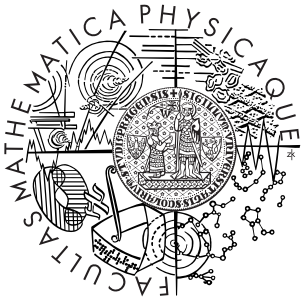
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Martin Kirschner

Konstrukce sémantického slovníku z neanotovaných dat

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. Jiří Semecký, PhD.
IBM Research

Studijní program: Informatika, Obecná informatika

2008

Chtěl bych poděkovat vedoucímu mé bakalářské práce za podnětné myšlenky a nápady na vylepšování této práce. Dále mé přítelkyni Vendule Trávníčkové, která mi pomáhala s klasifikací výstupních dat. A v neposlední řadě i mému bratrovi Janu Kirschnerovi, za technickou pomoc při tvorbě tohoto dokumentu.

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

V Praze dne 29.5.2008 Martin Kirschner

Obsah

1 Úvod **6**
1.1 O sémantických slovnících 6
1.2 Obsah práce 7

2 Metody a postupy při tvorbě sémantických slovníků **8**
2.1 Pomůcky pro ruční vytváření 8
2.2 Strojově tvořené slovníky 9
2.2.1 MindNet 9
2.2.2 Semantix 9

3 Popis metody **11**
3.1 Teoretické podklady 11
3.2 Implementace 12
3.2.1 Výběr kontextu 12
3.2.2 Získávání vztahů 13
3.3 Analýza dat 13
3.4 Zhodnocení výsledků 14

4 Uživatelská dokumentace **15**
4.1 Semantic-xml 15
4.2 Semantix 16
4.2.1 Popis instalace a konfigurace 16
4.2.2 Ovládání programu 17
4.2.3 Druhy sémantických vztahů 17
4.3 Semantic Comparator 19
4.4 Semantic-XML Browser 20

5 Programátorská dokumentace **21**
5.1 Semantix 21
5.1.1 Běh programu 21
5.1.2 Třídy a datové struktury 22
5.1.3 Použité algoritmy 23
5.2 Semantic Comparator 25
5.3 Semantic-XML Browser 25
5.3.1 Struktura databáze 25
5.3.2 Algoritmy a datové struktury 26

6 Závěr **28**

Literatura **29**

A Obsah přiloženého CD **31**

Název práce: Konstrukce sémantického slovníku z neanotovaných dat
Autor: Martin Kirschner
Katedra (ústav): Ústav formální a aplikované lingvistiky MFF UK
Vedoucí bakalářské práce: RNDr. Jiří Semecký, PhD.
e-mail vedoucího: jiri.semecky@mff.cuni.cz

Abstrakt – V předložené práci jsou zkoumány možnosti konstrukce sémantických slovníků z neoanotovaných dat, tedy z prostého textu. Jsou zde porovnánány výhody a nevýhody několika možných přístupů anotace. Blíže je rozpracováno jedno řešení na základě latentní sémantické analýzy (LSA), které na rozdíl od ostatních metod řeší problém polysemie a synonymie. Dále jsou zde uvedeny podrobné detaily implementace a vyhodnocení. Výsledkem je sada nástrojů pro vytváření, převádění a prohlížení sémantických slovníků v českém jazyce. Použité postupy nejsou závislé na jazyce, pro který jsou implemantované.

Klíčová slova: sémantický, slovník, konstrukce, neanotovaný, lsa

Title: Unsupervized construction of semantic lexicon
Author: Marin Kirschner
Department: Institute of Formal and Applied Linguistics
Supervisor: RNDr. Jiří Semecký, PhD., IBM Research
Supervisor's e-mail address: jiri.semecky@mff.cuni.cz

Abstract – In present work are studied possibilities of unsupervized constuction of semantic lexicons. There are compared advantages and disadvantages of several annotation methods. More closely developed is one solution, which is based on latent semantic analysis (LSA). Unlike the others, this method solves the problems of polysemy and synonymy. Below are described details of the algorithm implementanion and evaluation. This thesis provides a set of tools for construction, browsing and conversion of semantic lexicons for czech language. Used methods do not depend on the language.

Keywords: semantic, lexicon, construction, unsupervized, lsa

Kapitola 1

Úvod

Na úvod jsou zde přiblíženy možnosti užití sémantických slovníků a přístupy k jejich vytváření. Blíže se tu také seznámíte s obsahem práce a s programy, které jsou její součástí.

1.1 O sémantických slovnících

Sémantický slovník může být velkým rozšířením a vylepšením aplikací, které pracují s přirozeným jazykem. Například textové vyhledávače, strojový překlad, nebo některé okruhy umělé inteligence. Úspěšnost aplikace využívající sémantický slovník je ale závislá na kvalitě a velikosti jejího slovníku. Nejspolehlivější slovníky jsou vytvářené ručně, například WordNet (5) nebo CyC (11), ty ale pokrývají spíše obecné platné skutečnosti a nejsou tedy v porovnání s korpusem obsáhlé. Navíc je práce na nich velmi náročná jak časově, tak finančně.

Pro ruční vytváření lze použít nástroje, které zpracovávají korpus a pak nabízejí slova, která s největší pravděpodobností budou v sémantickém vztahu s přidávanými lematy. Například přístup představený v (8) je založen na postupném rozšiřování skupin sémanticky příbuzných slov na základě blízkosti v určitých syntaktických strukturách. Tyto pomůcky sice zvyšují produktivitu práce anotátora, ale nabízí se další řešení.

Strojově tvořený slovník výše zmíněné nevýhody nemá. Práce je rychlá a levná, navíc výsledek obsahuje více termínů a vztahů mezi nimi. Tím, že existuje nástroj, který slovník vytváří, vzniká možnost použít jej na různá jazyková odvětví a tím získat různé specializované soubory termínů a sémantických vztahů. Velkou nevýhodou takto vytvořených slovníků ale je, že obsahují značné procento chyb, takže se na jejich obsah nedá spoléhat. Příkladem strojově tvořeného slovníku je projekt Microsofu MindNet prezentovaný v (9).

1.2 Obsah práce

Hlavním cílem této práce je navrhnout a vyzkoušet metodu získávání sémantických vztahů z prostého textu, založenou na latentní sémantické analýze (LSA) viz (10) pro český jazyk. Následující kapitola představuje metody, které lze použít pro tvorbu sémantických slovníků. Kapitola Implementované řešení představuje kombinaci metod a způsob jejich využití k automatizované tvorbě sémantických slovníků z prostého textu.

Tato práce také prezentuje sadu nástrojů na tvorbu, prohlížení a porovnávání sémantických slovníků. Funkcionalita je rozdělená do tří aplikací.

První je Semantix, program na vytváření a spojování sémantických sovníků. Tento program zpracovává data ve formátech prostý text, tektogramaticky anotovaný text ve formátu PML, WordNet a XML podle definice semantic-xml. Výstupem Semantixu je XML podle normy semantic-xml.

Dalším programem je SemanticComparator, který buď porovnává dva semantic-xml soubory, nebo dovoluje uživateli postupně určovat platnost vztahů ve vstupním souboru. Tento nástroj se používá hlavně na hodnocení úspěšnosti strojového získávání sémantických vztahů z prostého textu.

Posledním programem je Semantic-XML Browser. To je webová aplikace napsaná v technologii asp .NET, která umožňuje procházení a hledání cest v sémantickém slovníku vytvořeném aplikací Semantix.

Jednotlivými částmi všech výše zmíněných programů se budeme zabývat zvlášť v kapitolách Uživatelská a Programátorská dokumentace.

Kapitola 2

Metody a postupy při tvorbě sémantických slovníků

V úvodu byly zmíněny možné přístupy ke tvorbě sémantických slovníků. V Této kapitole jsou blíže popsané používané metody s jejich výhodami i nevýhodami. Tato část bude výchozím bodem pro následují oddíl, kde je detailně popsané implementované řešení z teoretického hlediska.

2.1 Pomůcky pro ruční vytváření

Pro ruční vytváření sémantických slovníků se často používají pomocné nástroje, které anotátorovi nabízí slova nějak sémanticky příbuzná s anotovaným slovem, nebo slovní kategorií. Jeden takový nástroj je popsaný v (7) a rozšířený v (8). Tato metoda je založená na rozšiřování sémantického okruhu zadaných slov na základě společného výskytu v určitých syntaktických konstrukcích. Například slova vyskytující se společně v seznamech (*lvi, tygři, medvědi*), slova která jsou souřadně spojená (*lvi a tygři a medvědi*), přívlastky (*lev, král zvířat*) nebo složené názvy (*trojklanný nerv*).

Výše popsaná metoda zjistí z textu okruh slov, ale už nezjistí, jaké jsou mezi nimi závislosti. Proto lze použít v této formě jen jako pomůcka a ne jako automatický anotátor. Nicméně úspěšnost tohoto způsobu dokazuje, že je možné získávat sémantické vztahy z určitých syntaktických konstrukcí, a že se prvky stejné sémantické kategorie mohou vyskytovat blízko sebe. V mírně pozměněné podobě byl tento přístup využit v první fázi získávání vztahů v implementované metodě.

2.2 Strojově tvořené slovníky

Plně automatická tvorba sémantických slovníků má, jak již bylo zmíněno v úvodu, mnoho výhod, jako jsou rychlost, nízké náklady a univerzálnost. Vyvinuté metody automatické anotace se dají použít i na různé jazyky, pro které je ruční anotace z důvodu omezených prostředků těžko dostupná.

Na druhou stranu mají takto vytvořené slovníky nižší spolehlivost obsažených relací. Je tedy potřeba hledat a zkoušet další metody k odstranění, nebo přinejmenším snížení této nevýhody.

2.2.1 MindNet

Již zmiňovaný projekt MindNet (9), má vysokou spolehlivost relací, přestože je vytvářen strojovou anotací. Anotovány jsou totiž výkladové slovníky, které mají jednodušší stavbu věty, takže se v nich nevyskytuje tolik víceznačných nebo jinak složitých konstrukcí, jako v běžném textu.

Popis každého hesla slovníku je rozpársován pomocí speciálního parseru způsobem popsaným v (6). Další vztahy jsou získány převrácením této struktury. Pomocí sémantického pársování je možné získat podobné vztahy, jako z tektogramatických funktorů, proto mohou být druhy relací používané v semantixu inspirované těmi v MindNteu viz tabulka 4.1 níže a (9).

Z tohoto druhu anotace textu se nedá přímo zjistit synonymie, nebo polysemie slov. Nelze tedy vytvořit slovník stejného typu jako WordNet. Ten neobsahuje přímo vztahy mezi jednotlivými lematy, ale mezi jejich skupinami se stejným významem.

2.2.2 Semantix

Semantix je jeden z programů prezentovaných v této práci. Jeho součástí je i hlavní předmět této práce, strojová anotace prostého textu. V této sekci ale bude popsána pouze část, která získává vztahy z tektogramaticky anotované vrstvy pražského závislostního korpusu (PDT) (13).

Tektogramaticky anotované věty (15) obsahují pouze plnovýznamová slova, spojená ohodnocenými hranami do stromů. Hrany jsou označené tektogramatickými funktory, z nichž některé odpovídají sémantickým vztahů. Slova spojená těmito funktory jsou semantixem překódována do vztahů popsaných níže. Viz 4.1.

Takto vytvořený slovník je sice komplexní v tom smyslu, že obsahuje jevy z přirozeného jazyka, tedy má širší pokrytí, než ručně anotovaný slovník. Je však svým rozsahem

limitován na velikost tektogramaticky anotovaného korpusu. Sice již existuje nástroj na automatickou tektogramatickou anotaci (2), ale úspěšnost strojové anotace z prostého textu se pohybuje těsně nad 75%.

Navíc takto získané vztahy, stejně jako v MindNetu, vedou pouze mezi lematy, ne mezi významy slov. Nastává tu tedy opět problém s polysemií a synonymií. Jiný přístup, který řeší tyto problémy je popisován v následující kapitole.

Kapitola 3

Popis metody

V této kapitole je podrobně popsaná implementovaná metoda získávání sémantických vztahů z prostého textu, se zaměřením na řešení probému synonymie a polysemie. Výstupem níže popsaného postupu jsou většinově synonymické, hyperonymické a meronymické relace. Postup lze ale po menších úpravách aplikovat na data získaná jinými metodami a doplnit rozlišení významů slov do skupin.

3.1 Teoretické podklady

Hlavním cílem popisované metody je vypořádat se polysemií a synonymií slov. Polysemie je jev, kdy jednen zápis slova má několik významů. Synonymie je naproti tomu jev, kdy více různých slov má stejný nebo podobný význam. Shlukováním slov podobného významu je tvořena struktura WordNetu. Proto bude pro vytváření slovníků tohoto druhu důležité vyřešit hlavně problém polysemie a rozpoznání synonymie.

Pokud má konkrétní slovo v textu více významů, dá se poznat zamýšlený pouze podle kontextu, ve kterém se nachází. Chceme-li rozlišit jednotlivé významy, musíme mít možnost sumarizace a porovnání jejich kontextů. Důkazem, že na kontextu záleží, je třeba metoda popsaná v (7) a zmíněná výše v části zabývající se pomůckami pro ruční vytváření slovníků.

Možnost sumarizace a porovnánání kontextů nám nabízí metoda LSA (10), původně navržená na klasifkaci dokumentů pomocí v nich obsažených specifických termínů. Na začátku je každému slovu přiřazen vektor, jehož každá složka odpovídá počtu výskytů daného slova v jednom dokumentu. Význam každého slova je tak zastoupen vektorem v sémantickém prostoru. Složením těchto vektorů vznikne matice incidence slov a dokumentů.

Dokument jako kontext slova je seznam termínů, které se vyskytují v jeho blízkosti (v rámci jednoho dokumentu). Obecněji je možné brát kontext slova jako skupinu slov jemu blízkých v textu. Například v rámci jedné věty. Pomocí LSA lze tedy klasifikovat zvlášť i části dokumentů, nebo samotná slova.

LSA provádí SVD (Singular Value Decomposition), což je rozklad na součin tří matic, z nichž prostřední je diagonální a na diagonále má singulární hodnoty původní matice. Odebráním rozměrů, kterým odpovídají menší hodnoty se z matice odstraní šum získaný s daty. Šum v datech tady odpovídá výkladům významů slov, které jsou platné jen za omezených podmínek. Součin nových tří matic má stejné rozměry, jako původní matice. Směr vektorů už ale není ovlivněn odstraněnými řídkými významy a tak se k sobě přibližují sémanticky příbuzná slova a tvoří skupinky. Velikost těchto skupin slov záleží na tom, kolik je odebráno singulárních hodnot.

Tento přístup je vyzkoušený třeba v (3), kde je použitý pro rozpoznávání synonymických vztahů. Jako výstup této práce jsou očekávány synonyma, hyperonyma (hyponyma) a holonyma (meronyma). Tyto druhy vztahů se kromě specifických syntaktických konstrukcí rozpoznat už jen z kontextu.

3.2 Implementace

Tato sekce se zabývá komplikacemi, které mohou nastat v samotné implementaci programu. A to v části výberu kontextu a nastavení LSA.

3.2.1 Výběr kontextu

Vstupem této části je souvislý text z libovolného zdroje, seskupený do vět.

Na správném výběru kontextu záleží úspěch celého dalšího postupu. je potřeba vybrat dostatečně široký na to, aby určoval zkoumané slovo, ale ne tak rozsáhlý, aby zbytečně nezvětšoval objem dat a nezabíral výpočetní kapacitu na úkor výstižnějších slov. V této práci jsou otestovány tři druhy kontextu. Celá věta, trojice po sobě jdoucích slov ve větě a sousedé v závislostním stromě analyticky anotované věty. Jakožto referenční slovní druhy, které se počítají do kontextu, byly vybrány pouze ty, které nesou význam. Tedy substantiva, adjektiva, verba a adverbia.

Zkoumaným druhem byla zvolena jen podstatná jména. Důležité je také, aby do svého kontextu bylo zahrnuto i samotné zkoumané slovo.

Výstupem této části je incidenční matice, jejíž řádky reprezentují zkoumaná slova a její sloupce referenční slova, jichž jsou zkoumaná slova součástí.

3.2.2 Získávání vztahů

Vstupem této části je řídká matice incidence z předchozí části.

První fází LSA je SVD, která není nijak parametrizovaná, tedy zde nelze nic ovlivnit. V další fázi je potřeba odebrat správné množství singulárních hodnot. Parametr pojmenovaný *singulars* zde určuje, jaký bude poměr počtu zbylých singulárních hodnot vůči původnímu počtu. Čím větší je tento parametr, tím více původních singulárních hodnot zůstává a tím menší je generalizace významů zkoumaných slov. Při hodnotě *singulars=1* zůstane matice stejná, ke generalizaci nedojde, naproti tomu hodnota *singulars=0* způsobí absolutní generalizaci, tedy určí, že všechna slova mají stejný význam.

Násobením původních dvou získaných matic a nové diagonální matice singulárních hodnot získáme generalizovaný sémantický prostor. Podobnost významů dvou slov pak získáme nějakou korelační funkcí. V této implementaci je použito Spearmanovo ρ, protože na malých zkužebních datech více rozlišovalo nesouvisející slova.

Z takto získaných ohodnocení mezi jednotlivými slovy je potřeba vybrat, která jsou si nejbližší. Tedy stanovit určitou hladinu, nad kterou musí být hodnota jejich korelace. Parametr určující hladinu nazývejme *limit*.

Výsledkem jsou skupinky slov, vzájemně významově blízké, rozepsané do dvojic. Lze použít i jiné metody, například z tohoto prostoru přímo získávat celé skupinky specializovanými metodami z oblasti statistiky, nebo strojového učení.

3.3 Analýza dat

Původní ambicí bylo porovnávat výsledky této metody se vztahy převedenými z CzechWordNetu. To ale vycházelo velmi špatně. Maximálně se shodovaly jednotky procent získaných vztahů. Přtom při ručním hodnocení výsledků vycházely čísla mnohem vyšší.

Důvodem je rozdíl mezi obsahem zpracovávaných dat a obsahem CzechWordNetu. Zatímco CzechWordNet zachycuje všeobecně platné vztahy (*zvíře hyp pes*), zkoušený nástroj zachycoval z novinových článků aktuální informace. Například významové seskupení slov *útok, arab, Fatah, Arafat, ...* vzniklo ze článků o situaci na blízkém východě.

Dalším měřítkem je ruční hodnocení. Jako správné vztahy byly ohodnoceny ty vztahy, které byly v jednom z výše vyčtených vztahů (*syn, hyp, mero, holo*), nebo pokud obě slova byla přímo významově podřazená jinému (*Poděbrady, Pardubice - oboje jsou česká města*). Pokud byla slova v jiném než příjmaném vztahu (například *atribut*), byla jejich vazba ohodnocená jako jiná. Ostatní vztahy byly vyhodnoceny jako špatné.

3.4 Zhodnocení výsledků

Rozdělení získaných vztahů do tří kategorií (*správné,jiné,špatné*) prováděné ručně závisí do jisté míry na subjektivním rozhodnutí anotátora, proto jsou níže uvedená procenta jen orientační hodnoty. Při testování na malých datech bylo získáno také velmi málo vztahů, což je opět řadí mezi orientační hodnoty.

K vyzkoušení metody na opravdu velkých datech nebyla k dispozici dostatečná výpočetní kapacita. Dalším pokračováním tohoto projektu bude tedy testování s více daty.

Tabulka 3.1 ukazuje procentuální rozdělení získaných vztahů. Metody získávání kontextu (Kont) jsou: p - analyticky pársovaná věta, t - tři po sobě jdoucí slova a s - celá věta. Hodnota podílu počtu singulárních hodnot (Sing) je testována na začátku, uprostřed a na konci jejího oboru. Parametr limit je nastavován tak, aby metoda vracela potřebný počet relevantních vztahů.

Tabulka 3.1: Výsledky metody

Kont	Limit	Sing	Vět	Relací	správné	jiné	špatné
s	0.999999	0.2	34	84	26%	20%	54%
s	0.9999	0.5	34	148	28%	20%	52%
s	0.99	0.8	34	152	29%	16%	55%
t	0.95	0.2	34	176	26%	13%	61%
t	0.87	0.5	34	46	44%	15%	41%
t	0.80	0.8	34	5	60%	20%	20%
p	0.99	0.2	34	39	40%	14%	46%
p	0.95	0.5	34	21	35%	23%	42%
p	0.85	0.8	34	7	57%	43%	0%
s	0.999999	0.2	328	356	23%	14%	62%
s	0.9999	0.6	328	461	37%	18%	45%
s	0.99	0.8	328	396	39%	19%	42%
p	0.9	0.9	328	153	42%	21%	39%

Jako nejlepší metoda výběru kontextu se z těchto orientačních dat jeví analytické pársování s výběrem sousedících slov, při volbě hodnoty *Singular* bližší 1, společně s vybráním celé věty a vyšším limitem.

Kapitola 4

Uživatelská dokumentace

Tato kapitola obsahuje informace o instalaci, spuštění a ovládání všech výše zmíněných programů. Tedy Semantixu, Semantic comparatoru a Semantic-XML Browseru.

4.1 Semantic-xml

Semantic-xml je definice XML, podle které se ukládají sémantické vztahy mezi lematy do XML. Jeden takový XML soubor obsahuje orientovaný graf, jehož uzly jsou lemata a hrany jsou vztahy. Každé lema má svoje identifikační číslo, slovní druh a zápis. Příklad uložení dvou lemat:

```
<lemma id="l_2547" pos="n">vchod</lemma>
<lemma id="l_2548" pos="n">vjezd</lemma>
```

Text uzavřený uvnitř je zápis lematu. Značka má jako povinné atributy bližší vlastnosti lematu. Id =identifikátor lematu ve tvaru $l\_<$*číslo lemmatu*$>$ a pos =slovní druh lematu, který může nabývat hodnot $(n|adj|adv|v)$ pro podstatná jména, přídavná jména, příslovce a slovesa resp. Nemá význam ukládat jiné slovní druhy, protože vztahy s nimi nejsou pro sémantický slovník hodnotné. Právě naopak.

Každý sémantický vztah má svoje unikátní identifikační číslo, typ a spolehlivost. Dále jsou v párové značce *relation* uzavřené odkazy na lemata, mezi kterými relace vede a odkaz na zdroje. Zdrojů může být i více, podle toho kde se relace vyskytuje. Mezi dvěma lematy může být i několik různých vztahů. Příklad uložení vztahu mezi výše uvedenými lematy:

```
<relation id="r_1023"~cat="SYN" reliability="0.75">
    <lemma_ref id_ref="l_2547" role="0"/>
    <lemma_ref id_ref="l_2548" role="0" sense="1"/>
    <source name="CzechWordnet" line="3488"/>
</relation>
```

Značka *relation* má povinné atributy *id* = identifikátor vztahu *cat* = druh vztahu a *reliability* = spolehlivost vztahu v rozmezí <0;1>. Značky *lemma_ref* určují, kterých lemat se vztah týká (atributy *id_ref* odkazují do sekce lemat). Atribut *role* určuje úroveň lematu ve vztahu. Vztah vede z lematu s vyšší hodnotou *role* do nižší. Pokud jsou *role* stejné, je vztah obousměrný. Atribut *sense* zastupuje číslo významu daného lematu ve wordnetu. Je nepovinný, protože jej nelze získat ze všech druhů zdrojů. Dále musí být uvedena alespoň jedna značka *source*. Tam jsou uložena místa, odkud je tento vztah získaný *(name, line/link).*

4.2 Semantix

Program semantix je určený k běhu pod systémem unix. Následující oddíly předpokládají běh pod tímto systémem. Pokud chcete spouštět semantix na MS Windows, prosím kontaktujte autora.

Semantix je nástroj na získávání sémantických vztahů popsaných níže z PDT (13), WordNetu (5), semantic-xml a prostého textu. Vztahy jsou ukládané podle definice semantic-xml popsané výše.

4.2.1 Popis instalace a konfigurace

K instalaci semantixu stačí rozbalit archiv *semantixcz.tar.gz* a v adresáři spustit dávkový soubor *install.sh* v adresáři, kam byl semantix rozbalen. Instalátor přeloží zdrojové soubory a připraví spustitelný soubor *semantixcz* do adresáře, kam byl distribuční balíček rozbalen. Ke správnému běhu všech funkcí programu musí mít uživatel nainstalované následující programy: TrEd, R (s balíčkem lsa a Rstem) a Collinsův parser spolu s převaděčem formátů PDT.

TrEd lze stáhnout z (14), tam jsou také instrukce k jeho instalaci. R lze stáhnout z (12). Balíček lsa lze stáhnout a nainstalovat pomocí správce balíčků R, viz dokumentace R. S balíčkem Rstem jsou při instalaci problémy. Pro jeho začlenění do programu je doporučeno spustit v R příkaz
$install.packages(\text{"}Rstem\text{"}, repos = \text{"}http://www.omegahat.org/R\text{"}, type = \text{"}source\text{"})$
Collinsův parser a převaděč mezi formáty PDT je možné stáhnout z

http : *//ufal.mff.cuni.cz/pdt*2.0/*tools/*
Pokud budou parser a převaděč formátů (adresáře *machine-annotation* a *format-conversions*) umístěné jinde, než v podadresáři *Tools* instalace, bude potřeba upravit cestu k nim v dávkových souborech, které zajišťují kontext. Ty jsou v adresáři
./Tools/WordGroups

4.2.2 Ovládání programu

Argumenty při startu programu jsou jména souborů se vstupními daty. Semantix umí zpracovat data typu semantic-xml (přípona *.xml*), WordNet (přípona *.ewn*), tektogramaticky anotovanou vrstvu Pražského závislostního korpusu - PDT (přípona *.t.gz*) a prostý text (přípona *.txt*). Podle přípony zadaných souborů se určí jejich typ ($EWN|PDT|XML|TXT$). Jako název zdroje dat při ukládání vztahů z procházeného souboru se u EWN použije název souboru bez přípony. U XML se nechá původní název. U PDT je název zdroje *PDT*, protože adresa ve zdroji (atribut line viz specifikace) je u PDT přesně dané absolutní ID. Adresa ve zdroji je u vztahů získaných z prostého textu nevyplněná, protože vztahy jsou získávány globálně z celého textu, takže přesný zdroj nelze určit. Jako název zdroje se vyplní TXT.

Pro získávání sémantických vztahů z PDT je potřeba mít nainstalované programy TrEd a Perl, pomocí kterých je PDT procházeno. Ve stejném adresáři jako je spouštěný soubor semantixcz musí také být soubor *functor.btred*, ve kterém je uložené makro v Perlu pro získávání vztahů z PDT. Ve stejném adresáři se také hledá soubor *semantixcz.cfg*, kde je uložená cesta k programu btred, což je procesor maker programu perl pro TrEd. Pokud tento soubor chybí, nebo je prázdný, hledá se tred na cestách uložených v proměnné $path.

Výstupem programu je orientovaný graf, ve kterém jsou uzly tvořeny lematy a hrany relacemi. Výstupní data jsou ve formátu semantic-xml, tedy v XML uložený seznam lemat a seznam relací mezi nimi. Semantix posílá výsledné XML na standardní výstup, je typicky potřeba přesměrovat výstup do souboru. Pro správnou funkci programu musí být vstupní soubory v kódování UTF-8. Výstup je pak ve stejném kódování.

4.2.3 Druhy sémantických vztahů

Druhy sémantických vztahů jsou inspirované vztahy ve WordNetu (5), tektogramatickými funktory v PDT (13) a vztahy v MindNetu (9). Výsledkem je tabulka 4.1.

Tabulka 4.1: Sémantické vztahy a jejich zdroje

Zkratka	Sémantický vztah	z tekt. funktorů	ve WordNetu
ACMP	Doprovází / je doprovázen	ACMP	-
ADDR	Je adresátem / Má adresáta	CONJ, CM	-
ADVS	Je v odporovacím vztahu	ADVS, CONFR, CONTRA, CONTRD	-
ANT	Jsou Antonyma	-	near_antonym
ATTR	Je vlastností / má vlastnost	APP, COMPL, MANN, MAT, MEANS, ORIG, RSTR, CRIT, EFF	-
AUTH	Původce	AUTH, HER	-
BEN	je prospěšný (komu) / má prospěch (z)	BEN	-
CAUS	Způsobuje / je způsobeno	CAUS	-
CONJ	Slučovací vztah	CONJ, CM	-
CPR	Porovnání, míra	CPR, DIFF, EXT	-
LOC	Jit odkud, kudy, kam? Kde?	DIR1, DIR2, DIR3, LOC	-
DISJ	Vylučovací vztah	DISJ	-
DPHR	Slovní spojení, sousloví	DPHR, ID	-
INTT	Má záměr / je záměrem, účel.	INTT, AIM, CSQ, REAS, RESL, REG	-
PRED	Predikace, Rekce	PRED, ACT	-
SUBS	Substituce	SUBS	-

Tabulka 4.1: (pokračování)

Zkratka	Sémantický vztah	z tekt. funktorů	ve WordNetu
TDUR	Trvání	THFL, THL, THO, TPAR	-
TIME	Čas kdy	TFRWH, TWH, TSIN, TTILL, TWHEN	-
MERO	Je meronymem / je holonymem	-	has_mero_part, has_holo_part, has_mero_member, has_holo_member, is_subevent, has_subevent
HYP	Je hyperonymem / má hyperonymum	-	has_hyponym, has_hyperonym
SYN	Je synonymem / má synonymum	APPS	Zápisy jednoho synsetu
UNDEF.	Neurčený vztah	-	-

4.3 Semantic Comparator

Semantic Comparator je nástroj, který umožňuje porovnávání a hodnocení spolehlivosti sémantických vzahů. Jeho vstupem jsou buď dva soubory typu semantic-xml, jejichž obsah je porovnán a vypsán na standardní výstup, anebo jeden soubor typu semantic-xml, z něhož jsou postupně nabízeny vztahy a uživatel určí, jestli je vztah platný, jiný nebo špatný.

V prvním případě je výstup ve tvaru $< relacev1.souboru >:< relacev2.souboru >$. Pokud je vztah obsažen v obou souborech, jsou vypsány obě strany dvojtečky. Pokud je relace pouze v prvním (druhém) souboru, vypíše se pouze levá (pravá) strana dvojtečky. Comparator vypisuje porovnané vztahy na standartní výstup. Pro shrnutí výsledku porovnání je připraven dávkový soubor jménem *run_comparator.sh*, který vyjádří počty shodujících se relací v obou souborech.

V druhém případě comparator rozděluje vztahy to tří sloupečků oddělených ddvojtečkou. Do levého sloupečku jsou zapsané platné vztahy, do druhého jiné a do třetího

neplatné. Po skončení je vypsán editace na standartní výstup. Pokud je hodnocení předčasně ukončeno příkazem "e", je zbytek relací vypsán ve formátu semantic-xml na standardní výstup za rozdělené relace. Obě části jsou oddělené prázdným řádkem. Je důležité si výstup přesměrovat do souboru, jinak bude práce ztracena.

Instalace semantic comparatoru je blíže popsána v sekci instalace semantixu.

4.4 Semantic-XML Browser

Semantic-XML Browser je webový nástroj v asp .NET pro orientační procházení semantic-xml, který umožňuje hledat nejkratší cesty mezi lematy a procházet lemata a vztahy v databázi uložené na serveru. Tento prohlížeč je nasazený v testovacím provozu na adrese *http* : *//semantix.aspx.sk*. Podrobnosti jeho používání jsou popsány anglicky přímo u aplikace v sekci *Documentation*.

Kapitola 5

Programátorská dokumentace

V následující kapitole jsou popsané programátorské podrobnosti implementací všech programů, které patří do této práce. Detailní popis funkcí se nachází přímo v komentářích zdrojového kódu.

5.1 Semantix

K programování semantixu byl použit jazyk C++ a jeho standartní knihovny. Dále byl použit soubor knihoven STL (standard template library), kde jsou obsažené generické algoritmy a datové struktury. Semantix také spouští příkazovou řádku systému a z ní pomocné vnější programy.

5.1.1 Běh programu

Po startu programu se pro každý argument se rozhodne, o jaký typ zdroje se jedná. Pro EWN a XML se spustí specifické dekódovací metody, které načtou daný vstupní soubor do objektu typu Source. Pro PDT a TXT se nejdříve vyberou všechny soubory daného typu a spustí se pro ně dekódovací metoda (důvod viz metody převádění PDT a prostého textu). Třída Source obsahuje spojové seznamy sémantických vztahů a lemat. Po získání informací ze všech vstupních souborů se sloučí vztahy mezi stejnými lematy se stejnými významy. Pokud význam jednoho nebo druhého lematu není určen a lemata se shodují, jsou relace také sloučeny. Při sloučení dvou relací se přepočítá výsledná spolehlivost, podle spolehlivosti původních relací. Viz níže. Nakonec následuje výpis dat na standardní výstup.

5.1.2 Třídy a datové struktury

Objekty pro uložení informací o lematu a relaci se jmenují Lemma a Relation a jsou uloženy ve stejně pojmenovaných souborech. Jejich metody jsou blíže okomentovány ve zdrojovém kódu. Třída, která pracuje s lematy se jmenuje Lemma_rel. Vytváří spojový seznam z objektů typu Lemma a z dalších informací o lematech. Obsahuje všechny informace potřebné pro uložení lemat u relace, tedy význam (int sense) lematu a jeho roli (int role) v relaci. S relacemi pracuje třída Relation_list, která podobně jako Lemma_rel má funkce spojového seznamu. Pro uložení zdrojů u relace se používá třída Source_rel, která obsahuje spojový seznam informací o jednotlivých zdrojích relací, tedy název zdroje (string name) a adresu ve zdroji (string line).

Nejvyšší třída pro práci s daty se jmenuje Source. Z ní jsou odvozeny třídy zajišťující načítání a konverzi dat ze vstupních souborů. Seznam všech lemat je ve třídách dědících od Source uložen v proměnné lemmas typu ukazatel na Lemma_rel (Lemma_rel *lemmas) a seznam všech vztahů v proměnné relations typu Relation_list (Relation_list *relations).

Všechny následující třídy jsou odvozené od třídy Source a převádějí data v souborech zadaných jejich parametry. Třída, která zpracovává WordNet se jmenuje Transform_EWN. Konstruktor Transform_EWN(char *filename) převádí data ze souboru, jehož jméno je zadané ve filename do svých zděděných vnitřních struktur lemmas a relations. Stejným způsobem funguje převádění XML, PDT a prostého textu. Transform_XML(char *filename) je konstruktor stejnojmenné třídy, která převádí do vnitřních struktur semantic-xml. Konstruktor Transform_PDT(string filenames) převádí data z PDT. Vstupem je zde seznam souborů PDT oddělený mezerami. A nakonec kostruktor Transform_TXT(string filenames), který zpracuje soubory prostého textu, určené jmény oddělenými mezerou v parametru filenames.

Třída Source také obsahuje metody pro sjednocení obsahu. Metoda void UnifyLemmas(void) unifikuje lemata - smaže duplicitní položky a pak přiřadí unikátní ID. To dělá pomocí binárního vyhledávacího stromu. Metoda void UnifyRelations(void) sloučí relace se stejnými kategoriemi, lematy a jejich významy a přepočítá jejich pravděpodobnost. Sloučení relací se provede po setřídění algoritmem quicksort podle unikátního indexu relace. Ten obsahuje druh vztahu, ID obou lemat a jejich role a významy. Index tak jednoznačně určuje relaci, takže po setřídění budou stejné relace za sebou a konečné slučování proběhne v lineárním čase. Nakonec metoda UnifyRelations přiřadí každé relaci unikátní ID.

Počítání pravděpodobnosti sloučených vztahů zajišťuje metoda void ComputeReliability(void) třídy Relation.

Třídy, které ukládají informace o relaci, lematu, nebo zdrojích obsahují metodu Out (string *Out()), která vrací jejich obsah v semantic-XML jako řetězec. Tyto metody

pak používá i třída Source ve své metodě StdOut (void StdOut(void)) která vypisuje obsah vnitřních datových struktur objektu ve formátu XML na standartní výstup.

5.1.3 Použité algoritmy

K získávání vztahů z WordNetu se používá třída EWNDocument, která v sobě uloží obsah a strukturu vstupního souboru. Třída Transform_EWN pak jen převede vnitřní závislosti z EWNDocument do vlastních vnitřních struktur relations a lemmas.

Z WordNetu se získávají jen vztahy typu syn, hyp, mero a attr. Všechny zápisy jednoho synsetu (WordMeaning) se převádějí jako synonyma. Převádění ostatních vztahů z části synsetu označené *INTERNAL_LINKS* je nastavitelné v konfiguračním souboru *EWNRels.cfg*. Každá řádka tohoto souboru představuje převádění jednoho druhu vztahu podle WordNetu. Každá řádka obsahuje název vztahu ve WordNetu, roli prvního lematu, roli druhého lematu a název vztahu podle semantic-xml oddělené dvojtečkami. Řádek obsahující na místě názvu vztahu z WordNetu symbol * určuje druh relace, pokud se neshodoval název testované relace z WordNetu s žádným názvem uvedeným nad ním.

Tektogramaticky anotovaná vrstva PDT je dekódována pomocí maker jazyka perl ovládajícím program btred uložených v souboru *functor.btred*. Použitá cesta k programu btred je uložená v konfiguračním souboru *semantixcz.cfg* pod názvem *tred_path*. Semantixcz spustí v příkazové řádce btred pro všechny vstupní soubory a výstup přesměruje do dočasného souboru. V tomto souboru jsou uložené na každém řádku vlastnosti jednoho vztahu. Formát řádků souboru je:

$--REL--$ *druh_relace slovni_druh*1 *lema*1 *slovni_druh*2 *lema*2 *adresa*

Kde $--REL--$ značí že jde o řádek obsahující data. *druh_relace* je jeden ze sémantických vztahů určených definicí semantic-xml. *lema*1 je zápis lematu od kterého daný vztah vede. *slovni_druh*1 je jeho slovní druh. další dvě položky, *slovni_druh*2 a *lema*2, značí stejné informace o cílovém lematu relace. *adresa* je ID uzlu v PDT, ze kterého lze získat absolutní adresu v korpusu, odkud byl vztah získán.

Makra získávající vztahy z PDT nejdříve vyfiltruje všechna lemmata, která nejdou použít (např. #PersPron, nebo uzel, který nemá otce) a pak rozhodne podle tektogramatických funktorů, do jakého vztahu patří probíraný uzel a jeho otec.

Makra v souboru *functor.btred* lze libovolně upravovat, ale jejich výstup musí mít strukturu zmíněnou výše. Řádky, které nebudou začínat prefixem *–REL–* nebudou brány v potaz a ukončí načítání dalších relací.

Převádění prostého textu je velmi konfigurovatelné. Nejprve je vstupní text rozdělen do souborů po tolika řádcích, kolik je v konfiguračním souboru *semantixcz.cfg* pod názvem *num_sentences*, aby mohlo být další zpracování v případě potřeby paralelní. Soubory se uloží do adresáře
./Tools/WordGroups/out pod názvem uvedeným ve zmiňovaném hlavním konfiguračním souboru u položky *total_txt*. Text v těchto souborech se musí překódovat ze znakové sady UTF-8 na ISO-8859-2. To je kódování příjmané parserem. Viz níže. Dále se z příkazové řádky spustí příkaz uložený v tomtéž konfiguračním souboru pod názvem *word_groups*. Ve výchozím nastavení se takto spustí dávkový soubor
./Tools/WordGroups/a_parsed.sh (na výběr jsou ještě *./Tools/WordGroups/m_trigrams.sh* pro vytvoření trigramů a *./Tools/WordGroups/a_sent_cont.sh* pro celou větu jako kontext). Z něj se spustí collinsův parser na rozkouskovaný vstupní text. Jako výstupní adresář je určený
./Tools/WordGroups/out v něm parser vytvoří pět adresářů, z nichž poslední (*5-pml*) obsahuje soubory se strojově anotovanými vstupními větami na analytické rovině.

Tyto soubory jsou dále zprácovány btredem, který vypíše kontextové skupiny slov do souboru se stejným názvem, jako obsah položkoy *groups_txt* v konfiguračním souboru *semantixcz.cfg*.

Z tohoto souboru pak vytvoří semantix incidenční matici, podle toho, která slova jsou spolu na řádce. Matici vypíše ve formátu csv do souboru určeném položkou *matrix_csv* v konfiguračním souboru *semantixcz.cfg*. Matice je následně načtena při provádění dávkového souboru programu R, jehož jméno je uložené pod názvem *make_relations.sh* ve stejném konfiguračním souboru.

Výstupem dávky programu R je seznam sémantických vztahů v dočasném souboru jehož jméno je pod položkou *out* v konfiguračním souboru
./Tools/Matrix2Relations/lsa.cfg a pod názvem *relations_txt* v hlavním konfiguračním souboru. V konfiguračním souboru převáděcího dávkového soboru jsou také uložené parametry $limit$ a $singulars$, parametrizující běh LSA.

Řádky dočasného výstupního souboru jsou ve tvaru

$--REL--$ $lema1$ $lema2$ *spolehlivost*

Které jsou stejně jako při převádění PDT uvedeny předponou $--REL--$. $lema1$ a $lema2$ jsou lemata mezi kterými vztah vede a spolehlivost je orientační hodnota rozdílu významů daných lemat. Z dočasného souboru jsou vztahy převedeny obdobně jako v případě PDT.

Obě externí fáze programu je možné upravovat do té míry, aby zůstal stejný přechod mezi vnějšími programy a semantixem. Například vstup z první externí fáze, kdy se vytvářejí skupinky slov, které mají společný kontext, může být generován třemi různými výše zmíněnými dávkovými soubory.

5.2 Semantic Comparator

Semantic Comparator využívá stejné knihovny, jako semantix. Hlavní třídou která pracuje s daty je *Transform_XML*, zmíněná výše, pomocí jejíž metody *void LoadXML(char* filename)* jsou data načtena. Po unifikaci seznamů lemat a vztahů zděděnými metotodami z třídy *Source* je podle počtu vstupních argumentů rozhodnuto, jestli se použije vypisovcí metoda *void ComparedOut(Source *compared)*, nebo metoda *void ConfirmedSelection(void)*, která vybraným neplatným vztahům přiřadí nulovou spolehlivost a pak všechny je vypíše.

První metoda seřadí relace v *this* a v *compared* a postupně odebírá, z toho seznamu, kde je na aktuální pozici vztah výše v abecedě. Pokud jsou vztahy na právě probíraných pozicích v obou seznamech stejné, vypíše oba oddělené dvojtečkou, jak je popsáno v uživatelské dokumentaci.

Druhá metoda postupně nabízí uživateli relace ze vstupního souboru, a ten určuje, jestli je relace platná, nebo ne. Výstup je rovnou vypisován ve formátu popsaném v uživatelské části na standardní výstup. V případě přerušení příkazem "e" je pak zbytek vypsán v semantic-xml na standardní výstup metodou Source::StdOut(void).

5.3 Semantic-XML Browser

Webová aplikace Semantic-XML Browser je napsaná v jazyce C# pro platformu asp .NET. Lemata a relace jsou po nahrání souboru semantic-xml na web uložena do databáze v MSSQL.

5.3.1 Struktura databáze

Databáze obsahuje tři tabluky, jejichž obsah kopíruje definici semantic-xml a jednu navíc, kam se ukládají soubory nahrané na server.

Tabulka *Lemmas*, kam se ukládají lemata, obsahuje primární klíč *lemID (nchar(10))*, kam se ukládá id lematu, a sloupce *lemText (nvarchar(50))* a *lemPOS (nchar(10))*, kam se ukládá zápis a slovní druh lematu.

Do tabulky *Relations* se ukládají vztahy. Do sloupců *relLemma1 (nchar(10))*, *relLemma1Role (int)* a *relLemma1Sense (int)* se ukládají informace o prvním lematu. Sloupec *relLemma1* je cizí klíč odkazující do tabulky *Lemmas*. Druhé lema ve vztahu je uložené v obdobně pojmenovaný sloupcích, akorát s číslem "2". Další informace o relaci jsou ve sloupcích *relID (nchar(10))*, *relType (nchar(10))* a *relReliability (decimal)*. První z nich je id relace a zároveň je to primární klíč tabulky, druhý ukládá kategorii sémantického vztahu a poslední spolehlivost.

V tabulce *Sources* jsou uložené odkazy na zdroje relací. Sloupec *srcID (int)* je primárním klíčem tabulky. Do sloupců *srcName (nvarchar(50))* a *srcLine (nvarchar(50))* se ukládají informace o zdroji, stejně jako v semantic-xml. Sloupec *srcRelID (nchar(10))* je cizí klíč, odkazující do tabulky *Relations*. Tento sloupec určuje, ke kterému vztahu zdroj náleží.

V tabulce *Files* jsou uloženy informace o souborech nahraných na server. Sloupec *filID (int)* je primárním klíčem tabulky. Ve sloupcích *filName (nvarchar(50))* a *filSize (int)* se ukládá název a velikost souboru na serveru. Sloupec *filLoaded (bit)* určuje, jestli jsou data ze soboru právě nahraná v databázi.

5.3.2 Algoritmy a datové struktury

Procházení lemat se odehrává na stránce *SearchLemmas.aspx*. Filtrování a hledání podle zadaných kritérií se odehrává na úrovni databáze. Stejně je tomu i na stránce *SearchRelations.aspx*, kde je možné procházet lemata v databázi.

K procházení semantic-xml souboru je pořeba jej nejdříve nahrát na server pomocí stránky *Database.aspx* a poté kliknutím na *Load* u jeho názvu se nahraje jeho obsah do databáze. Při prvním hledání nejkratší cesty na stránce *SearchPath.aspx* se obsah databáze v tabulkách *Lemmas* a *Relations* nahraje do paměti a všechna další hledání se pak provádějí v datových strukturách přímo v paměti.

Datová struktura, ve které jsou relace a lemata uloženy se jmenuje *Graph* a je uložená v souboru *Graph.cs*. Lemata a vztahy mezi nimi jsou tam uloženy v kontejneru *List* obsahujícím prvky *GraphLemma*. Každý prvek *GraphLemma* obsahuje ID uloženého lematu a seznam s ním sousedících vztahů uložených ve struktuře *GraphRelations*, kde jsou otočené tak, aby první lema bylo vždy to, které se shoduje s lematem uloženým v *GraphLemma*.

Nejkratší cesta se hledá v komponentě *ShortestPath.ascx*. Na výběr jsou dva algoritmy. První najde jednu nejkratší cestu a druhý najde všechny cesty stejně dlouhé jako nejkratší cesta. Počítání délky cesty viz níže.

První způsob hledání nejkratší cesty v grafu používá Dijkstrův algoritmus kde je k počítání délky cesty P použitý vzorec $-\sum_{r \in P} \ln r.spolehlivost$. Tím se udrží význam spolehlivosti ve smyslu pravděpodobnosti že daný sémantický vztah existuje. Pro dva jevy P_1 a P_2 je pravděpodobnost jejich průniku vyjádřená jako $P_1 \cap P_2 = p_1 * p_2$, což se dá transformovat na vzdálenost tak, aby se zachovalo uspořádání vzdáleností (nejmenší pravděpodobnost = největší vzdálenost) pomocí logaritmování: $-(\ln p_1 + \ln p_2)$ podle vzorce $p_1 * p_2 = e^{\ln p_1 + \ln p_2}$.

Druhá metoda hledání nejkratší cesty také používá Dijkstrův algoritmus, ale s tím, že si každý uzel ukládá všechny předky, přes které má stejně dlouhou cestu ke startovnímu uzlu. Cílová podmínka je také upravená. Algoritmus neskončí, pokud najde cílové lema, ale nejdříve projde zbytek otevřených uzlů ve stejné hloubce a pokud přes ně vede cesta do cílového uzlu, jsou i tyto cesty vráceny.

Cesty jsou brané jako stejně dlouhé, pokud rozdíl jejich délek nepřesahuje hranici určenou proměnnou *SAME_DISTANCE_TOLERANCE* v komponentě *ShortestPath*. Její výchozí hodnota je 0.5.

Kapitola 6

Závěr

Představená metoda není průlomovým počinem, který by měl nahradit jiné sémantické nástroje. Spíše by mohla být přidána jako střípek do mozaiky vzájemně se doplňujících nástrojů.

Tato metoda je zatím ještě v začátcích a tento text měl přinést první analýzu její vhodnosti pro český jazyk. Při větším množství dat by jeho úspěšnost měla dosáhnout hodnot, při kterých by už byl samostatně použitelný. Jáké množství vstupních dat by to ale mělo být tato práce nezjistila.

Velkým kladem této metody je, že jejím výstupem jsou přímo skupinky významově příbuzných slov, tedy základ strktury WordNetu. Při správném okruhu vstupních dat by se výstup mohl obsahu WordNetu blížit. Aplikace by tak mohla fungovat přinejmenším jako pomocník při anotaci.

Literatura

[1] Jurafsky D., Martin J. H.: *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice-Hall, 2000.

[2] Klimeš V. :*Analytical and Tectogrammatical Analysis of a Natural Language* PhD thesis, FMP Charles Univesity, Prague, 2006.

[3] Kumaran A., Makin R., Pattisapu V., Sharif S. E., Kacmarcik G., Vanderwende L.:*Automatic Extraction of Synonymy Information: An Extended Abstract*, Ontologies in Text Technology, Osnabrück, 2006.

[10] Landauer T. K., Foltz P. W., Laham D.:*An Introduction to Latent Semantic Analysis*, Discourse Processes, 25, 259-284, 1998.

[5] Pala K., Sevecek P.: *Czech wordnet: Final report Brno*, EuroWordNet, LE4-8283, 1999.

[6] Richardson S. D., Dolan W. B., Vanderwende L.: *MindNet: Acquiring and Structuring Semantic Information from Text*, ACL, CONF 17, Vol. 2, pp. 1098-1102, 1998.

[7] Riloff E., Shepherd J.: *A Corpus-Based Approach for Building Semantic Lexicons*, Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, 1997.

[8] Roark B., Charniak E.: *Noun-phrase cooccurerence statistics for semi-automatic-semantic lexicon construction.*, Proc. ACL-1998, 1998.

[9] Vanderwende L., Kacmarcik G., Suzuki H., Menezes A.: *MindNet: An Automatically-Created Lexical Resource*, Proceedings of HLT/EMNLP 2005 Interactive Demonstrations, Vancouver, British Columbia, Canada, 2005.

[10] Landauer T. K., Foltz P. W., Laham D.:*An Introduction to Latent Semantic Analysis*, Discourse Processes, 25, 259-284, 1998.

[11] *Cyc*, http://www.cyc.com/cyc/technology/whatiscyc_dir/whatsincyc.

[12] *R-project*, www.r-project.org.

[13] *PDT*
http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/cz/html/index.html.

[14] Pajas P.: *TrEd* http://ufal.mff.cuni.cz/~pajas/tred/.

[15] ÚFAL: *Tektogramatická rovina anotace PDT*,
http://ufal.mff.cuni.cz/pdt2.0/
/doc/pdt-guide/cz/html/ch02.html\#a-layers-tecto.

Příloha A

Obsah přiloženého CD

```
bakalarka.pdf  -text bakalářské práce
semantixcz.tar.gz  -zabalený program včetně zdrojového kódu

obsah archivu semantixcz:
+-./src/ -adresář obsahující zdrojové soubory a makefile programů
|         semantixcz a sem_comparator
+-./Tools/
|  +-./Tools/Matrix2Relations/
|  |  +./Tools/Matrix2Relations/lsa.cfg -konfigurační soubor pro LSA
|  |  +./Tools/Matrix2Relations/lsa.r -dávkový soubor pro LSA
|  |  +./Tools/Matrix2Relations/relations.sh -dávkový soubor spouštění LSA
|  +-./Tools/WordGroups/
|     +-./Tools/WordGroups/out/ -pomocný adresář pro parsing
|     +-./Tools/WordGroups/txt/ -pomocný adresář pro parsing
|     +-./Tools/WordGroups/a_parsed.btred -dávka pro parse kontext
|     +-./Tools/WordGroups/a_parsed.sh -dávka pro parse kontext
|     +-./Tools/WordGroups/a_sent_cont.btred -dávka pro věta kontext
|     +-./Tools/WordGroups/cont_sentence.sh -dávka pro věta kontext
|     +-./Tools/WordGroups/guess_enc.pl -určuje kódování
|     +-./Tools/WordGroups/latin2_to_utf8.pl -konverze kódování
|     +-./Tools/WordGroups/m_trig.btred -dávka pro trigram kontext
|     +-./Tools/WordGroups/m_trigrams.sh -dávka pro trigram kontext
|     +-./Tools/WordGroups/utf8_to_latin2.pl -konverze kódování
+-./EWNRels.cfg -konfigurace získávání vztahů z WordNetu
+-./INSTALL  -instalační instrukce
```

```
+-./clean.sh -vyčistí pomocné soubory
+-./functor.btred - makro pro převádění vztahů z PDT
+-./install.sh - instalační skript
+-./run_comparator.sh - skript na vyhodnocování výsledků
+-./run_tests - skript na spouštění více testů
+-./semantic-xml.dtd - definice semantic-xml
+-./semantix.cfg - hlavní konfigurační soubor
```