarrow$	$A \mid$	(atomický pojem)
	$\top \mid$	(univerzální pojem)
	$\bot \mid$	(prázdný pojem)
	$\neg C \mid$	(negace)
	$C \sqcap D \mid$	(průnik)
	$C \sqcup D \mid$	(sjednocení)
	$\forall R.C \mid$	(hodnotová restrikce)
	$\exists R.C \mid$	(existenční kvantifikátor)
	$\geqslant n\, R \mid$	(maximální kardinalita)
	$\leqslant n\, R$	(minimální kardinalita)

Obrázek 2.2: Syntax deskripční logiky [3]

2.2.3 Interpretace deskripční logiky

Formální sémantiku DL definujeme pomocí *interpretace* $\mathcal{I}$. Interpretace $\mathcal{I}$ se skládá z neprázdné množiny $\Delta^{\mathcal{I}}$ (doména interpretace) a interpretační funkce, která každé atomické třídě A přiřadí množinu $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ a každé atomické roli R přiřadí binární relaci $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Interpretace odvozených konstrukcí je zapsána na obrázku 2.3.

Říkáme, že třídy C a D jsou ekvivalentní ($C \equiv D$), když $C^{\mathcal{I}} = D^{\mathcal{I}}$ při libovolné interpretaci $\mathcal{I}$. Například třída zapsaná jako $\forall \mathsf{hasChild}.\mathsf{Female} \sqcap \forall \mathsf{hasChild}.\mathsf{Student}$ je ekvivalentní třídě $\forall \mathsf{hasChild}.(\mathsf{Female} \sqcap \mathsf{Student})$. Vztah podtřídy a nadtřídy (subsumpce) $C \sqsubseteq D$ je definován analogicky $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \ : \ \forall b.\ (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \ : \ \exists b.\ (a,b) \in R^{\mathcal{I}} \ \wedge \ b \in C^{\mathcal{I}}\} \\
(\geqslant n\, R)^{\mathcal{I}} &= \left\{a \in \Delta^{\mathcal{I}} \ : \ \left|\{b \ : \ (a,b) \in R^{\mathcal{I}}\}\right| \geq n\right\} \\
(\leqslant n\, R)^{\mathcal{I}} &= \left\{a \in \Delta^{\mathcal{I}} \ : \ \left|\{b \ : \ (a,b) \in R^{\mathcal{I}}\}\right| \leq n\right\}
\end{aligned}
$$

Obrázek 2.3: Interpretace deskripční logiky [3]

2.2.4 TBox

TBox reprezentuje terminologii, dalo by se též říci slovník znalostní báze. Uvnitř TBox-u jsou definovány *třídy* a *role* znalostní báze pomocí *axiomů* DL. Tyto axiomy se tvoří pomocí relací subsumpce ($\sqsubseteq$) a ekvivalence ($\equiv$).

Důležitou vlastností TBox-u je jeho bezespornost tj. existence jeho *modelu*. Modelem rozumíme takovou interpretaci $\mathcal{I}$, která splňuje všechny axiomy terminologie (všechny ekvivalence a subsumpce). Podrobnosti je možné nalézt v [3].

Příklad formálně zapsaného TBox-u je na obrázku 2.4.

2.2.5 ABox

ABox je množina tvrzení o pojmenovaných individuích. Pro vyjádření jednotlivých tvrzení používáme terminologii (TBox) znalostní báze. ABox můžeme též nazvat množinou tvrzení o světě. Jednotlivá individua se zde zařadí do tříd a rolí pomocí následující syntaxe.

Obrázek 2.4: Terminologie (TBox) pro popis rodinných vztahů [3]

$$
\begin{aligned}
\mathsf{Woman} &\equiv \mathsf{Person} \sqcap \mathsf{Female} \\
\mathsf{Man} &\equiv \mathsf{Person} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Father} &\equiv \mathsf{Man} \sqcap \exists\mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Parent} &\equiv \mathsf{Father} \sqcup \mathsf{Mother} \\
\mathsf{Grandmother} &\equiv \mathsf{Mother} \sqcap \exists\mathsf{hasChild}.\mathsf{Parent} \\
\mathsf{MotherWithManyChildren} &\equiv \mathsf{Mother} \sqcap \geqslant 3\,\mathsf{hasChild} \\
\mathsf{MotherWithoutDaughter} &\equiv \mathsf{Mother} \sqcap \forall\mathsf{hasChild}.\neg\mathsf{Woman} \\
\mathsf{Wife} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasHusband}.\mathsf{Man}
\end{aligned}
$$

Nechť a, b, c jsou jména individuí.

Tvrzení, že a patří do třídy C, zapíšeme: $C(a)$.
Tvrzení, že c je s b spojeno rolí R zapíšeme: $R(b, c)$.

Sémantiku ABox-u zavedeme rozšířením výše zmíněné *interpretace* $\mathcal{I}$ na jednotlivá individua. Každému individuu a přiřadíme nějaký prvek $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ z domény. Toto přiřazení provedeme vzájemně jednoznačně. Tedy různým individuím a, b odpovídali různé prvky domény ($a^{\mathcal{I}} \neq b^{\mathcal{I}}$).

Důležitou vlastností ABox-u je jeho konzistentnost – existence modelu, který splňuje podmínky kladené použitou terminologií (TBox).

Říkáme, že interpretace $\mathcal{I}$ *splňuje* tvrzení $C(a)$ když $a^{\mathcal{I}} \in C^{\mathcal{I}}$
a tvrzení $R(b, c)$ když $(b^{\mathcal{I}}, c^{\mathcal{I}}) \in R^{\mathcal{I}}$.

Pokud interpretace $\mathcal{I}$ splňuje všechna tvrzení ABox-u, říkáme, že je jeho *modelem*. Podrobnosti je možné nalézt v [3].

Příklad formálně zapsaného ABox-u je na obrázku 2.5.

$\mathsf{MotherWithoutDaughter}(\mathsf{MARY})$ $\quad$ $\mathsf{Father}(\mathsf{PETER})$
$\mathsf{hasChild}(\mathsf{MARY}, \mathsf{PETER})$ $\quad$ $\mathsf{hasChild}(\mathsf{PETER}, \mathsf{HARRY})$
$\mathsf{hasChild}(\mathsf{MARY}, \mathsf{PAUL})$

Obrázek 2.5: Tvrzení o individuích (ABox) rodinných vztahů [3]

2.3 Ontologie

Ontologie v tradičním filosofickém pojetí označuje nauku o bytí (jsoucnu). V oblasti informatiky jsou ontologie chápány jako explicitní, formální specifikace pojmů a vztahů mezi nimi. V [5] je uvedena definice W. Borsta: „Ontologie je formální specifikace sdílené konceptualizace." Tedy cílem ontologie je definovat společné, jednotné chápání pojmů určité oblasti. Pro sémantickou anotaci jsou ontologie ideální referenční základnou pro vyjádření znalostí, které je možné sdílet například v rámci sémantického webu.

Hlavními prvky, které tvoří strukturu ontologie, jsou opět *třídy*, *role* a *individua*. Naproti DL se však tyto prvky používají více „informaticky", množina individuí je rozšířena o *primitivní hodnoty* definovaných datových typů, role se klasifikují a přesněji specifikují, zavádějí takzvané vlastnosti – *properties*, je možné určit jejich definiční obor, obor hodnot, dají se deklarovat jako funkční, tranzitivní atd.

Základní kostrou ontologie je hierarchie tříd – taxonomie. Ta se narozdíl od DL[1] většinou definuje explicitně pomocí role *rdfs:subClassOf* nebo pomocí nějakého jejího ekvivalentu v jazycích, které nejsou založeny na RDF. Definice tříd je možné doplnit axiomatikou, která je například v OWL přímo převzata z DL. Pomocí axiomů je v OWL možné vyjádřit ekvivalenci tříd či rolí, disjunknost tříd, implicitní zařazení individuí do třídy pomocí dodatečných podmínek, atd. Ontologie tak může představovat plnohodnotnou znalostní bázi s reasoningem DL.

Není jisté, jestli pro potřebu sémantického webu potřebujeme axiomatiku DL, která jazyk ontologie zesložiťuje. Představa sémantického webu jako jedné veliké znalostní báze, ve které se každý dotaz dá vyhodnotit pomocí obecného reasoningu, se zdá být poněkud utopická. Naproti tomu větší vyjadřovací síla jazyků může skutečně zvyšovat míru axiomatizace teorií, a tím snad i objektivitu ontologií [5]. Otázkou je, kde všude je možné matematickou axiomatizaci aplikovat. Například už u medicíny by byl matematický způsob dosahování objektivity v řadě případů kontroverzní, o většině společenských věd ani nemluvě.

Při tvorbě stránek pro sémantický web bychom se rádi opřeli o slovník nějaké standardní, uznávané ontologie. Větších i menších ontologií vzniklo v posledních letech mnoho, na standard si však pravděpodobně budeme muset ještě počkat. Přehled několika známých ontologií uvádíme v oddílu 3.2. Ukazuje se, že použití „špatné" ontologie nemusí být až takový problém. Konstrukce jednotlivých ontologií se na sebe dají vzájemně „mapo-

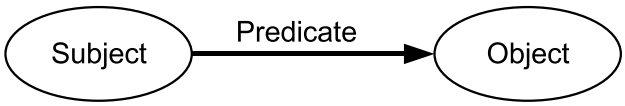
[1] Subsumpce tříd je v DL vyhodnocována dynamicky na základě popisů tříd, tedy i celá taxonomie vzniká dynamicky.

vat“. Problémem mapování ontologií se podrobně zabývá Michal Fiedler ve své práci [6].

Uvedeme nyní dva hlavní formální jazyky používané pro reprezentaci ontologií na webu: RDF a OWL.

2.3.1 RDF - Resource Description Framework

Technologickým základem sémantického webu by se podle organizace W3C měl stát její standard RDF – Resource Description Framework[2]. Jde o obecný rámec pro popis, výměnu a znovupoužití metadat. Rámec RDF poskytuje jednoduchý model pro popis zdrojů. Datový model RDF je založen na tříprvkové konstrukci *subject* (subjekt) – *predicate* (predikát) – *object* (objekt), viz obrázek 2.6.



Obrázek 2.6: RDF triple: *subject – predicate – object*

Této tříprvkové konstrukci říkáme *tvrzení* (statement, případně RDF triple). Každé takové *tvrzení* říká, že vlastnost identifikovaná jako *predikát* daného *subjektu* má hodnotu *objekt*. Neboli *subjekt* je s *objektem* spojen rolí *predikát*. Každý z trojice prvků libovolného *tvrzení* reprezentuje nějaký *zdroj* (resource) a je jednoznačně identifikovaný svým URI[3]. Zdrojem může být web-stránka, objekt web-stránky či libovolná entita webu i mimo web, která se dá identifikovat pomocí URI. Výjimku z tohoto pravidla představuje situace, kdy *objekt* obsahuje *literál* – primitivní hodnotu nějakého datového typu (viz dále). Nejčastěji jako zdroje používáme jednotlivé *třídy*, *predikáty*. Individua nejsou v RDF explicitně zavedena.

V zápisech zdrojů pomocí URI máme možnost používat zkratky – takzvané *namespace prefix* známé z XML. V dalším textu budeme používat následující prefixy.

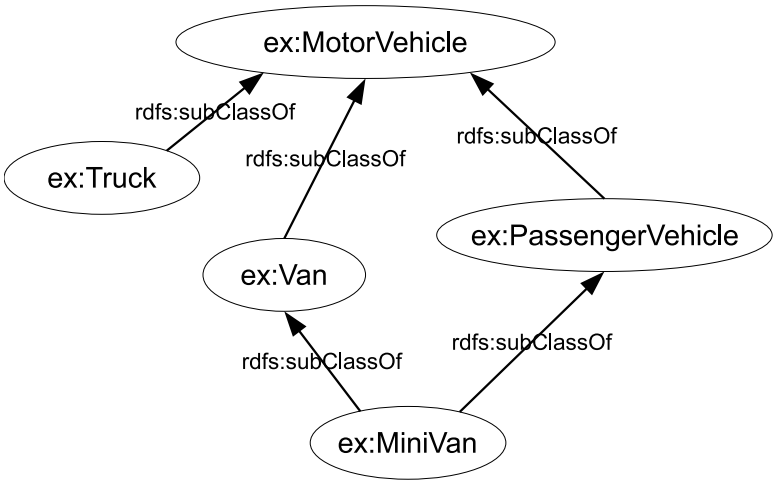
[2]http://www.w3.org/RDF/
[3]URI – Uniform Resource Identifier, RFC 1630

prefix	namespace URI	poznámka
rdf:	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`	RDF zdroje
rdfs:	`http://www.w3.org/2000/01/rdf-schema#`	RDFS zdroje
owl:	`http://www.w3.org/2002/07/owl#`	OWL zdroje
xsd:	`http://www.w3.org/2001/XMLSchema#`	datové typy
ex:	`http://example.org/schemas/vehicles#`	příklady W3C
exterms:	`http://example.org/terms/`	...
exthings:	`http://example.org/things/`	...

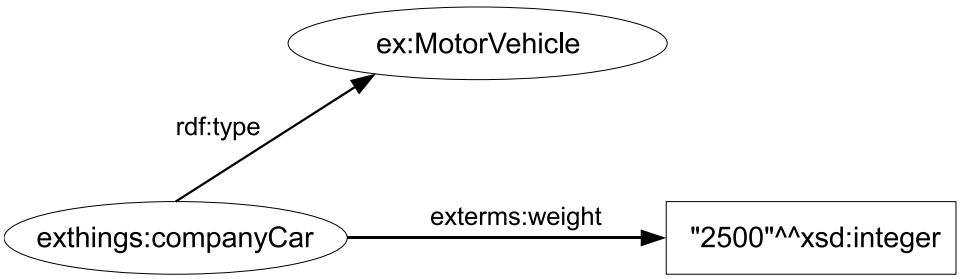
RDF Schema

RDF Schema[4] (RDFS) je jednoduchou nadstavbou RDF, která definuje několik konstrukcí pro reprezentaci znalostí. Na obrázku 2.7 je vidět použití predikátu *rdfs:subClassOf* k vytvoření jednoduché taxonomie motorových vozidel. Na obrázku 2.8 je vidět použití predikátu *rdf:type* pro přiřazení „individua“ *exthings:companyCar* do třídy *ex:MotorVehicle*. Dále je zde vidět použití nového predikátu *exterms:weight* k přiřazení váhy tomuto vozidlu pomocí jednoduché hodnoty *2500* datového typu *xsd:integer*. Těmto hodnotám říkáme *literály*.

[4]http://www.w3.org/TR/rdf-schema/



Obrázek 2.7: Příklad hierarchie tříd v RDF: A Vehicle Class Hierarchy
Příklad pochází z dokumentu RDF Primer.
http://www.w3.org/TR/rdf-primer/



Obrázek 2.8: Přiřazení individua do třídy a použití literálu. Příklad pochází z dokumentu RDF Primer.

Konstrukce *rdfs:domain* a *rdfs:range* umožňují přiřadit predikátům jejich definiční obor a obor hodnot. Toto přiřazení však není restriktivní. RDF není formalismus pro syntaktickou kontrolu. Tím, že deklarujeme definiční obor predikátu, nezužujeme možnosti jeho použití, ale dodáváme o něm další znalosti. Například deklarací definičního oboru *ex:MotorVehicle* pro predikát *exterms:weight* říkáme, že všechny zdroje uvedené jako *subjekt* predikátu *exterms:weight* automaticky patří do třídy *ex:MotorVehicle*. Podrobnosti o sémantice RDF je možné nalézt v článku RDF Semantics[5] na webu W3C.

Každý popis zapsaný v RDF je složený ze samostatných *tvrzení* – uspořádaných trojic zdrojů. Jako takový je nezávislý na konkrétní fyzické interpretaci, nejčastěji se však ukládá ve formě XML. Popis této XML syntaxe je podrobně zpracován v článku RDF/XML Syntax Specification[6] na webu W3C.

Velkou výhodou RDF je jeho široké rozšíření, RDF je doporučením W3C z roku 1999 pro reprezentaci struktury webových metadat. RDF je však velmi obecný, například neumožňuje precizněji specifikovat podmínky příslušnosti ke třídám. Pro konstrukci ontologií se většinou používají jeho nadstavby jako například OWL. Další nevýhodou RDF je jeho přílišná jednoduchost. Některé složitější konstrukty ontologií (logické výrazy) musí být rozloženy do několika trojic RDF, které se spojují pomocí proměnných. Sémantika tvrzení je pak poměrně „zatemněná“, hovoří se o takzvaných „dark triples“ [5].

[5] http://www.w3.org/TR/rdf-mt/
[6] http://www.w3.org/TR/rdf-syntax-grammar/

2.3.2 OWL - Web Ontology Language

Jazyk OWL[7] nabízí bohatší slovník a sémantiku pro budování ontologií. Vychází z RDF a RDFS, využívá jejich vlastnosti a přidává nové možnosti pro přesnější definici tříd, rolí i individuí. OWL přebírá většinu konstrukcí deskripční logiky a propojuje je s RDF, rozšiřuje koncept definovaných tříd, umožňuje použití logických podmínek. Jazyk OWL vydalo W3C jako své doporučení v roce 2004.

Kvůli snazší implementaci OWL v programových nástrojích a možnostem efektivního reasoningu je jazyk OWL rozdělen do tří tříd podle složitosti:

- OWL Lite
- OWL DL
- OWL Full

Sémantika OWL je trochu odlišná od RDF. Například RDF příliš nerozlišuje mezi individui a třídami. Stejný zdroj může v jednom tvrzení vystupovat jako třída a v jiném jako „individuum“. Sémantika OWL je přísnější. Například takto volný vztah mezi individui a třídami povoluje až na úrovni OWL Full. Až jazyk OWL Full je plným nadjazykem RDF/S. Díky těmto restrikcím můžou v ontologii definované pomocí OWL vzniknout nekonzistence a sporná tvrzení. Složitost programového algoritmu pro kontrolu konzistence se liší na různých úrovních jazyka OWL. Podrobnosti o OWL sémantice je možné najít v článku OWL Web Ontology Language Semantics and Abstract Syntax[8] na webu W3C.

Uvedeme nyní některé konstrukce OWL rozdělené do oblastí jejich vyžití.

Množinové operace

`owl:disjointWith`	Označuje dvě třídy jako disjunktní.
`owl:unionOf`	Definuje třídu jako sjednocení dvou stávajících.
`owl:complementOf`	Definuje třídu jako doplněk stávající.
`owl:intersectionOf`	Definuje třídu jako průnik dvou stávajících.

[7] http://www.w3.org/2004/OWL/
[8] http://www.w3.org/TR/owl-semantics/

Definice rolí

Pomocí následujících definovaných tříd v OWL vymezujeme typ role.

`rdf:Property`	Třída všech rolí.
`owl:ObjectProperty`	Role jejichž objektem jsou individua.
`owl:DatatypeProperty`	Role jejichž objektem jsou literály.
`owl:TransitiveProperty`	Tranzitivní role.
`owl:SymmetricProperty`	Symetrické role.
`owl:FunctionalProperty`	Funkční role.
`owl:inverseOf`	Predikát pro označení dvou inverzních rolí.

Omezení rolí podmínkami

Role je možné v RDF omezit pomocí konstrukcí *owl:Restriction* a *owl:onProperty*. Takovým omezením role můžeme například definovat novou podtřídu, která obsahuje jen individua, která vyhovují kladeným podmínkám. Podmínky je pak možné rozlišit na nutné a postačující. K dispozici jsou následující konstrukce známé z DL.

`owl:allValuesFrom`	Univerzální kvantifikátor.
`owl:someValuesFrom`	Existenční kvantifikátor.
`owl:minCardinality`	Minimální kardinalita role.
`owl:maxCardinality`	Maximální kardinalita role.
`owl:cardinality`	Přesné určení kardinality role.

Ekvivalence a rozdílnost tříd

OWL obsahuje i několik konstrukcí pro vzájemné vymezení tříd.

`owl:equivalentClass`	Množina individuí příslušných tříd je stejná.
`owl:equivalentProperty`	Deklaruje dvě role jako shodné.
`owl:sameAs`	Deklaruje ekvivalenci individuí.
`owl:differentFrom`	Deklaruje rozdílnost individuí.

Kapitola 3

Sémantická anotace

Sémantická anotace není v současné době přesně vymezený termín. V této práci budeme s tímto pojmem pracovat poměrně volně až na kapitolu o teoretických otázkách sémantické anotace 3.6, kde se o takové vymezení pokusíme. V knize [2] se tomuto termínu vyhýbají opisem *annotation for the Semantic Web* tedy anotace pro sémantický web. V tomto smyslu budeme užívat pojem sémantické anotace ve zbytku práce. Avšak nemusíme se nutně omezovat pouze na anotaci zdrojů určených pro veřejnou publikaci na internetu. Již v motivačním příkladu 1.1.2 jsme naznačili, jak by bylo přínosné proces sémantické anotace aplikovat v oblasti psaní analytických zpráv. Proces sémantické anotace se v tomto případě téměř neliší od anotace pro sémantický web. Stále se snažíme o přesné, formální a volně-přenosné zachycení znalostí, které jsou ve zdroji obsaženy.

Pro přesnost ještě doplňme, že anotací jako hotovým dílem rozumíme výsledek, výstup případně výstupní data procesu anotace. Může se jednat o nějak označkovaný text, který byl vstupem tohoto procesu. Častěji to budou strukturovaná data - HTML případně XML, obohacená novými - sémantickými značkami. Sémantickou anotací však mohou vzniknout i data na původním zdroji nezávislá. Případně data sice nezávislá, ale obsahující v sobě odkazy k původu svého vzniku. Například samostatný datový soubor obsahující sémantickou anotaci nějakého textu může u jednotlivých „znalostí“ obsahovat i pointery na slova, ze kterých tyto znalosti vznikly ve vstupním textu. Podrobněji rozebereme uložení a reprezentaci sémantické anotace v oddíle 3.1.

3.1 Reprezentace anotací

RDFa, HTML-A

3.2 Uznávané ontologie

3.3 Autorům Web-stránek

3.4 Dodatečná anotace, strojová

3.4.1 Rozdělení

předělat, doplnit, rozvést

Po praktické stránce: dostupnost, upravitelnost (jiná doména), stabilita - náchylnost na změny v datech

Po teoretické stránce: čísla (úspěšnost), metody, podle [7] + doplnit.

```
*  Multistrategy
      o Pattern-based
            + Discovery (Seed expansion)
            + Rules (JAPE, Taxonomy label matching)
      o Machine Learning-based
            + Probabilistic (Hidden Markov Models, N-gram analysis)
            + Induction (Linguistic, Structural)
```

Platform	Method	Machine Learning	Manual Rules	Bootstrap Ontology
AeroDAML	Rule	N	Y	WordNet
Armadillo[1]	Pattern Discovery	N	Y	User
KIM	Rule	N	Y	KIMO
MnM[2]	Wrapper Induction	Y	N	KMi
MUSE[3]	Rule	N	Y	User
Ont-O-Mat: Amilcare[4]	Wrapper Induction	Y	N	User
Ont-O-Mat: PANKOW[5]	Pattern Discovery	N	N	User
SemTag[6]	Rule	N	N	TAP

3.5 Projekty

3.5.1 Semantic MediaWiki

Semantic MediaWiki (SMW) [8] je rozšířením známého *MediaWiki* enginu, který kromě mnoha jiných wiki-stránek zajišťuje i chod otevřené Wikipedia encyklopedie[7]. Už samotná MediaWiki v některých rysech připomíná sémantický web. Uživatel, který píše článek v klasické MediaWiki, má mnoho možností, pro formátování textu. Kromě formátování však může jednotlivým slovům přiřadit význam tak, že je označí jako hypertextové odkazy. Podívejme se například na větu z české Wikipedie:

```
Praha je [[hlavní město]] [[Česko|České republiky]].
```

Slovní spojení „hlavní město“ je označeno a toto označení znamená, že jeho *sémantiku* vysvětluje článek se stejným názvem. Podobně sémantika slovního spojení „České republiky“ je popsána v článku s názvem „Česko“. Nepřipomíná to sémantickou anotaci?

Dalšími možnostmi anotace v klasické MediaWiki jsou zařazení článku do některé kategorie a použití takzvané *šablony*. Šablony jsou v klasické MediaWiki určeny především pro snazší formátování podobných článků, ale lze jimi definovat i jisté „datové schéma“.

SMW původní MediaWiki rozšiřuje o další sémantické vlastnosti a skutečně sémanticky wiki-data interpretuje za použití ontologií a OWL/RDF. Všechna sémantické data, která sem autoři článků vkládají je možné snadno exportovat do OWL/RDF formátu. V článcích SMW má autor následující tři základní možnosti anotace:

Kategorie

V SMW tvoří všechny články dohromady jedinou ontologii. Ta se s každým novým článkem rozšiřuje a mění. Kategorie hrají v této ontologii roli tříd, články roli individuí. Tím, že článek zařadíme do některé kategorie, zařadíme i odpovídající individuum do odpovídající třídy.

To, že Praha patří do kategorie město, v SMW vyjádříme uvnitř článku o Praze zápisem: `[[Category:City]]`

[7] http://wikipedia.org

Relace

Relace umožňují definovat binární vztah mezi články SMW respektive binární relaci mezi individui ontologie SMW. Výše uvedená věta v článku o Praze by v SMW vypadala takto:

```
Praha je [[Relation:capital of|hlavní město]]
[[capital of::Česko|České republiky]].
```

„Relation:capital of“ reprezentuje odkaz na článek, který popisuje relaci s názvem „capital of“.

Atributy

Atributy se používají k definici vlastností článku, jejichž hodnota má jednoduchý datový typ. Jsou to většinou čísla, časy, data, které se k článku vztahují. Jako příklad uveďme větu o rozloze našeho hlavního města:

```
Katastrální výměra Prahy je [[area:=496km^(2)]].
```

V ontologii se opět hodnota atributu vztahuje k individuu, které článek reprezentuje. Kategorie, relace i atributy může do SMW kdokoliv přidávat stejně jako jakékoliv jiné články. Prostě vytvoří článek s odpovídajícím názvem (například „Attribute:Area“). Hezkou vlastností atributů je možnost definovat jejich typ. Například atribut *Attribute:Area* je typu *Type:Area*. Při čtení článku si čtenář může nechat všechny atributy přepočítat do jednotek které mu vyhovují (vzdálenosti na metry či míle, objem na litry či galony a podobně).

Anotace článků SMW přináší několik výhod. Jednou z nich je možnost do článků vkládat takzvané „inline query“. Jedná se o dotaz na data ontologie. Výsledek dotazu se objeví v textu článku na místě, kde je dotaz zapsán. Obsah článku se tak stává dynamickým, takto vložená data jsou vždy aktuální. Například můžeme chtít při psaní článku o České republice zmínit pět našich největších měst. Místo abychom do článku opsali tabulku s těmito městy a jejich počty obyvatel případně rozlohou, vložíme do článku dotaz, který nám tuto tabulku vygeneruje. Čísla v tabulce budou vždy aktuální, tak jak jsou právě uvedena v článcích o jednotlivých městech.

Už klasická MediaWiki je výjimečná tím, jak kdokoliv může měnit a rozšiřovat její obsah. Velmi to připomíná samotný web a jeho živelné vznikání. SMW tento rys zachovává ve všech svých vlastnostech. V řeči ontologie to

znamená, že kdokoliv může měnit jak ABox (individua) tak TBox (třídy, relace, definice atributů). Při sémantické anotaci obecně bychom rádi měli nějakou globálně uznávanou a používanou ontologii. Taková ontologie by časem mohla vzniknout z sémantické Wikipedia encyklopedie. Na jejím vzniku i používání by se podílelo velké množství lidí, brzy by se ukázalo, které koncepty jsou funkční a které nikoliv.

SMW engine je volně dostupný včetně zdrojových kódů na stránkách SMW projektu[8]. Zkušební verze SMW běží na Internetu na stránkách Ontoworld.org a každý si ji může vyzkoušet. SMW na Ontoworld.org je poměrně bohatá a poskytuje dobrou dokumentaci celého SMW enginu. V naší práci jsme Ontoworld.org využili k publikaci některých pracovních poznámek[9].

3.5.2 Artequakt

[9]

Automaticky generuje texty životopisů známých umělců. Podklady sbírá sémantickou anotací článků na webu.

3.5.3 WEESA

V článku [10] je navržena přímočará deterministická metoda, jak XML data převádět na RDF data. K XML datům je nejprve nutné vytvořit jednoduchou ontologii pokrývající třídy a atributy, které chceme převést. Tato ontologie se pak automaticky naplní daty ze zdrojového XML. Metoda dobře funguje pro "dobře strukturované" XML data (XML elementy odpovídají třídám atp.) Problémy nastávají s daty tvaru:

```
<SearchDataElement name="Processor Speed" value="500MHZ">
```

To, že se jedná o vlastnost Processor Speed s hodnotou 500MHZ, tato metoda nedokáže postihnout (záměrně – důraz byl kladen na jednoduchost a konzistenci návrhu).

3.5.4 The Lixto Project

[11]

[8] http://sourceforge.net/projects/semediawiki

[9] Například článek se souhrnnými informacemi má adresu:
http://ontoworld.org/wiki/Semantic_annotation_of_data_from_web_resources

!!!upravit!!! Software umožňující automaticky stahovat vybrané informace z webových stránek. Uživatelsky příjemné a propracované prostředí. Jedná se o nástroj, ve kterém si uživatel může naprogramovat svůj wrapper šitý na míru svým potřebám.

Lixto server

Lixto server umožňuje automaticky spouštět hotové Wrappery a získaná data dále zpracovat. Pro zpracování dat je k dispozici široká paleta možností. Zpracovaná data server umí doručit nejrůznějším aplikacím i zařízením.

Možnosti automatické sémantické anotace

Pro využití při automatizované sémantické anotaci je velkou překážkou nutnost pokaždé znovu ručně naprogramovat Lixto na každou stránku.

3.5.5 GATE

[12] !!!doplnit!!!

3.5.6 The KIM Platform

[13] The KIM Platform: Knowledge & Information Management, Bulharsko

Při prezentaci vyvolal výpadek serveru úsměv u většiny posluchačů. K povznesené náladě jistě přispěl i fakt, že se jedná o projekt bulharský.

3.5.7 Image Semantics without Annotations

Teoreticky bohatý článek [14], zabývá se indexací a vyhledáváním obrázků v internetu podobné databázi. Definuje internet jako graf provázaných dokumentů a obrázků. Navržena je algebra pro manipulaci s takovým grafem (operace jako: insert, delete, nodes, edges, union, ...) Velmi propracované je porovnávání obrázků pomocí (adaptivní) podobnostní míry, která se skládá ze tří složek:

- Linguistic Modality
  - popisky, okolní text
- Closed Word Modality

- podobná Linguistic Modality, ale slova zde mají přesný význam
- ustálené, přesné pojmy uvnitř uzavřené komunity

- Emergent Modality příp. User Modality
  - míra vzniklá z akcí a operací souvisejících s obrázky
  - Nastavuje se např. pomocí dialogu ve kterém uživatel "přetahuje" podobné obrázky k sobě.

3.5.8 MUMIS - Multi-Media Indexing and Searching

NLP projekt MUMIS[10] využívající sémantickou anotaci (nejen) textů o fotbalových zápasech k (sémantické) indexaci videozáznamů těchto zápasů. Informace o zápasech jsou získávány z různých zdrojů (texty - vice i méně strukturované, zvukový záznam řeči) v různých jazycích (En, Ge, Nl - Dutch).

3.6 Teorie

Domnívám se, že nebylo mnoho teoretických otázek týkajících se sémantické anotace vůbec formulováno, natož uspokojivě vědecky vyřešeno.

3.6.1 Otázky

V čem tkví sémantičnost anotace?

Aby notace byla sémantická, musí být jistě konzistentní s tím, jak lidé znalosti a informace chápou.

Na sémantickou anotaci se můžeme dívat ze dvou úhlů. Totiž z hlediska její standardnosti a z hlediska její otevřenosti k odvozování znalostí.

- Standardnost anotace můžeme měřit tím, jak široké publikum „posluchačů“ jí bude rozumět.
- Otevřenost anotace k odvozování znalostí chápeme jako možnost ji přímo použít při automatické dedukci znalostí. Chceme, aby se tato anotace mohla zapojit do znalostní báze a mohla zde sloužit jako zdroj dalších znalostí.

[10] http://hmi.ewi.utwente.nl/Projects/mumis/

Co si představujeme pod ideální sémantickou anotací?

V kontextu dvou pohledů výše:

- Plně standardní anotace, které by každý rozuměl.
- Anotace, která má maximální možný užitek pro znalostní bázi, všechny anotované informace lze využít pro odvozování dalších znalostí.

Jak vypadá ideální anotace v ideálním případě?

Pokusit se stanovit podmínky které ideální anotaci braní, resp. předpoklady které by ji umožnily.

Jistě budeme chtít jedinou všemi používanou ontologii, případně dokonalá mapování. Budeme však též požadovat, aby už obsahovala všecky znalosti světa? My bychom pak pouze označovali, které části textu respektive zdroje vyjadřují jednotlivé znalosti. Pak už bychom ale nepotřebovali nic anotovat, už známe všechny odpovědi.

Jaký je vztah lingvistické a sémantické anotace?

Jsou mezi nimi hranice? Kde přibližně?

Dá se lingvistická anotace převést na sémantickou?

Dal by se z toho odvodit nějaký univerzálnější návod / algoritmus, jak od lingvistické anotace k té sémantické přejít?

Jaký je vztah mezi přirozeným jazykem a deskripční logikou?

Kapitola 3.7

3.7 Lingvistika a znalostní inženýrství

3.7.1 Nejednoznačnost při lingvistické anotaci

Potřebujeme další znalosti, *znát* kontext. Mnoho informací je vyjádřeno implicitně.
u příkladů doplnit zdroj...

Banky snižují úroky z ekonomických důvodů.
Banky snižují úroky z krátkodobých půjček.

Ženu ženu.

I saw a man with a telescope.

Včera jsem chytil tlouště na višni.

3.7.2 Znalosti formulované v přirozeném jazyce

Existuje více typů vět i zdrojů - provést kategorizaci.

Žena se stane matkou, když porodí dítě.

Uspořádaná dvojice je dvouprvková množina, ve které rozlišujeme pořadí prvků.

Hasičský oddíl z Hostinného vyrazil v 7.46 k požáru v Dolní Kalné.

Vstup zakázán.

Podepište se, prosím.

Uchazeč vyplní obor studia, výzkumné téma, školicí pracoviště, zajistí podpis školitele a podpis předsedy oborové rady.

Poslední věta je formulovaná obecně: pro všechny uchazeče, pro libovolné téma, pracoviště, podpis libovolného školitele, předsedy. Avšak školitel je pevně spojen s tématem práce a předseda je spojen s pracovištěm.

Jak anotovat takovou větu? Jaká je její sémantika?

Máme několik možností.

- TBox - tvrzení o pojmech.
- Zapojit proměnné, které probíhají prvky domény a vytvořit na ně restrikce.
- Nebo se na větu dívat v kontextu jazyka formulářů a brát ji jako obsah jedné konkrétní instance formuláře.

Co přináší jaké výhody?

3.7.3 Rozdíl mezi konkrétním a abstraktním

Třídy X instance, pojmy X individua.

Tvrzení se často dají chápat jako obecná z jednoho pohledu a konkrétní z druhého.

Banky snižují úroky z ekonomických důvodů.
v kontextu
Investiční i Komerční banka informovaly o plánované změně úrokové míry.

Kapitola 4

Lingvistická anotace

Lingvistickou anotací budeme v této práci označovat činnost, při které se text přirozeného jazyka obohacuje o lingvistickou informaci o slovech, větách, vztazích mezi slovy, mezi větami, o typu a původu textu atp. Lingvistická anotace nebo též značkování korpusu je jedna z činností korpusové lingvistiky. Korpusem rozumíme soubor textů spolu s dodanou lingvistickou informací. Korpusová lingvistika je poměrně novou disciplínou, jejíž vznik, stejně jako vznik celé počítačové lingvistiky vůbec, umožnil rozvoj výpočetní techniky posledních let. Korpusová lingvistika se zabývá zkoumáním a shromažďováním textů přirozeného jazyka (vytvářením korpusu). Texty se anotují za velké podpory počítače - například morfologická desambiguace (viz 4.4.3) by byla bez softwarové podpory nadlidský úkol. Avšak pro anotaci korpusu je stále nutná spousta lidské „ruční“ práce. Takto anotované texty představují velmi cenná data, ze kterých se především pomocí statistických metod dají vyvodit nové poznatky o jazyce. Díky ručně anotovaným korpusům vynikla a stále vyniká většina softwarových nástrojů pro počítačové zpracování přirozeného jazyka.

Korpusová lingvistika dnes nemalou mírou přispívá k jazykovému výzkumu. Na stránkách Českého národního korpusu[1] se dokonce uvádí, že přináší natolik nové poznatky o jazyce, že do dosavadního vývoje jazykovědy vnáší radikální převrat. Toto tvrzení nemusí působit překvapivě, pokud například srovnáme původní latinskou lingvistickou terminologii s tou, která vzniká v počítačové lingvistice v poslední době.

Korpusy se v zásadě značkují třemi druhy značek:

Značky správní zachycují identifikační údaje o každém textu - informace o jeho původu a zdroji.

[1] http://ucnk.ff.cuni.cz/

Značky strukturní zachycují hierarchickou strukturu textu tj rozdělení textu do kapitol, odstavců, vět a *tokenů* (slov a interpunkčních znamének).

Značky lingvistické jsou přiřazeny k jednotlivým slovům a nesou informaci o lingvistických kategoriích, které dané slovo nese.

Samostatnou kapitolou lingvistické anotace je potom zachycení gramatické stavby věty [16]. K tomu se používají dva typy gramatik - *složková* a *závislostní* gramatika. Závislostní gramatika má dlouholetou tradici v popisu jazyků evropského kontinentu a zdá se, že má určité výhody i pro popis angličtiny, která bývá častěji zpracovávána gramatikou složkovou.

Složkový popis je blíže Chomského pojetí jazyka. Věta je podle složkové gramatiky rekurzivně dělena do menších a menších složek. Postup začíná rozdělením věty na část podmětnou a přísudkovou a postupuje dělením těchto složek na podsložky, až dojde k jednotlivým slovům.

Závislostní přístup naproti tomu vezme jednotlivá slova a ta pospojuje závislostními hranami do takzvaného závislostního stromu. Velmi podrobně je problematika gramatik a větné syntaxe popsána v [16].

Kromě toho, že je možné strukturu věty přirozeného jazyka zapisovat pomocí různých typů gramatik, je též možné zapisovat tuto strukturu na různých významových rovinách. Lingvistický výzkum různých jazyků ukazuje vhodnost takového vícevrstvého popisu. Na různých rovinách je totiž možné přehledněji postihnout různé jazykové jevy [15].

Představme nyní tradiční trojici rovin lingvistické anotace, která je pro popis struktury věty používána v projektu PDT (viz oddíl 4.2). Z terminologie projektu PDT zde budeme vycházet, protože většina lingvistických nástrojů, které představíme (oddíl 4.4), buď z tohoto projektu pochází nebo je s ním úzce spojena a tuto terminologii sdílí.

Roviny lingvistické anotace

- Rovina morfologická
  - též morfématická – tvarosloví
  - `m-layer`
- Rovina analytická
  - rovina povrchové syntaxe případně rovina mluvnické stavby věty
  - `a-layer`

- Rovina tektogramatická
  - významová stavba věty nebo též hloubková syntax
  - `t-layer`

4.1 Lingvistické značky

Lingvistické značky rozdělíme do tří kategorií podle roviny lingvistické anotace. Značky morfologické roviny jsou nezávisle přiřazovány jednotlivým slovům. Naproti tomu značky analytické a tektogramatické roviny popisují strukturu věty a jejich značky popisující i vztahy mezi jednotlivými slovy.

Vztah jednotlivých rovin anotace je znázorněn na obrázku 4.1. Následuje stručný popis jednotlivých značek každé roviny. Podrobnější popis lingvistických značek je možné najít například v [18], [19], [20], [21].

4.1.1 Morfologická rovina

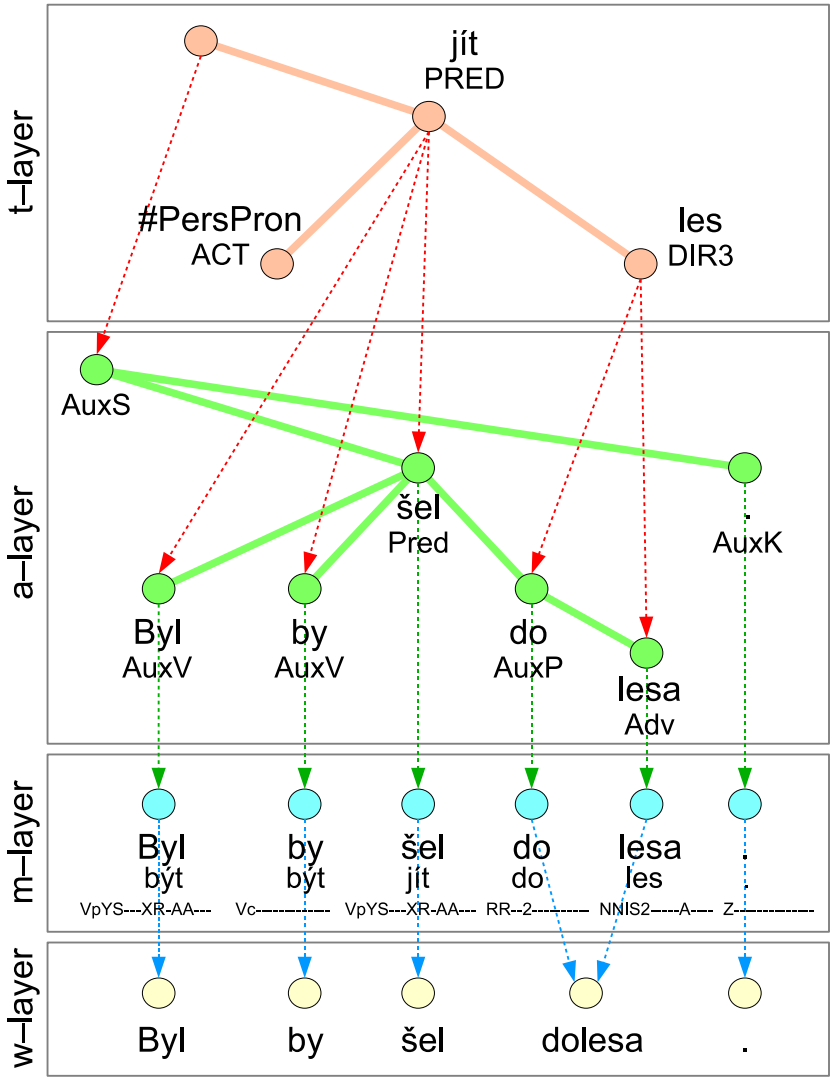
Slovní tvar

Tato značka obsahuje tvar, v jakém se dané slovo vyskytuje v původním textu, včetně zápisu malých a velkých písmen. Od původního výskytu se liší jen ve výjimečných případech, kdy například původní slovní tvar byl číslice s desetinnou čárkou (snaha o jednotný zápis čísel) nebo se jednalo o překlep. Slovní tvar je v korpusech PDT uložen pod atributem *m/form*. Na obrázku 4.1 je slovní tvar uveden v prvním řádku roviny *m-layer* a v popiscích uzlů roviny *a-layer*.

Lemma

Lemma je takzvaný základní tvar slova. Jednoznačně slovo identifikuje. V tomto tvaru je dané slovo obvykle uváděno ve slovnících.

Morfologické lemma je v korpusech PDT uloženo pod atributem *m/lemma*. Na obrázku 4.1 jsou morfologická lemmata ve druhém řádku roviny *m-layer*.



Obrázek 4.1: Roviny lingvistické anotace PDT 2.0 [18]

Příklad na obrázku je převzatý z Průvodce PDT 2.0 [18], znázorňuje vztah mezi sousedními rovinami anotace PDT.

Zobrazená česká věta „Byl by šel dolesa.“ obsahuje minulý čas podmiňovacího způsobu slovesa jít a tiskovou chybu.

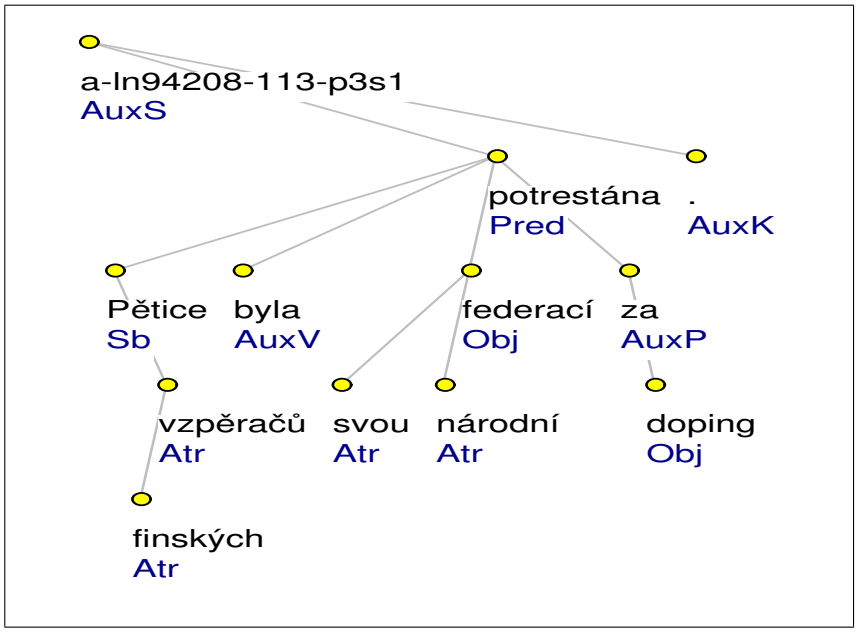
Poslední zobrazená takzvaná *slovní rovina* (w-layer) obsahuje „surový text“, ten je zde rozdělen do dokumentů a odstavců. Jsou tu rozlišeny slovní jednotky (slova, čísla, interpunkce) a jsou opatřeny jednoznačnými identifikátory.

Morfologická značka

Morfologická značka v sobě spojuje informaci o morfologických kategoriích, které dané slovo nese. Z morfologické značky je možné zjistit slovní druh, jmenný rod, číslo, pád, osobu, čas, atd.

V korpusech PDT je možné morfologickou značku najít jako *m/tag*, její hodnotou je patnácti místný řetězec, každý znak nese některou z morfologických kategorií. Například písmeno na první pozici vyjadřuje slovní druh, hodnota `N` (Noun) znamená podstatné jméno, `V` (Verb) sloveso, `R` (Preposition) předložka, atd. Pro podrobnosti viz například [19]. Na obrázku 4.1 jsou morfologické značky v posledním řádku roviny *m-layer*.

4.1.2 Analytická rovina



Obrázek 4.2: Příklad anotace na analytické rovině
Podrobný popis obrázku je v sekci 4.1.4.

Analytická rovina je první úroveň pro strukturní anotaci. Opouští se zde lineární anotace, kdy je každé slovo bráno samostatně bez ohledu na kontext a do anotace textu se zavádí větná struktura. Všechna původní slova textu zůstávají zachována a dostávají ve výsledné struktuře svou funkci.

Na analytické rovině se vytváří stromová struktura věty (stromem rozumíme orientovaný acyklický graf s jedním kořenem). Uzly stromu jsou

tvořeny jednotlivými slovy, respektive tokeny. Hrany stromu reprezentují vztahy závislosti.

Do kořene stromu[2] je umístěno řídící sloveso věty, na toto sloveso se pak zavěšují ostatní slova. V případě, že se jedná o souřadné souvětí, kořenem stromu je spojka, případně čárka, která jednotlivé věty souvětí odděluje. Základním cílem je korektní zachycení struktury věty a označení typu závislosti. Typ závislosti je uložen uvnitř lingvistické značky *analytická funkce*.

Analytická funkce

Analytická funkce je poměrně dobře známý pojem, který se používá na českých základních a středních školách při takzvaném větném rozboru. Tam se ale většinou neoznačuje jako analytická funkce ale jako *větný člen*.

V závislostním stromu označujeme analytickou funkcí jednotlivé hrany stromu, analytická funkce vyjadřuje typ závislosti, kterou hrana znázorňuje.

V korpusech PDT je hodnota analytické funkce uložena pod atributem *a/afun* uvnitř závislého uzlu. Na obrazcích 4.2 a 4.1 jsou zkratky analytických funkcí zapsány ve druhém řádku pod každým uzlem. Následují příklady analytických funkcí spolu se svými zkratkami užívanými v PDT, úplný seznam je možné nalézt například v [20].

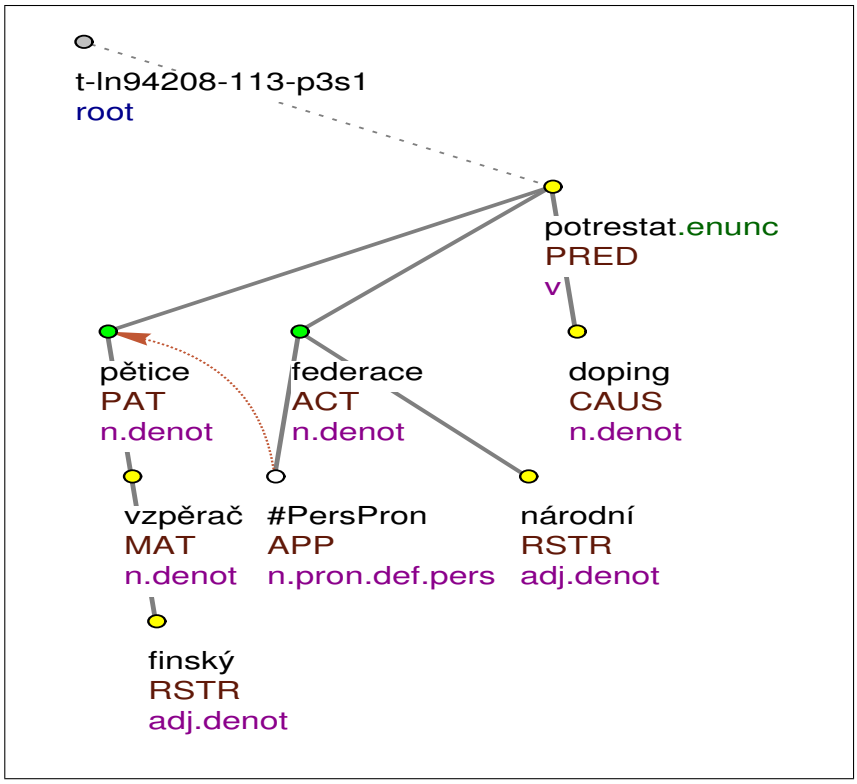
Predikát	přísudek	`Pred`
Subjekt	podmět	`Sb`
Objekt	předmět	`Obj`
Atribut	přívlastek	`Atr`
Adverbiale	příslovečné určení	`Adv`
...		

4.1.3 Tektogramatická rovina

Tektogramatická rovina je poměrně rozsáhlý koncept s hlubokou lingvistickou teorií v pozadí. Byla popsána už v roce 1961 v článku [17].

Tektogramatická rovina slouží k zachycení významové struktury věty. Struktura reprezentace zůstává stejná jako na analytické úrovni, avšak některé uzly se vypouští, některé se přidávají a struktura věty může být obecně

[2]Kořenem zde myslíme skutečný kořen závislostního stromu. V anotacích PDT se z administrativních důvodů používá ještě takzvaný *technický kořen*, který je otcem skutečného kořene věty.



Obrázek 4.3: Příklad anotace na tektogramatické rovině
Podrobný popis obrázku je v sekci 4.1.4.

jiná, než na analytické úrovni. Na obrázku 4.1 je vidět, jak se tři uzly (`Byl`, `by`, `šel`) analytické rovny „smrsknou“ do jediného uzlu (`jít`) tektogramatické roviny. Naproti tomu t-uzel `#PersPron`, který vyjadřuje činitele děje, je do tektogramatického stromu přidán bez vazby na nižší roviny anotace.

U vět, které připouštějí více různých významů (víceznačné věty), je teoreticky možné vytvořit více tektogramatických stromů. V případě synonymie může naopak různým větám odpovídat tentýž tektogramatický strom. Tedy zatímco na morfologické rovině jsou každému slovu věty přiřazeny jeho *lema* a *tag* (morfologická značka) a na analytické rovině uzel analytického stromu a *analytická funkce*, tektogramatická rovina už tento těsný vztah k povrchovému zápisu věty nemá.

Uzly tektogramatické roviny v sobě nesou informaci rozdělenou do několika atributů. Základními atributy uzlu tektogramatického stromu jsou *tektogramatické lema*, *gramatémy* a *funktor*. Vztah mezi uzly tektogramatické a analytické roviny (který je obecně typu M:N), je též zachycen v atributech uzlů tektogramatického stromu. Následuje popis jednotlivých atributů.

Tektogramatické lemma

Tektogramatické lemma (t-lemma) zachycuje lexikální význam uzlu. U jednoduchých uzlů odpovídá lemmatu, které bylo řídícímu slovu tektogramatického uzlu přiřazeno na morfologické rovině. Uzlům na tektogramatické rovině nově vytvořeným je přiřazeno zástupné t-lema speciálního tvaru.

V korpusech PDT je tektogramatické lemma uloženo jako atribut *t_lemma* uzlů tektogramatického stromu. Na obrázcích 4.1 a 4.3 jsou tato lemmata vytištěna na prvním řádku popisků jednotlivých tektogramatický uzlů.

Sémantický slovní druh a jeho podskupiny

Uzly tektogramatického stromu (respektive jejich řídící slova) se rozdělují do takzvaných *sémantických podskupin slovního druhu*. Toto dělení začíná rozdělením uzlů podle takzvaných *sémantických slovních druhů*. Z původních deseti slovních druhů, které v češtině rozlišujeme, vzniknou čtyři sémantické slovní druhy: sémantická substantiva, sémantická adjektiva, sémantická adverbia a sémantická slovesa. Tato se pak dále dělí do sémantických podskupin. Například sémantická substantiva se dělí na *pojmenovací*, *pronominální* a *kvantifikační*. Podrobně je celé toto rozdělení popsáno v [21].

V korpusech PDT je sémantický slovní druh se svými podskupinami zapsán v tečkové notaci uvnitř atributu *gram/sempos* uzlů tektogramatického stromu. Na obrázku 4.3 je sémantický slovní druh a jeho podskupiny vytištěn na třetím řádku popisků.

Gramatémy

Gramatémy jsou tektogramatickým rozšířením morfologických značek. Gramatémy nalezneme pouze mezi atributy uzlů, u kterých to má smysl, tedy u uzlů, které se vztahují k nějakému významovému slovu věty. Navíc různým slovním druhům lze přidělit jen některé gramatémy (například nemá smysl určovat slovesný čas u podstatného jména). Podle toho, do které sémantické podskupiny slovního druhu daný uzel patří, je možné určit, které gramatémy pro něj mají smysl a které nikoli.

V korpusech PDT jsou gramatémy uloženy v několika atributech tektogramatických uzlů. Tyto atributy začínají prefixem *gram/*. Například *gram/tense* (slovesný čas), *gram/negation* (slovo se vyskytuje v negaci – slova `byl` a `nebyl` mají stejné lemma `být` ale různou hodnotu atributu *gram/negation*), *gram/verbmod* (slovesná modalita – oznamovací, rozkazovací, podmiňovací). Výstup automatické lingvistické anotace má většinou vyplněné jen velmi omezené množství tektogramatických gramatémů.

Funktor

Funktory jsou velkým přínosem tektogramatické roviny po praktické stránce. Funktory chápeme jako sémantické ohodnocení hran mezi uzly tektogramatického stromu. Tektogramatické funktory můžeme též chápat jako ekvivalent analytických funkcí. Rozdíl mezi tektogramatickými funktory a analytickými funkcemi je v tom, že funktory se snaží postihnout sémantiku vztahu, zatímco analytické funkce se zaměřují na jeho syntaktickou roli.

V korpusech PDT je funktor uložen uvnitř atributu *functor* závislého uzlu. Na obrázcích 4.1 a 4.3 jsou funktory vytištěna ve druhém řádku popisků.

Následuje popis některých důležitých funktorů, vždy s několika příklady jejich výskytu ve větě. Vyčerpávající seznam je možné nalézt například v [21].

- Funktor ACT (actor) označuje původce děje, nositele děje nebo vlastnosti.
  - Její *manžel.*ACT tam však pracuje dál.
  - Ten *román.*ACT mě oslovil.
  - Českým *skokanům.*ACT se dařilo dobře.
  - Je *mi.*ACT smutno.
- Funktor ADDR (addressee) odpovídá roli příjemce děje.
  - Dal *dítěti.*ADDR hračku.
  - Učí *děti.*ADDR angličtinu.
  - Obrátil se na *soud.*ADDR s problémem.
- Funktor PAT (patiens) označuje předmět dějem zasažený.
  - Snědl *polévku.*PAT
  - Neubližujte *zvířatům.*PAT
  - Učil se *kominíkem.*PAT
  - Mít dost *peněz.*PAT

- Funktor MANN (manner) vyjadřuje, hodnotí způsob provedení děje.
  - Pracuje *pomalu.*MANN
  - *Nějak.*MANN to uděláme.
  - *Prudce.*MANN se zvýšily mezibankovní úrokové míry.
- Funktor TWHEN (temporal : when) vyjadřuje časové určení odpovídající na otázku "kdy?".
  - *Zítra.*TWHEN má být už hezky.
  - *Hned.*TWHEN se vrátím.
  - Součástka se *časem.*TWHEN opotřebuje.
- Funktor LOC (locative) označuje místo, do kterého je děj nebo stav vyjádřený řídícím slovem lokalizován.
  - Zůstaň *doma.*LOC
  - *Nalevo.*LOC stál pěkný dům.
  - *Místy.*LOC ležel v ulicích ještě sníh.
- Funktor DIR1 (directional: from) vyjadřuje určení místa odpovídající na otázku "odkud?".
  - Přijel z *Prahy.*DIR1
- Funktor DIR2 (directional: which way) vyjadřuje určení místa odpovídající na otázku "kudy?".
  - Jdou *lesem.*DIR2
- Funktor DIR3 (directional: to) vyjadřuje určení místa odpovídající na otázku "kam?".
  - Přišel *domů.*DIR3
- Funktor RSTR volně doplňuje blíže specifikující řídící substantivum.
  - *velký.*RSTR dům
- Funktor CONJ (conjuction) je kořen souřadné struktury (tektogramatického podstromu), která reprezentuje spojení dvou a více obsahů.
  - Jezte ovoce *a.*CONJ zeleninu.

4.1.4 Příklady

Pro ilustraci předkládáme dva obrázky závislostních stromů, které vznikly ruční anotací věty:

> *Pětice finských vzpěračů byla svou národní federací potrestána za doping.*

Na obrázku 4.2 je strom analytické roviny a na obrázku 4.3 je strom tektogramatické roviny. Za povšimnutí stojí rozdíl v počtu uzlů obou stromů, který je na tektogramatické rovině o něco menší. Oba tyto příklady pocházejí z PDT 2.0 - sample data. Obrázky jsou vygenerované pomocí editoru TrEd (viz 4.4.2).

Kromě struktury jednotlivých stromů v této kapitole popisovaných jsou na obrázcích vidět i další, především technické jevy, spojené s konkrétní reprezentací lingvistické anotace. Například tektogramatické funktory a analytické funkce nejsou přiřazeny k hranám stromu, ale k závislému uzlu. Každý strom má navíc takzvaný *technický kořen*, na kterém teprve skutečný kořen lingvistického stromu visí. Technický kořen nese administrativní atributy stromu (například atribut *id* – jednoznačný identifikátor). U analytického stromu též stojí za povšimnutí umístění tečky na konci věty, která je potomkem umělého kořene. Uzly obou stromů jsou zleva doprava uspořádány podle pořadí, v jakém se slova odpovídající daným uzlům vyskytují v původní větě.

Schéma analytického stromu je jednodušší. Pod každým uzlem jsou vytištěny hodnoty dvou atributů (hodnoty dvou lingvistických značek). První atribut obsahuje původní *tvar slova* a v druhém je zkratka *analytické funkce* tohoto uzlu.

Schéma tektogramatického stromu je o něco složitější. Pod každým uzlem jsou vytištěny hodnoty tří atributů. První atribut obsahuje *lemma*, prostřední tektogramatický *funktor* a poslední vyjadřuje *sémantickou podskupinu slovního druhu*, do které řídící slovo tohoto uzlu patří. Podrobnosti o těchto atributech a jejich hodnotách je možné nalézt v [21].

Na obrázku 4.3 (tektogramatického stromu) je vidět speciální zahnutá šipka od uzlu *#PersPron* k uzlu *pětice*. Konec této šipky ukazuje na cíl, na který odkazuje zájmeno (`svou`) uvnitř prvního (počátečního) uzlu. Tento jev lingvisté označují jako *koreference*. V automaticky generovaných anotacích PDT koreference nejsou zachyceny. Strojové vyhodnocování lingvistických koreferencí je prezentováno v projektu Artequakt (viz sekce 3.5.2).

4.2 The Prague Dependency Treebank

Pražský závislostní korpus (PDT) je probíhající projekt Centra počítačové lingvistiky Ústavu formální a aplikované lingvistiky (ÚFAL) v Praze[3].

Náplní projektu je především ruční lingvistická anotace velkého množství českých textů. Projekt se vyznačuje velkou hloubkou anotace, která sahá až po tektogramatickou rovinou. Kromě velkého množství anotovaných textů bylo v souvislosti s projektem vyvinuto i množství užitečných nástrojů pro práci s anotacemi a nástroje, které umožňují automatickou lingvistickou anotaci českého textu.

Historie projektu PDT začíná v roce 1995. Od té doby se korpus PDT rozšířil až na současnou (PDT 2.0) velikost 2 milióny slov s provázanými anotacemi na úrovni morfologie (2 milióny slov), povrchové syntaxe (1,5 mil. slov) a hloubkové syntaxe a sémantiky (0,8 mil. slov). Poprvé byl korpus PDT (verze 0.5) veřejně představen v roce 1998. V roce 2001 bylo publikováno CD-ROM PDT 1.0, které obsahovalo přibližně 1,5 mil. slovních jednotek anotovaných na analytické rovině.

V roce 2006 byla publikována poslední současná verze korpusu jako CD-ROM PDT 2.0 [22]. Korpus PDT je v této verzi rozšířen o tektogramaticky anotovaná data. CD-ROM PDT 2.0 dále obsahuje množství kvalitních lingvistických nástrojů (viz 4.4) a publikací, mezi které patří obsáhlý manuál (více než 1000 stran) pro tektogramatické značkování [21].

4.3 Jazyky pro zápis lingvistických anotací

Nyní krátce rozvedeme, v jakých formátech se lingvistické anotace uchovávají. Jedná se většinou o formáty vzniklé pro potřeby korpusu PDT a příbuzných nástrojů. Podrobnosti[4] o těchto formátech je možné nalézt v Průvodci PDT 2.0 [22].

4.3.1 CSTS - Czech Sentence Tree Structure

Formát zvaný CSTS, založený na SGML[5], byl hlavním formátem dat v PDT 1.0. Nyní (v PDT 2.0) je používán jen jako přechodný formát pro kompatibilitu se staršími nástroji pro zpracování jazyka (taggery, parsery, ...). CSTS

[3] http://ufal.mff.cuni.cz/

[4] http://ufal.mff.cuni.cz/pdt2.0/doc/data-formats/

[5] http://www.w3.org/MarkUp/SGML/

může reprezentovat jen morfologickou a analytickou anotaci, není schopen plného popisu tektogramatické roviny.

4.3.2 PML - The Prague Markup Language

Hlavním formátem dat v PDT 2.0 je formát nazvaný PML. PML je založený na XML, je navržený pro reprezentaci bohaté lingvistické anotace textů, jako jsou morfologické značkování, závislostní stromy apod. V PML se mohou jednotlivé oddělené roviny anotace překrývat a mohou být konzistentně propojeny jak mezi sebou, tak i s dalšími zdroji dat. Každá rovina anotace je popsána v souboru PML schéma, který je jakousi formalizací abstraktního anotačního schématu pro tu konkrétní rovinu anotace.

Anotace PDT 2.0 je rozdělena do čtyř rovin, naskládaných jedna na druhou. Každá z těchto rovin má vlastní PML schéma a zpravidla se ukládá do zvláštního soubotu. Jedná se o tyto čtyři roviny: rovina slovní (soubory *.w*), rovina morfologická (soubory *.m*), rovina analytická (soubory *.a*) a rovina tektogramatická (soubory *.t*). Pro podrobnosti o rovinách lingvistické anotace viz 4.1. Vztah jednotlivých rovin PDT je znázorněn na obrázku 4.1.

Další informace je možné nalézt na stránkách PML projektu[6], případně v publikaci [27].

4.3.3 FS - Feature Structure

Formát FS („feature structure“) je formát souborů pro reprezentaci stromů, jejichž uzly jsou struktury atribut-hodnota. Může být chápán jako „meta formát“, podobně jako SGML nebo XML. Konkrétní použití tohoto formátu je plně specifikováno deklarací atributů v hlavičce FS souboru. Formát FS byl primárně vytvořen pro vyhledávací program Netgraph (viz 4.4.1).

4.3.4 PLS - Perl Storable Format

Čistě z důvodů optimalizace a časové úspory se při práci s nástroji TrEd a btred požívá formát PLS. Nástroje TrEd a btred jsou založeny na Perlu, při načítaní dat ve formátu PML a převodu PML dat do vnitřní paměťové reprezentace Perlu se spotřebuje mnoho času. Této časově náročné transformaci se lze vyhnout využitím formátu PLS (Perl Storable Format). Jde o binární datový formát, který přímo odráží vnitřní paměťovou reprezentaci

[6] http://ufal.mff.cuni.cz/jazz/PML/

dat v Perlu. Jeho ukládání a zpětné načítání nástroji TrEd a btred je tedy mnohem rychlejší.

4.3.5 Konverze mezi formáty PDT

Problém s konverzí mezi formáty lingvistické anotace je v tom, že všechny formáty nemohou nést přesně stejné množství informací. Přesto jsou v projektu PDT 2.0 zahrnuty skripty pro konverzi některých formátů:

- konverze analytické anotace typu PDT 1.0 do PML
- konverze a-dat PML do CSTS
- konverze m-dat PML do CSTS
- konverze dat PDT 2.0 do FS pro Netgraph
- konverze dat PDT 2.0 do PLS

4.4 Lingvistické nástroje

Nyní popíšeme některé lingvistické nástroje, které mohou pomoci při zpracování textů přirozeného jazyka a extrakci informací z nich. Většina těchto nástrojů je vyvíjena na Ústavu formální a aplikované lingvistiky (ÚFAL). Tyto nástroje je možné (kromě tektogramatického analyzátoru 4.4.4) získat jako součást PDT 2.0 CD-ROM [22].

4.4.1 NetGraph

Netgraph je aplikace typu klient-server, která umožňuje prohledávat korpus podobný PDT (anotace mají strukturu závislostního stromu) současně několika uživateli, připojenými přes internet. Netgraph je navržený tak, aby prohledávání bylo co nejjednodušší a intuitivní, při zachování vysoké síly dotazovacího jazyka. Funkčnost aplikace je rozdělena na část, kterou vykonává klient, a na část, kterou vykonává server.

Netgraph klient je napsán v Javě a je nezávislý na platformě. Existuje ve dvou formách - jako samostatná Java aplikace a jako Java applet. Applet verze je oproti plné Java aplikaci ochuzena o některé funkce, přesto však poskytuje plnou vyhledávací sílu. Funkce klienta zahrnuje vytvoření (návrh)

dotazu, jeho odeslání serveru a zobrazení, případně další zpracování výsledků vrácených serverem.

Netgraph server je napsán v C a C++ a běží v operačním systému Linux i dalších systémech - podrobnosti je možné nalézt v [24]. Umožňuje nastavit uživatelská konta s různými přístupovými právy. Korpus, určený k prohledávání Netgraphem, musí být ve formátu FS (viz 4.3.3). Funkce serveru spočívá ve vyhodnocování dotazů zasílaných klienty. Server prohledává korpus a stromy, které vyhovují dotazu, vrací jako odpověď.

Dotazy v Netgraphu jsou definovány pomocí vlastního dotazovacího jazyka. Jedná se o jazyk formálně velmi jednoduchý, avšak s vysokou expresivitou. Definovat dotaz v Netgraphu, znamená definovat podstrom, který se má v prohledávaných stromech vyskytovat. Tedy v dotazu můžeme definovat požadovanou strukturu stromu. Navíc můžeme v každém uzlu dotazu vynutit hodnotu některých atributů tohoto uzlu.

Velmi jednoduchý dotaz, kdy chceme vyhledat všechny stromy obsahující slovo „hasič“ se skládá z jediného uzlu a restrikce na hodnotu atributu *lemma* (viz 4.1.1) na hodnotu *hasič* v tomto uzlu.

Podrobnosti o možnostech a syntaxi tohoto dotazovacího jazyka je možné nalézt například v [24]. Poznamenejme ještě, že dotazy mohou být dále rozšířeny tzv. meta atributy, které umožňují určení pozice dotazu v nalezených stromech, omezení velikosti nalezených stromů, určení vztahů mezi hodnotami atributů u různých uzlů v nalezených stromech, negaci a mnoho dalších podmínek.

V průběhu experimentu, který je popsán v kapitole 6, jsme narazili na potřebu dotazovacího jazyka, pomocí kterého bychom se mohli programově dotazovat na hodnoty atributů lingvistických stromů. Též by se nám velmi hodil jazyk, ve kterém by bylo možné vyjádřit vzory stromů, které se v korpusu našeho experimentu častěji vyskytují. Dotazovací jazyk, který používá aplikace Netgraph, je naší představě velmi blízký. Tuto problematiku podrobněji rozebíráme v oddíle 7.1.

Dotazy se v Netgraph klient vytvářejí v uživatelsky přívětivém grafickém prostředí. Uživatel si zde může „naklikat“ celý strom, který se má při vyhodnocování dotazu hledat. V grafickém rozhraní Netgraph klient má uživatel snadný přístup k možnostem dotazovacího jazyka, aniž by musel tento jazyk podrobně znát.

Další informace o aplikaci Netgraph je též možné nalézt na její domovské stránce[7].

[7] http://quest.ms.mff.cuni.cz/netgraph/

4.4.2 Tree Editor TrEd

Tree Editor TrEd je velmi komplexní grafický editor, který umožňuje rychlé, pohodlné a flexibilní procházení, prohlížení a úpravu stromů v korpusech podobným PDT. TrEd prvotně sloužil jako hlavní anotační nástroj PDT, ale může být použit i k prohlížení dat a obsahuje několik druhů vyhledávacích funkcí. TrEd se vyznačuje svými možnostmi nastavení a přizpůsobení celé aplikace na míru potřebám nejrůznějších uživatelů. Silný nástroj představují uživatelská makra, která mohou být do aplikace kýmkoliv doprogramována v jazyce Perl. TrEd podporuje velké množství vstupních a výstupních formátů dat, jmenujme například FS, CSTS, PDT-PML (podrobnosti k jednotlivým formátům – viz 4.3).

TrEd je možné nainstalovat na většině v současné době používaných operačních systémů: Linux, UNIX (MacOS X, BSD, Solaris, ...) i Windows (funguje díky ActivePerl for Windows, který musí být v systému nainstalovaný). Na domovské stránce[8] editoru TrEd lze získat jednotlivé instalační balíčky i podrobné instrukce pro instalaci na konkrétní operační systém.

Ukázkové obrázky

Na obrázcích 4.2 a 4.3 jsou schémata stromů získaná přímo z editoru TrEd. Takto jsou při výchozím nastavení v TrEd editoru zobrazovány analytické a tektogramatické stromy.

btred / ntred

Součástí editoru TrEd jsou též dva nástroje - *btred* a *ntred*, které umožňují automatické (dávkové) zpracování stromů korpusu. Ntred je pouze rozšířením nástroje btred o možnost zpracovávat korpus paralelně více počítači v síti najednou.

Tyto nástroje se spouštějí přímo z příkazového řádku, nemají grafické rozhraní. Činnost těchto nástrojů je řízena uživatelským programem (makrem btred-u), které uživatel musí napsat v programovacím jazyce Perl. Při psaní toho makra má uživatel k dispozici velké množství specializovaných funkcí pro práci se strukturami korpusu: s jednotlivými stromy, s uzly stromů, s atributy uzlů, snadno se řeší přechod mezi jednotlivými rovinami lingvistické anotace, atp.

[8] http://ufal.mff.cuni.cz/~pajas/tred/

Přehledný a dobře srozumitelný návod – „btred/ntred tutorial“[9], jak pracovat s nástroji btred a ntred je možné nalézt na stránkách editoru TrEd.

4.4.3 Tools for machine annotation - PDT 2.0

Jedná se o skupinu nástrojů, které provádějí plně automatickou lingvistickou analýzu českého textu. Ze surových českých vět vytvářejí závislostní stromy na analytické rovině. Proces anotace se skládá z následujících funkcí.

1. Rozpoznání slovních jednotek ve vstupním surovém textu a rozdělení textu na věty.
2. Morfologická analýza a tagging (morfologická desambiguace).
3. Závislostní parsing.
4. Přiřazení analytických (závislostních) funkcí všem uzlům zparsovaného stromu.

Tyto funkce jsou implementovány celkem v šesti oddělených nástrojích. Vstupem každého nástroje je vždy výstup předchozího s výjimkou prvního, jehož vstupem je prostý text. Nástroje jsou napsány z části v Perlu, zbytek tvoří přeložený kód (C++) pro Linux běžící na i386 architektuře.

Celý řetěz nástrojů se dá spustit jediným skriptem *run_all*.

Tyto nástroje a jejich podrobný popis[10] (včetně naměřené chybovosti) jsou k dispozici jako součást PDT 2.0 CD-ROM [22].

Následuje podrobnější popis jednotlivých nástrojů.

Segmentation and tokenization

Provádí rozdělení textu na slova a interpunkční znaménka (tokenizace) a rozdělí tyto tokeny do vět (segmentace).

Morphological analysis

Pro každé slovo vyhledá všechna možná lemmata a morfologické značky, která by mu mohly odpovídat.

[9] http://ufal.mff.cuni.cz/~pajas/tred/bn-tutorial.html
[10] http://ufal.mff.cuni.cz/pdt2.0/doc/tools/machine-annotation/

Morphological tagging

Ze všech možných alternativ získaných v předchozím kroku pro každé slovo vybere jedno lemma a morfologickou značku. Tento proces se často nazývá desambiguace. Tagging pro Češtinu je poměrně zajímavý vědecký problém, který je podrobně rozpracován v mnoha publikacích[11].

Parsing

Morfologicky označkovaná slova v každé větě uspořádá do závislostního stromu. Problém automatického závislostního parsingu[12] je stále poměrně živý. Aktuálně nejlepší parser [23] dosahuje přesnosti přibližně 86%

Analytical function assignment

Jednotlivým hranám závislostního stromu, které vznikly v předchozím kroku, přiřadí funktory analytické roviny. Nástroj pracuje jako klasifikátor založený na rozhodovacím stromu. Řídící rozhodovací strom byl vytvořen pomocí Quinlanova C5 klasifikátoru z dat PDT 1.0.

Conversion into PML

Zapíše výstup předchozího nástroje v PML jazyce. Pro podrobnější informace o PML viz oddíl 4.2.

[11] http://ufal.mff.cuni.cz/czech-tagging/
[12] http://ufal.mff.cuni.cz/czech-parsing/

4.4.4 Nástroj pro tektogramatickou analýzu češtiny

Jedná o nástroj, který provádí automatickou tektogramatickou lingvistickou anotaci. Jako vstup akceptuje na analytické rovině anotovaná data uložená ve formátu PML. Tedy dokáže výstup nástrojů výše (4.4.3 – Tools for machine annotation) povýšit na tektogramatickou rovinu.

Nástroj je založený na strojovém učení, pro které byl použit nástroj *fnTBL toolkit*[13] [26]. Pro češtinu bylo učení realizováno na datech PDT 2.0. Podrobnosti o algoritmu a jeho úspěšnosti je možné nalézt v článku [25].

Autorem tohoto nástroje je Václav Klimeš[14]. U něho je možné tento nástroj získat společně s dalšími informacemi a instrukcemi pro instalaci. Poslední verze (rok 2007) byla určena pro operační systém Linux.

[13] http://nlp.cs.jhu.edu/~rflorian/fntbl/index.html
[14] http://ufal.mff.cuni.cz/~klimes/

Kapitola 5

WordNet

WordNet [28], [29] je lexikální databáze vybudovaná na základě psycholexikologického výzkumu o lidské lexikální paměti. Jazykové jednotky nejsou ve WordNetu uspořádány abecedně, ale podle jejich sémantických vztahů, tedy hierarchicky a shlukově. Tento typ lexikální databáze se často označuje jako *sémantická síť*.

Už jednoduché mapování slov na jejich významy uložené jako *synsety* WordNetu (viz dále) se někdy označuje jako sémantická analýza textu (například v [32]). V experimentu, který je součástí této práce, jsme technologii WordNet chtěli použít k zobecnění metody pro extrakci dat z textu. Podrobnosti o tomto experimentu je možné nalézt v kapitole 6.

WordNet jakožto sémantická síť obsahuje pouze slova, která nesou nějaký kognitivní význam, tedy podstatná jména, přídavná jména, příslovce a slovesa. Navíc jsou ve WordNetu obsažena i slovní spojení (sousloví), například „vysoká škola“. V dalším textu si však pro lepší přehlednost dovolíme zjednodušení a o všech těchto jazykových výrazech, které můžeme ve WordNetu nalézt budeme mluvit jako o slovech.

Lexikální matice

Základním formálním prostředkem pro zachycení významu slova je *lexikální matice*. Řádky této matice tvoří jednotlivé významy, sloupce jednotlivá slova. Záznam lexikální matice na souřadnicích $[i\ ,\ j]$ znamená, že slovo j nese význam i. Pokud se objeví dva záznamy na stejném řádku, znamená to, že odpovídající dvě slova mají stejný význam, jsou synonymní. Pokud se naopak objeví více záznamů v jednom sloupci, znamená to, že toto slovo nese více možných významů, je polysémické.

Synset

Záznamy ve WordNetu jsou organizovány podle významu, tedy podle řádků lexikální matice. Každý takový řádek ve WordNetu označujeme jako *synset* (množina synonym nebo též synonymická řada). Synset je pro WordNet tím, čím je heslo pro obyčejný slovník.

Jelikož různé slovní druhy nemohou být synonymy v pravém slova smyslu, je sémantická síť WordNetu budovaná pro každý slovní druh zvlášť.

Číslování významů - literály

Jednotlivé prvky synsetů označujeme jako *literály*. Literál reprezentuje jeden záznam v lexikální matici, tedy dvojici slovo-význam. Literál budeme chápat jako slovo v daném významu.

Literály resp. významy daného slova se ve WordNetu číslují. Například anglické podstatné jméno *bank:1* označuje finanční instituci, *bank:2* – břeh.

Sémantické vazby

Wordnet obsahuje celou řadu sémantických vazeb mezi literály a zejména mezi synsety. Vzniká tak síť slov, tedy WordNet. Popišme nyní tyto vazby podrobněji.

- *Hyperonymie a hyponymie* jsou vztahy významové nadřazenosti a významové podřízenosti. Například *flanel* je druhem *textilie*. Vztahy hyperonymie a hyponymie vytvářejí základní hierarchickou strukturu WordNetu pro podstatná jména. V zásadě se jedná o stromovitou strukturu, kde blíže ke kořenu stromu znamená obecnější a blíže k listům znamená specifičtější. Tomuto vztahu se někdy též říká *lexikální dědičnost*. Příklad stromu lexikální dědičnosti je na obrázku 5.1.

- *Meronymie a holonymie* vyjadřují vztah mezi celkem a částí. Tedy například slovu *dům* je slovo *okno* meronymum a *město* holonymum.

- *Antonymie* vyjadřuje sémantickou protikladnost dvou synsetů. Například slova *mokrý* a *suchý* jsou antonymická.

5.1 Princeton WordNet

Duchovním otcem WordNetu je George A. Miller z univerzity v Princetonu. Zde je též pod Millerovým vedením stále vyvíjen a rozšiřován první a současně největší (americký) *Princeton WordNet*[1] (PWN). Současná verze WordNet 3.0 obsahuje 207 016 literálů (párů slovo-význam) v 117 597 synsetech. Data PWN jsou princentonským týmem poskytována volně.

5.2 EuroWordNet

Cílem projektu EuroWordNet[2] [30] bylo vytvořit WordNety pro další evropské jazyky a provázat je do multilingvální databáze.

Tento projet začal v roce 1997. V první fázi byly zpracovány jazyky: britská angličtina, holandština, italština a španělština, ve druhé pak čeština, estonština, francouzština a němčina. V roce 2001 byla tato činnost ještě rozšířena projektem BalkaNet[3] o dalších pět balkánských jazyků (bulharštinu, rumunštinu, řečtinu, srbštinu a turečtinu).

Velká snaha byla věnována co možná nejúplnějšímu provázání významů napříč různými jazyky. Společným podkladem všem novým slovníkům byl PWN 1.5. V něm každý synset dostal jednoznačný identifikátor. Tyto identifikátory sloužily pro vytváření ekvivalencí mezi synsety PWN a ostatních jazyků. Tak vznikl takzvaný mezijazykový index (Inter-Lingual Index, ILI).

Dalším vylepšením nových WordNetů bylo rozšíření počtu sémantických relací v rámci jednoho jazyka, tzv. Inter-Lingual Relations (ILR).

- Přibyla relace *near synonym* respektive *similar to*, která je určena k propojení literálů a synsetů, jejichž význam je sice blízký, avšak o úplná synonyma se nejedná.
  Například *krásný – pěkný – líbivý – pohledný.*

- Dále vznikl soubor relací, které propojují synsety *napříč slovními druhy.* Tyto relace jsou užitečné pro zachycení slovotvorných vztahů. Vytvářejí se tak shluky slov odvozených od stejného slovního základu.
  Například *učit – učitel – učitelský.*

[1] http://wordnet.princeton.edu
[2] http://www.illc.uva.nl/EuroWordNet/
[3] http://www.ceid.upatras.gr/Balkanet/

5.3 Český WordNet

Český WordNet [31] začal pod vedením doc. Karla Paly vznikat v roce 1998 na Fakultě informatiky Masarykovy univerzity v Brně[4] v rámci druhé fáze projektu EuroWordNet. V současné době obsahuje český WordNet přibližně 30 000 synsetů.

Online interface k českému WordNetu je dostupný přes internet, přístupný po domluvě podmínek s vedoucím projektu doc. Karlem Palou[5]. K dispozici je webové rozhraní a jednoduché dobře dokumentované programátorské API[6], jehož prostřednictvím má programátor přístup ke všem funkcím online databáze WordNet.

V rámci projektu DEB II je vyvíjen i vizuální prohlížeč a editor online WordNetu DEBVisDic, který je volně k dispozici na stránkách[7] projektu.

5.3.1 SAFT - Semantic Analyzer of Free Text

Zajímavý experiment s českým WordNetem provedl Tomáš Čapek ve své práci [32]. V této práci představuje nástroj SAFT - Semantic Analyzer of Free Text. Vstupem tohoto nástroje je text přirozeného jazyka, k slovům vstupního textu SAFT vyhledává jejich významy ve WordNet databázi.

Experiment spočíval ve spuštění tohoto nástroje na část českého korpusu DESAM (vyvinutého na Fakultě informatiky MU Brno). Ukázalo se, že přibližně 50% slov není v českém WordNetu zastoupeno vůbec, avšak uvědomíme-li si, že WordNet pokrývá pouze výrazy nesoucí kognitivní význam (tedy podstatná jména, přídavná jména, příslovce a slovesa), není tento výsledek tak špatný. Autor dokonce tvrdí, že většina nenalezených slov patří právě do kategorie slov bez kognitivní sémantiky.

[4] http://www.fi.muni.cz
[5] http://www.muni.cz/fi/people/Karel.Pala
[6] http://nlp.fi.muni.cz/trac/deb2/wiki/WordNetApi
[7] http://nlp.fi.muni.cz/projekty/deb2/

- entita:1
  - objekt:1
    - celek:1
      - artefakt:1, výtvor:2, výrobek:2
        - vybavení:2
          - přepravní prostředek:1, transportní prostředek:1
            - **veřejná doprava:1**
              - *autobus:1, autokar:1*
            - **dopravní prostředek:1**
              - kolové vozidlo:1
                - samohybné vozidlo:1, vozidlo s vlastním pohonem:1
                  - motorové vozidlo:1
                    - nákladní automobil:1
                      - *kamion:1*

Obrázek 5.1: Příklad stromu lexikální dědičnosti v českém WordNetu pro slova kamion a autobus

5.4 Kritika WordNetu

WordNet bývá často kritizován z různých důvodů. Tradiční lexikografové vidí mnoho problémů v nejasně definované (a v čase se měnící) koncepci tvorby hesel (synsetů), ve kterých panuje značný nepořádek [4] (synonyma nejsou přesnými synonymy, hyponyma jsou nestejnorodá, klasifikace nenavazují na běžné oborové klasifikace [33]).

V našem experimentu (viz kapitola 6) s českým WordNetem jsme se potýkali s nedostatečným pokrytím české slovní zásoby. Provázání synsetů sémantickými hranami je v českém WordNetu též poměrně řídké. Například nejbližší společné hyperonymum pro slova *autobus* a *kamion* není synset *motorové vozidlo:1* ani *dopravní prostředek:1* ale až synset *přepravní prostředek:1, transportní prostředek:1*. Na obrázku 5.1 je vidět stromu lexikální dědičnosti českého WordNetu, který tuto situaci ilustruje.

Kapitola 6

Experiment

Původním záměrem experimentu v této práci bylo vyzkoušet některou existující metodu automatické sémantické anotace. Při hledání vhodné metody jsme zjistili, že ve většině projektů je algoritmická dokumentace použitých metod velmi stručná (alespoň dokumentace, která je veřejně k dispozici). Získat nějaký použitelný kód nebo knihovnu bylo problematické. Výjimku tvoří projekty KIM (popsaný v sekci 3.5.6) a GATE (sekce 3.5.5). Avšak zaměření těchto dvou projektů příliš nevyhovovalo požadavkům na náš experiment.

Přestavba architektury projektu KIM by byla poměrně náročná a její výsledek nejistý. Pravděpodobně by vznikl pouze slabší „bratr“ bulharského KIM, který je podpořen masivní databází informací v pozadí.

Projekt GATE je naproti KIM otevřený a modulární. Jedná se o velmi obecnou softwarovou základnu pro lingvistickou anotaci a extrakci informací, která nabízí mnoho možností, k dispozici je velké množství funkčních modulů. Ale k čemu přesně bychom GATE chtěli použít? Ponechali jsme tedy GATE jako otevřenou možnost a začali přesněji hledat a specifikovat problém, který budeme řešit.

Paralelně s touto prací jsme mohli sledovat vývoj prací Dušana Maruščáka a Roberta Novotného !!!citovat!!!. V těchto pracích se pokoušejí o anotaci / extrakci informací ze strukturovaných web-stránek. Tento přístup se často označuje jako konstrukce *wrapper-u*. Obě tyto práce se opírají o zajímavý nápad využití opakujících se struktur uvnitř stránek. Vzniklé metody jsou pak téměř nezávislé na konkrétní podobě vstupní stránky. Avšak pevná (HTML) struktura stránky podmiňuje použití těchto metod.

Začali jsme si uvědomovat, mezeru v oblasti extrakce informací z přirozeného textu v Českém jazyce. Nejsou nám známy výsledky žádné práce,

která by se tímto tématem zabývala. Přitom česká počítačová lingvistika je na velmi vysoké úrovni.

Otevřela se nám možnost spolupráce s Martinem Labským a Vojtěchem Svátekem na části projektu The RAINBOW Project[1]. Konkrétně bychom se zde zabývali rozšířením jejich systému pro extrakci informací pomocí „extrakční ontologie“. Jedná se o propracovaný systém založený na široké paletě extrakčních pravidel, která jsou definovaná v extrakční ontologii. Jde však také především o konstrukci wrapper-u. Jedna z možností naší spolupráce měla spočívat v rozšíření palety pravidel jejich projektu o pravidla založená na lingvistice.

K žádné spolupráci zatím nedošlo, ale v experimentu této práce se pokoušíme otestovat dostupné nástroje pro lingvistickou anotaci českých textů a prozkoumat možnosti jejich využití pro extrakci informací a automatickou sémantickou anotaci.

6.1 Osnova prací

Postupem prací v experimentu se snažíme simulovat postup opravdového projektu, který by byl zaměřený na dodatečnou automatickou sémantickou anotaci některých zdrojů webu. Tedy jsme v situaci, kdy chceme nějakým způsobem využít data na webu publikovaná. Abychom mohli tato data použít, potřebujeme je získat v takové formě, aby se dala strojově zpracovávat a vyhodnocovat. Na webu jsou však tato data publikována tak, aby si je mohli prohlížet obyčejní lidští návštěvníci, nemají strukturu, kterou požadujeme.

Data která nás zajímají[2] mohou být vyjádřena přirozeným jazykem v textech článků několika webových portálů. V našem experimentu se jedná o články hasičského zpravodajství z českých regionů na portálu MVČR (podrobněji v oddíle 6.2). Abychom se od těchto článků dostali k datům, která potřebujeme, budeme postupovat podle následující osnovy, která je graficky znázorněna na obrázku 6.1.

6.1.1 Příprava vstupních dat

První, co musíme udělat, je stáhnout požadované články z internetu k dalšímu zpracování. Programy, které se zabývají touto činností nazýváme *web crawler*. V našem experimentu jsme naprogramovali velmi jednoduchý web

[1] http://rainbow.vse.cz/

[2] Podrobnější diskuse o vstupních datech experimentu je v oddíle 6.2.

Obrázek 6.1: Schéma aplikace

crawler, který využívá kanál RSS[3] publikovaný na portálu MVČR a stáhne všechny web-stránky s články, které nás zajímají.

Nyní potřebujeme ze stažených článků extrahovat text, který budeme analyzovat. V případě hasičských článků to nebyl velký problém. Stačil jednoduchý skript, který pomocí několika regulárních výrazů oddělil text článku od HTML struktury web-stránky. Při našem druhém pokusu s daty evidence úpadců ČR (viz 6.2.2) jsme narazili na problém se specifickými formáty textu (především formát DOC). Automatické zpracování těchto dat by bylo implementačně náročné a kladlo by přemrštěné časové nároky. Pro potřeby experimentu jsme tato data zpracovali v malém rozsahu ručně.

Nesnadným problémem je obecná automatizace předchozích dvou procedur. Například pro stahování „zajímavých“ článků by se dal použít univerzální web crawler nějakého internetového vyhledávače, tím je například Egothor[4] vyvíjený týmem Lea Galamboše. Stránky, které tento crawler stahuje, by se filtrovaly pomocí heuristiky. Ta by vybrala stránky, které má cenu dále zpracovávat. Následuje problém, jak na stránce automaticky najít texty,

[3]RSS – RDF Site Summary bývá označováno [4] jako technologie sémantického webu.
[4]http://www.egothor.org/

které nás zajímají. Pravděpodobně by se i tento problém dal řešit pomocí nějaké heuristiky s podporou metod pro konstrukci wrapper-u. Poznamenejme ještě, že Egothor kromě HTML zpracovává i zdroje ve formátech PDF, PS, DOC a XLS.

Poslední transformace, kterou jsme před lingvistickým zpracováním textů provedli, byl převod kódování českých znaků, překlad znakových entit HTML (`  &amp;` ...) a sjednocení zápisu časových údajů (10:45 → 10.45, dvojtečku považoval lingvistický analyzátor za oddělovač slov, zatímco časový údaj zapsaný s tečkou vyhodnocuje správně). Tyto transformace se dají snadno automatizovat, pouze při převodu kódování českých znaků musíme správně určit originální kódování zdroje.

Podrobnosti o implementaci této fáze našeho experimentu je možné nalézt v sekci 6.3.2.

6.1.2 Lingvistická anotace

Tato fáze představuje převod prostých textů na strukturovaná data lingvistických anotací. V současné době nemáme na výběr moc možností jak tuto fázi realizovat. Kromě lingvistických analyzátorů PDT, existují ještě nástroje vyvíjené na Masarykově univerzitě v Brně, o nich se zmiňuje například práce [32]. Bylo by jistě zajímavé v experimentu obě varianty porovnat, ale už samotné poskládání nástrojů PDT bylo organizačně poměrně náročné, realizace téhož s brněnskou stranou by práci časově protáhla a domníváme se, že by pro tuto práci nebyla „převratným“ přínosem. Tento potenciální přínos ale nepopíráme.

V našem experimentu se nyní nacházíme v situaci, kdy máme texty, které chceme analyzovat uložené v prostých textových souborech ve správném kódování (ISO 8859-2). Spustíme jejich automatickou lingvistickou anotaci, která se skládá z řetězu nástrojů Tools for machine annotation - PDT 2.0 (viz sekce 4.4.3) a nástroje pro tektogramatickou analýzu (sekce 4.4.4). Po delší době (lingvistická anotace je časově poměrně náročná) získáme výstup v podobě lingvistických anotací na všech rovinách popisu PDT, uložených ve formátu PML i PLS. Kvalita automaticky generovaných lingvistických anotací se různí věta od věty. Domníváme se však, že pro typ aplikací, který zde simulujeme, je kvalita anotací více-méně dostačující.

Programová realizace této fáze experimentu je popsaná v sekci 6.3.2.

6.1.3 Extrakce dat

V této fázi se budeme snažit pomocí struktury lingvistických anotací extrahovat data obsažená v původních textech. Popíšeme zde „řešení“, které jsme zvolili. Jedná se ale spíš o průzkum než o nějakou ucelenou metodu. Pří popisu tohoto řešení se budeme snažit komentovat další alternativy a možnosti.

XML nebo btred?

Nejprve jsme se chtěli s daty blíže seznámit. V tomto bodě jsme museli provést první rozhodnutí, totiž jestli bude pro naše účely vhodnější na programové úrovni pracovat s lingvistickými daty přímo nebo prostřednictvím nástroje btred (popsaný v sekci 4.4.2).

Lingvistická data ve formátu PML jsou uložena jako XML poměrně složité struktury. Manipulace s XML je v současné době podpořena širokou paletou programových nástrojů a nepředstavuje pro programátora velkou překážku, ale předpokládá detailní znalost struktury zpracovávaných dat.

Btred, který nám byl lingvisty doporučován, nabízí mnoho užitečných funkcí pro manipulaci s PML daty, tři práci s nástrojem btred tedy programátor nepotřebuje tak podrobné znalosti formátu PML, programátor ale musí zvládnout funkce tohoto nástroje. Jedinou možností, jak s nástrojem btred pracovat, je naprogramování vlastního btred-makra, které pak tento nástroj nad lingvistickými daty vyhodnotí. Tato makra se zapisují v jazyce Perl, který je v oblasti softwarových systémů zřídka používaný. Integrace btred-makra se zbytkem softwarového systému může být komplikovanější.

K počátečnímu průzkumu jsme zvolili druhou variantu – btred. Pro komplexní projekt by ale bylo vhodné tuto volbu ještě zvážit a porovnat s možnostmi vytvoření a použití dotazovacího jazyka nad lingvistickými stromy, viz dále.

Frekvenční analýza

Jako základní pohled na data nám posloužila frekvenční analýza uzlů v lingvistických stromech, konkrétně analýza tektogramatických lemmat, zvláště pak její podmnožina omezená pouze na slovesa. Výsledky těchto analýz jsou uloženy v souborech[5] *freq.txt* a *freq_verb.txt* SVN repository.

[5]Výsledky frekvenčních analýz jsou uloženy v souborech freq.txt a freq_verb.txt pro každý datový zdroj zvlášť, konkrétně v adresářích *data/hasici* a *data/upadci*.

Pravidla pro extrakci dat

Díky frekvenčním analýzám jsme se mohli zaměřit na ta tvrzení, která se v textech často vyskytují. Vizuální prohlídka jednotlivých vět nám pak umožnila vypozorovat ve větách jednoduché vzory typu: Na slovese `zranit` visí pod funktorem `PAT` podstrom věty, který blíže specifikuje osoby, jich počty, a druh zranění, které utrpěly a tento podstrom má opět ve většině případů podobnou strukturu. Na základě těchto pozorování je možné zkonstruovat deterministická programová pravidla pro extrakci dat. O konstrukci několika takových pravidel jsme se pokusili.

Ukázalo se, že naprogramování extrakčního pravidla pomocí nástroje btred představuje i pro velmi jednoduchý vzor mnoho práce. Přesto, že jsme se snažili program strukturovat do většího množství obecnějších funkcí a podprocedur, byl zápis pravidla velmi nepřehledný. Tento závěr není překvapivý, více-méně jsme ho předpokládali. Díky tomuto pokusu jsme ale podrobněji poznali úskalí, která konstrukce těchto pravidel přináší.

Zápis pravidel

Programování extrakčních pravidel čistě pomocí nástroje btred nám ukázalo výhody, které by přinesl formální dotazovací jazyk nad lingvistickými stromy. Tento jazyk by dále umožnil formalizovat pravidla pro extrakci dat a formální zápis těchto pravidel by umožnil jejich uživatelskou editaci, strojovou indukci i sdílení. Bohužel žádný takový přímo použitelný dotazovací jazyk pro lingvistické stromy neexistuje. V oddíle 7.1 jsme se pokusili o návrh takového jazyka a v oddíle 7.2 se zamýšlíme nad možnostmi strojové indukce pravidel pro extrakci dat.

Využití WordNetu

Už v zadání této práce je zmínka o možnosti využít databázi WordNet. Tato možnost se nabízí právě zde. Pomocí lexikální sítě WordNetu by bylo možné zobecnit extrakční pravidla. Například tam, kde bychom v původním extrakčním pravidle (bez WordNetu) požadovali přesnou hodnotu tektogramatického lemma, bychom mohli povolit i jeho libovolné synonymum. Na jiném místě bychom mohli požadovat libovolné hyponymum případně libovolný prvek podstromu lexikální dědičnosti, například ve zprávách o dopravních nehodách bychom libovolné *motorové vozidlo* nalezli v podstromu lexikální dědičnosti tohoto spojení.

Při zkoumání českého WordNetu jsme bohužel zjistili nedostatečné pokrytí české slovní zásoby a poměrně řídké provázání *synsetů* sémantickými

hranami. Zvláště patrné je to, pokud se zaměříme na nějakou specifickou oblast, jako jsou například motorová vozidla. Nepovažujeme tedy za přínosné současný český WordNet přímo na extrakční pravidla napojit. Na druhou stranu se nemusíme vzdávat zobecnění, které by sémantická lexikální síť přinesla a můžeme na základě WordNetu a dalších obdobných zdrojů takovou síť vytvořit. Tato síť může být úzce specializovaná na doménu ve které se pohybujeme, nemusí zdaleka dosahovat rozsahu a plné obecnosti, která je na WordNetu pozoruhodná.

Podrobnosti o programových nástrojích, které jsme pro zkoumání českého WordNetu vyvinuly je možné nalézt v sekci 6.3.4.

Shrnutí

V praktickém experimentu jsme zjistili, že extrakci dat z lingvisticky anotovaných textů je možné provést pomocí extrakčních pravidel. Podrobnosti o pravidlech, která jsme programově realizovali předkládáme v sekci 6.3.3. Vytvoření takových funkčních pravidel je ale se současnými programovými prostředky zbytečně pracné a málo účelné. Efektivní návrh a využití extrakčních pravidel by umožnil dotazovací jazyk a jeho interpret nad lingvistickými stromy. Návrh takového jazyka předkládáme v oddíle 7.1.

Další možností, jak extrahovat data z lingvisticky anotovaných textů, je zapojení některé metody strojového učení. Pro tyto metody bychom musely vytvořit trénovací data. Vytvoření takových dat představuje mnoho ruční práce, která však může být srovnatelná s prací nutnou k návrhu extrakčních pravidel. Za nejlepší možnost považujeme poloautomatické vytvoření extrakčních pravidel z opakujících se vzorů ve zkoumaných datech. Diskuse nad možnostmi indukce takových vzorů je v oddíle 7.2.

6.1.4 Formální reprezentace dat

Nacházíme se nyní v situaci, kdy se nám podařilo z lingvisticky anotovaných textů extrahovat data, která nás zajímají. Chtěli bychom je uchovávat v takové formě, která by vystihovala jejich sémantiku. Využijeme tedy nějaký konceptuální formální popis (ontologii), pomocí kterého data zapíšeme. Vznikne tak interpretace těchto dat pomocí slovníku zvolené ontologie. Libovolný programový nástroj, který „porozumí“ dané ontologii, bude moci s těmito daty pracovat.

Technicky není tato fáze náročná. Jedná se o jednoduchou datovou transformaci, kterou je například pro XML možné realizovat pomocí XSLT. V na-

šem experimentu jsme tuto fázi programově nerealizovali, důvodem byl mimo jiné nedostatek extrahovaných dat.

Náročnost této fáze spočívá v nalezení, případně vytvoření vhodné ontologie, pomocí které budeme data interpretovat. Je potřeba zvážit všechny možné případy, ve kterých by se data dala použít a najít takové řešení, které bude ve většině případů vyhovovat. Konkrétně použít rozšířenou ontologii, případně maximálně usnadnit mapování použité ontologie na koncepty ostatních. Pokud se podaří konceptuálně správně data zachytit, vznikne skutečně sémantická anotace těchto dat, která není závislá na účelu jejich použití.

Extrakční vzory vzniklé v předchozí fázi nejsou závislé na vstupních datech, jsou závislé pouze na jazyce (na Češtině). Navíc tektogramatický popis se snaží rozpouštět rozdíly mezi jednotlivými jazyky. Při překladu lemmat (například pomocí WordNetu a ILI – viz oddíl 5.2) by se tyto vzory mohli stát též na jazyce nezávislé. Ovšem za předpokladu, že bychom dokázali tektogramaticky analyzovat i ostatní jazyky a zdokonalili současný EuroWordNet. Mohla by tak vzniknout databáze lingvisticko-sémantických vzorů, která by se dala sdílet a rozšiřovat v širokém spektru projektů.

6.2 Vstupní data

Dlouho jsme hledali vhodný zdroj dat, na kterém bychom pomocí experimentu ukázali výhody lingvistického přístupu k extrakci informací a sémantické anotaci. Potřebovali jsme zdroj, kde jsou data vyjádřena přirozeným jazykem ve volném textu. K tomu, aby naše automatická metoda mohla ukázat nějaký přínos proti prostému manuálnímu přepisu dat, potřebujeme, aby se v textech opakovali informace podobného typu a aby byly podobně vyjádřeny. V textech díky tomu můžeme najít vzory opakujících se tvrzení a pomocí nich hromadně extrahovat data, která tato tvrzení nesou.

Tyto podmínky splňuje širší spektrum zdrojů. Uvažovali jsme o anotaci analytických reportů o výsledcích dataminingu, o článcích otevřené encyklopedie Wikipedia, o zprávách ze sportovních zápasů i o výročních zprávách podniků ČR. Výhody sémanticky anotovaných reportů zmiňujeme už v úvodu (sekce 1.1.2). V případě Wikipedia encyklopedie bychom naše sémantické anotace mohli ukládat pomocí SMW (viz sekce 3.5.1) a přispívat tak k vytvoření sémantické Wikipedie. V tomto případě bychom se ale museli zaměřit na nějakou užší oblast článků.

Nakonec jsme vybrali dva poměrně odlišné zdroje: hasičské zpravodajství a databázi úpadců ČR. Hlavními důvody tohoto rozhodnutí byla vysoká míra

opakování se podobných témat ve zprávách a poměrně snadná dostupnost těchto dat. Oba zdroje podrobněji popíšeme níže.

Další poněkud odlišnou možností vstupních dat představuje korpus PDT. V tomto případě bychom se mohli opřít o velmi kvalitní „ruční“ lingvistické anotace. Data korpusu PDT jsou pestrá a opakující témata bychom zde hledali obtížně. Navíc důraz experimentu byl kladen na co možná největší přiblížení se k podmínkám a problémům skutečného projektu. V takovém případě bychom se těžko mohli opřít o to, že by nám data která chceme analyzovat někdo ručně lingvisticky anotoval. Nicméně pokusy s ručními lingvistickými anotacemi dat PDT[6] (především z tréninkových důvodů) proběhly.

6.2.1 Hasiči

Ministerstvo vnitra České republiky pravidelně zveřejňuje krátké zprávy o akcích, kterých se účastnily hasičské sbory jednotlivých regionů ČR. Tyto zprávy se nám z výše zmíněných důvodů zdály vhodné jako zdrojová data experimentu. Tyto články jsou k dispozici na internetovém portálu MVČR jako „Bleskové hasičské zpravodajství RSS z regionů“[7]. Odtud je pomocí programových nástrojů (popsaných v sekci 6.3.2) stahujeme a dále zpracováváme. Aktuálně zpracovávaný archiv obsahuje přibližně 700 článků (celkem cca 1MB textových dat).

Tyto zprávy se nejčastěji týkají výjezdu hasičských oddílů k dopravní nehodě nebo požáru, méně často informují o různých hasičských slavnostech a dalších zásazích hasičské služby (ochrana vodních zdrojů před chemickým znečištěním, opatření proti ptačí chřipce, kuriózní nehody v domácnostech).

6.2.2 Úpadci

Na Informačním serveru českého soudnictví[8] je veřejně dostupná *Evidence úpadců* Ministerstva spravedlnosti České republiky. Tato rozsáhlá a neustále aktualizovaná evidence obsahuje kromě dalších dat i texty jednotlivých soudních rozhodnutí a ustanovení. Právě tyto texty jsme zvolili jako vstupní data experimentu.

Automatické stažení a pročištění dat z tohoto datového zdroje by bylo poměrně komplikované. Texty jsou na serveru uložené ve formátu DOC, k těmto dokumentům neexistují perzistentní URL, stahování jednotlivých

[6] Jedná se o *sample data* PDT 2.0, http://ufal.mff.cuni.cz/pdt2.0/data/sample/
[7] http://www.mvcr.cz/rss/regionhzs.html
[8] http://portal.justice.cz/

dokumentů probíhá přes náhodnou dočasně zvolenou adresu, která vzniká při generování web-stránky s detailním popisem každého konkurzu. Ještě náročnější by však bylo automaticky extrahovat texty, které nás zajímají z jednotlivých DOC souborů. Tyto soubory nemají jednotnou strukturu, jsou napsané různými autory, jsou různě formátované, texty ustálených nadpisů se vyskytují v několika variantách.

Tento datový zdroj hraje v našem experimentu spíše doplňkovou roli. Chtěli jsme porovnat kvalitu lingvistických anotací a možnosti naší metody na datech v další doméně. K tomuto účelu jsme tato data zpracovali v malém rozsahu ručně, jsou uložena v souboru *data/upadci/sample_data.txt* SVN repository.

6.3 Software

Pro ukládání a verzování programové části práce jsme využili služeb veřejného serveru BerliOS[9], kde nám byl poskytnut účet spolu s veřejným SVN repository[10]. V tomto SVN úložišti je možné nalézt aktuální verze softwarových komponent i textů této práce. Na CD-ROM přiloženém k této práci je v adresáři *czsem* umístěna kopie SVN repository ve verzi z 10. 8. Chystáme se však práci dále vyvíjet, proto doporučujeme data v SVN repository na přiloženém CD-ROM před používáním či prohlížením aktualizovat (Pro podrobnosti viz soubor ReadMe.txt na přiloženém CD-ROM).

6.3.1 Instalace

Pro otestování jednotlivých programových částí je nutná poměrně komplikovaná instalace použitých lingvistických nástrojů. K publikaci některých z nich navíc nemáme svolení. Čtenář, který pravděpodobně bude chtít programovou část práce vyzkoušet, má dvě možnosti. Buď si všechny nástroje i s licencí pro jejich používání obstará sám nebo se může obrátit na autory práce a vyžádat si zapůjčení těchto nástrojů za účelem testování této práce – tato činnost pak bude pokryta platnou licencí autorů práce. Druhá varianta zahrnuje i propůjčení uživatelského účtu (včetně hesla) pro přístup k webovému rozhraní českého WordNetu (pro podrobnosti o českém WordNetu viz 5.3).

[9] Naše stránky zde mají adresu http://czsem.berlios.de/

[10] Adresa naší SVN repository: http://svn.berlios.de/svnroot/repos/czsem/trunk/

Postup instalace včetně odkazů na poskytovatele jednotlivých nástrojů je popsaný v souboru *install.txt*[11].

Autoři práce jsou si vědomi toho, že pokud by chtěli svůj software veřejně publikovat nebo dokonce prodávat, bylo by nutné instalaci zjednodušit a ošetřit právní nároky třetích stran. To ale nebylo předmětem tohoto experimentu. Problematiku instalace externích komponent ponechme jejich autorům. Nejméně příjemná je instalace tektogramatického analyzátoru (sekce 4.4.4), tento nástroj je ale stále ve vývoji a jeho instalaci je nutné považovat za dočasné řešení.

6.3.2 Skripty pro přípravu dat

Skripty, pomocí kterých se stáhnou a transformují stránky hasičského zpravodajství jsou umístěné v adresáři *data/hasici* SVN repository. Jedná se o sadu *bash* skriptů, které by měli fungovat na většině UNIX-ových systémů. Před jejich spuštěním je nutné do systému nainstalovat potřebné lingvistické nástroje (pro popis instalace viz 6.3.1). Kompletní dávku spustíme (na pozadí) skriptem *run_parse_background*. Tato dávka postupně vykoná všechny akce popsané v sekcích 6.1.1 a 6.1.2. Výstupní lingvistické anotace budou uloženy v adresáři *data/hasici/pml*. Celý běh této dávky může trvat i více než tři hodiny. Hlášení o průběhu jednotlivých akcí je zaznamenáváno v log-souboru *parse.log*. Podrobnosti k funkci jednotlivých skriptů je možné nalézt v komentářích uvnitř každého skriptu.

6.3.3 Makra pro extrakci dat

V průběhu experimentu jsme vytvořili několik maker pro nástroj btred (sekce 4.4.2). Jednalo se o makra pro frekvenční analýzu, makra pomocí kterých jsme podrobněji zkoumali vstupní data a makra která simulují extrakci dat. Příklad extrakčního pravidla, které jsme se pokusili realizovat je znázorněno v sekci 7.1.1.

Všechny tyto makra (Perl-skripty) jsou umístěny v adresáři *src/perl* SVN repository. Podrobnosti o jednotlivých makrech je možné nalézt v souboru *src/perl/info.txt* a v komentářích uvnitř kódu jednotlivých skriptů.

6.3.4 Hledání příbuzných slov pomocí WordNetu

[11] Soubor install.txt je umístěný v adresáři *docs/UserGuide* SVN repository.

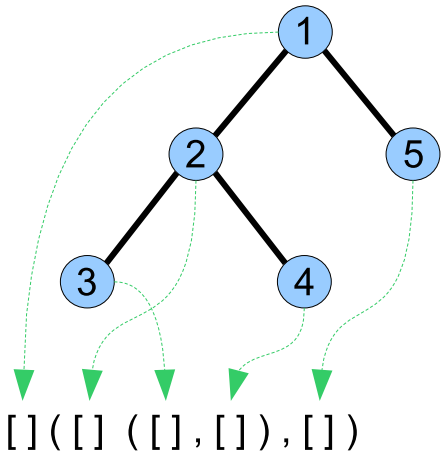
Kapitola 7

Návrhy a zkušenosti

7.1 Dotazování nad lingvistickými závislostními stromy

interpretace - XPath, XPath přehlednější v lineárním zápisu, vizualizace dotazu podobně jako v NG klientu

Skryté uzly v nástroji Netgraph XXX btred a různé roviny, `PML_T::GetANodes($this)`



[]([](□,□),□)

Za každým uzlem následuje v kulatých závorkách čárkami oddělený seznam jeho dětí.

7.1.1 Extrakční pravidla

7.1.2 Indexace

Indexace XML dat (pro optimalizaci vyhodnocování dotazů XPath, XQuery a podobných) je v současnosti otevřeným problémem. Stále ještě hledáme algoritmus, který by dokázal efektivně indexovat XML data libovolné struktury. Tuto situaci ilustrují například články [35], [36].

7.2 Indukce vzorů

klasifikace - dendrogram od frekvenční analýzy lemmat dále po různých osách struktury, různé délky tranzitivních hran stromu, zahrnou různé atributy

konstrukce dendrogramu interaktivní, např. se fixují jednotlivá lemmata

převod vzoru na pravidlo: uživatel označí atributy které ho zajímají, zapíše transformaci výstupu na formální reprezentaci znalostí (4. krok osnovy), lze též vizualizovat ontologii a mapovat vizuálně.

Označí korpus a nasadit strojové učení. .. pravidla by vznikala jako černá - až šedá skříňka .. problém s vytvářením korpusu, ruční anotace

7.3 Závěr práce

Otevírá mnoho možností

Základní průzkum
- teoretický
- praktický

Seznam obrázků

Literatura

[1] Tim Berners-Lee, James Hendler, Ora Lassila. The Semantic Web. Scientific American, 2001.

[2] S. Handschuh, S. Staab (edited by). Annotation for the Semantic Web. Volume 96 Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, The Netherlands, 2003. ISBN 1-58603-345-x.

[3] Franz Baader, Werner Nutt. Basic Description Logics. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. F. Patel-Schneider, editors, The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003.
URL http://citeseer.ist.psu.edu/baader03basic.html

[4] P. Matulík, T. Pitner, P. Smrž. Sémantický web a jeho technologie (1,2,3). Zpravodaj ÚVT MU. ISSN 1212-0901, 2004, roč. XIV, č. 3,4,5. 15–17, 9–13, 14–16.
URL http://www.ics.muni.cz/zpravodaj/issues/serials.html#2

[5] Vojtěch Svátek. Ontologie a WWW. Dušan Chalpek (editor). DATAKON 2002, Masarykova univerzita, Brno, 2002. 27-–55.
URL http://nb.vse.cz/˜svatek/onto-www.pdf

[6] Michal Fiedler. Ontology Matching. Diplomová práce, Katedra softwarového inženýrství, Matematicko-fyzikální fakulta, Univerzita Karlova, Praha, 2007.

[7] Lawrence Reeve, Hyoil Han. Survey of Semantic Annotation Platforms. SAC '05: Proceedings of the 2005 ACM symposium on Applied computing, ACM Press, New York, USA, 2005. 1634–1638
URL http://dx.doi.org/10.1145/1066677.1067049

[8] Krotzsch M., Vrandecic D., Volkel M. Semantic MediaWiki. LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER-VERLAG, Germany, 2006. 935–942

[9] Kim S., Alani H., Hall W., Lewis P., Millard D., Shadbolt N., Weal M. Artequakt: Generating Tailored Biographies from Automatically Annotated Fragments from the Web. Proceedings of Workshop on Semantic Authoring, Annotation & Knowledge Markup (SAAKM'02), the 15th European Conference on Artificial Intelligence (ECAI'02), France, Lyon, 2002. 1–6 URL http://eprints.ecs.soton.ac.uk/6913/

[10] M. Klein. Using RDF Schema to interpret XML documents meaningfully. S. Handschuh, S. Staab (editors). Annotation for the Semantic Web, volume 96 of Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, 2003. 79–89.

[11] Julien Carme, Michal Ceresna, Oliver Frölich, Georg Gottlob, Tamir Hassan, Marcus Herzog, Wolfgang Holzinger, Bernhard Krüpl. The Lixto Project: Exploring New Frontiers of Web Data Extraction. Lecture Notes in Computer Science, Volume 4042/2006, Springer Berlin / Heidelberg, 2006. 1–15 URL http://dx.doi.org/10.1007/11788911_1

[12] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, 2002. URL http://www.gate.ac.uk/sale/acl02/acl-main.pdf

[13] Atanas Kiryakov, Borislav Popov, Damyan Ognyanoff, Dimitar Manov, Angel Kirilov, Miroslav Goranov. Semantic Annotation, Indexing, and Retrieval. Elsevier's Journal of Web Semantics, Vol. 2, Issue (1), 2005. URL http://www.ontotext.com/publications/SemAIR_SWJ.pdf

[14] Simone Santini. Image Semantics without Annotations. S. Handschuh, S. Staab (editors). Annotation for the Semantic Web, volume 96 of Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, 2003. 156 –179

[15] E. Hajíčová, M. Plátek, P. Sgall: Komunikace s počítačem v češtině, Sborník referátov seminára SOFSEM 81, Výzkumné výpočtové stredisko Bratislava, 1981. 85–114.

[16] P. Sgall a kolektiv: Úvod do syntaxe a sémantiky, Academia, Praha, 1986.

[17] Curry, H. B.: Some logical aspects of grammatical structure, Structure of Language and Its Mathematical Aspects (red. R. Jakobson), Proceedings of Symposia in Applied Mathematics 12. American Math. Society, Providence, RI 1961.

[18] Jan Hajič, Eva Hajičová, Jaroslava Hlaváčová, Václav Klimeš, Jiří Mírovský, Petr Pajas, Jan Štěpánek, Barbora Vidová Hladká, Zdeněk Žabokrtský. Průvodce PDT 2.0. Prague Dependency Treebank 2.0, CDROM, Linguistic Data Consortium, In press, 2006. URL http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/cz/pdf/pdt-guide.pdf

[19] Daniel Zeman, Jiří Hana, Hana Hanová, Jan Hajič, Barbora Hladká, Emil Jeřábek. A Manual for Morphological Annotation, 2nd edition. Technical Report 27, ÚFAL MFF UK, Prague, Czech Republic, 2005. URL http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/m-layer/pdf/m-man-en.pdf.

[20] Eva Hajíčová, Zdeněk Kirschner, Petr Sgall. A Manual for Analytic Layer Annotation of the Prague Dependency Treebank (English translation). Technical report, ÚFAL, MFF UK, Prague, Czech Republic, 1999. URL http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/pdf/a-man-en.pdf.

[21] Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářova-Řezníčková, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, Zdeněk Žabokrtský. Anotace Pražzského závislostního korpusu na tektogramatické rovině: pokyny pro anotátory [A Manual for Tectogrammatical Layer Annotation of the Prague Dependency Treebank]. Technical report, ÚFAL, MFF UK, Prague, Czech Republic, 2005. URL http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/cz/t-layer/pdf/t-man-cz.pdf.

[22] Jan Hajič, Eva Hajičová, Jaroslava Hlaváčová, Václav Klimeš, Jiří Mírovský, Petr Pajas, Jan Štěpánek, Barbora Vidová Hladká, Zdeněk Žabokrtský. Prague Dependency Treebank 2.0, CDROM, Linguistic Data Consortium, In press, 2006. URL http://ufal.mff.cuni.cz/pdt2.0/.

[23] Daniel Zeman, Zdeněk Žabokrtský. Improving Parsing Accuracy by Combining Diverse Dependency Parsers. Proceedings of the International Workshop on Parsing Technologies (IWPT 2005). Association for Computational Linguistics, Vancouver, British Columbia.

[24] Jiří Mírovský. Netgraph: a Tool for Searching in Prague Dependency Treebank 2.0. Proceedings of The Fifth International Treebanks and Linguistic Theories conference, Prague, Czech Republic, 2006. 211–222.

[25] Václav Klimeš. Transformation-Based Tectogrammatical Analysis of Czech. Proceedings of Text, Speech and Dialogue 2006, Springer-Verlag, Berlin Heidelberg, 2006. ISSN 0302-9743.

[26] Grace Ngai, Radu Florian. TransformationBased Learning in the Fast Lane. Proceedings of NAACL 2001, Pittsburgh, PA, 2001. 40–47.

[27] Petr Pajas, Jan Štěpánek. XML-Based Representation of Multi-Layered Annotation in the PDT 2.0. Proceedings of LREC 2006 Workshop on Merging and Layering Linguistic Information, ELRA, Genoa, Italy, 2006.

[28] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller. Five papers on wordnet. Technical Report CSL Report 43, Cognitive Science Laboratory, Princeton University, 1990.

[29] Christiane Fellbaum (editor). WordNet: An Electronic Lexical Database. Bradford Books, The MIT Press, 1998. ISBN 0-262-06197-x.

[30] Vossen P. EuroWordNet: a multilingual database for information retrieval. In Proceedings of DELOS workshop on Cross-language Information Retrieval, 1997.

[31] Pala K., Ševeček P. The Czech WordNet, final report. Technical report, Masarykova univerzita, Brno, 1999.

[32] Tomáš Čapek. Systém pro částečné sémantické značkování volného textu. Diplomová práce, Fakulta informatiky, Masarykova univerzita, Brno, 2006.

[33] Martin Ph. Correction and Extension of WordNet 1.7. ICCS 2003, 11th International Conference on Conceptual Structures, Springer Verlag, Dresden, Germany, 2003. 160–173
URL http://www.webkb.org/doc/papers/iccs03/

[34] Tomečková M., Rauch J., Berka P. STULONG -– Data from Longitudinal Study of Atherosclerosis Risk Factors. Berka P. (editor). Discovery Challenge Workshop Notes. ECML/PKDD-2002, Helsinki, 2002.

[35] Jongik Kim, Hyoung-Joo Kim. A partition index for XML and semi-structured data. Data & Knowledge Engineering 51, 2004. 349–368

[36] Radim Bača, Michal Krátký, Václav Snášel. Evaluation of Multidimensional Approach to Indexing XML data with Compressed R-trees. Proceedings of Znalosti 2007, Ostrava, Czech Republic, 2007.