VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EXTRAKCE SÉMANTICKÝCH VZTAHŮ Z TEXTU

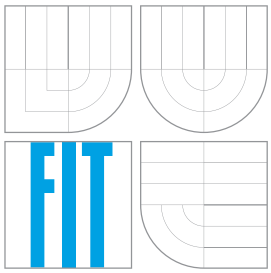
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK SCHMIDT

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EXTRAKCE SÉMANTICKÝCH VZTAHŮ Z TEXTU

EXTRACTION OF SEMANTIC RELATIONS FROM TEXT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK SCHMIDT

VEDOUCÍ PRÁCE
SUPERVISOR

PAVEL SMRŽ

BRNO 2008

Abstrakt

Práce se zabývá extrakcí sémantických vztahů z anglických textů. Zaměřuje se především na použití syntaktické analýzy pro extrakci příznaků, které využívá jak pro různé statistické metody, tak i pro metodu založenou na syntaktických vzorech. Je vyhodnocena metoda extrakce vztahu hypernymie srovnáním s anglickým thesaurem WordNet. Na základě zkoumaných metod je pak navržen systém pro extrakci sémantických vztahů z textu spolu s uživatelským rozhraním, které je rovněž implementováno.

Klíčová slova

extrakce sémantických vztahů, extrakce termů, učení ontologií, ontologie, zpracování přirozeného jazyka

Abstract

Extraction of semantic relations from english text is the topic of this thesis. It focuses on exploitation of a dependency parser. A method based on syntactic patterns is proposed and evaluated in addition to evaluation of several statistical methods over syntactic features. It applies the methods for extraction of a hypernymy relation and evaluates it on WordNet. A system for extraction of semantic relations from text is designed and implemented based on these methods.

Keywords

extraction of semantic relations, term extraction, text mining, ontology learning, ontology, natural language processing

Citace

Marek Schmidt: Extrakce sémantických vztahů z textu, diplomová práce, Brno, FIT VUT v Brně, 2008

Extrakce sémantických vztahů z textu

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Pavla Smrže

.....
Marek Schmidt
12. května 2008

Obsah

Kapitola 1

Úvod

Název této práce zní extrakce sémantických vztahů z textu. V dalších kapitolách podrobně rozepíšu, co si pod takovým názvem vlastně představuji, jaké metody k tomu hodlám použít a nakonec cestu k snad úspěšné výsledné implementaci systému. Tato diplomová práce navazuje na stejnojmenný semestrální projekt. Součastí onoho projektu byl vytvořen systém pro extrakci vztahů založeného na syntaktických vzorech, který bude popsán v sekci o *lexiko-syntaktických vzorech*. S touto prací souvisí také článek [15] publikovaný na EEICT zabývající se použitím XML technologií pro dotazování se nad syntaktickými vzory, kteréhož některé principy jsou součástí sekce o *návrhu reprezetnace závislostního stromu*.

Následující kapitola přibližuje problematiku extrakce sémantických vztahů z textu a vymezuje oblast zájmu pro tuto práci. Třetí kapitola srovnává některé metody pro extrakci zajímavých termů z textu. Čtvrtá kapitola se zbývá různými aspekty extrakce pojmů a vztahů. Pátá kapitola popisuje návrh a implementaci systému pro extrakci sémantických vztahů z textu založeného na metodách popsaných v předcházejících kapitolách.

Kapitola 2

Extrakce sémantických vztahů

Extrakce sémantických vztahů je jen jedna z podúloh obecnějšího problému, který se nazývá *učení ontologií*. Pro vysvětlení toho, co si vlastně pod pojmem extrakce sémantických vztahů máme představit, bude tedy nezbytné definovat pojem ontologie. Náročnější to bude proto, že to, jak si představit pojmy je právě to, čím se ontologie jako věda zabývá.

Slovo ontologie má podle [1] dva pro nás podstatné významy:

- věda o jsoucnu a bytí;
- rigorózní a úplná organizace nějaké znalostní domény, která je obvykle hiearchická a obsahuje všechny relevantní entity a jejich vztahy;

Při učení ontologií se zřejmě budeme zabývat druhým významem onoho slova. Ontologie v tomto významu je tedy jakýsi model reprezentace znalostí.

Základem ontologie je *pojem* (angl. Concept). Vztah mezi objektem (denotátem), slovem a pojmem je vyjádřen na diagramu 2.1 Ogdena a Richardse [11].

- pojem; Dojem, který se v mysli vybaví při vnímání objektu.
- slovo; Symbol, který je používán ke komunikaci.
- denotát; Objekt samotný, který máme na mysli při uvažování v pojmech, či na který se odkazujeme slovy.

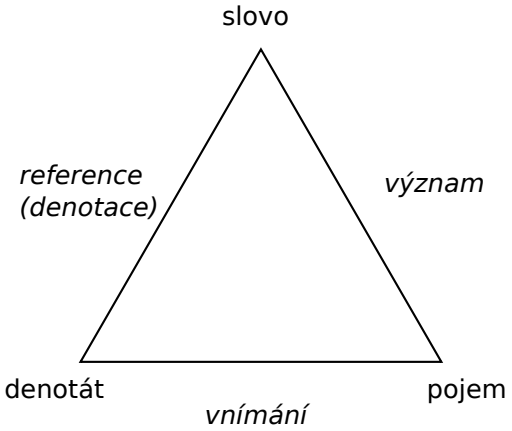
Mezi slova označující stejný pojem platí sémantický vztah *synonymie*.

Na objekty se můžeme dívat s různou mírou abstrakce. Tyto různé abstrakce jsou tedy i různě pojmenovány a tyto názvy pak označují různé pojmy. Na základě abstrakce uvažujme relaci částečného uspořádání *zahrnuje* (subsumption), také známé jako *taxonomie*. Tato relace se také označuje jako relace *is-a*, podle slovesa být, které se často používá k vyjádření tohoto vztahu v textu:

Guitar is a musical instrument. Apple is a kind of fruit.

Pojem hudební nástroj zahrnuje kytaru a je tedy *hypernymem* pojmu kytara. Opačný vztah se nazývá *hyponymie*.

Kromě toho má smysl uvažovat i o obecných, netaxonomických relacích nad pojmy, které popisují různé vztahy mezi instancemi pojmů. Následující věta vyjadřuje vztah mezi pojmy člověk a hudební nástroj, který vyjadřuje, že lidé mohou hrát na hudební nástroje:



Obrázek 2.1: Významový trojúhelník

John plays the guitar.

Sémantickými vztahy rozumíme jakékoliv relace nad pojmy. Pojmy spolu se sémantickými vztahy tvoří ontologii.

Učení ontologií obsahuje čtyři hlavní podúlohy:

1. extrakce pojmů; Pojmy mohu být určeny
   - intenzemi; Popisují nutné obecné vlastnosti, které objekt splňuje, aby byl představován daným pojmem. Intenze můžeme modelovat například jako množinou syntaktických vlastností, které pojem v textu může mít. Na tomto přístupu je založen například metoda FCA[1] popsána P. Cimianem [3].
   - extenzemi; Množna objektů, které jsou představovány daným pojmem. Extenze můžeme modelovat množinou slov, které se v textu vyskytly.
2. extrakce taxonomických relací; Taxonomické relace tvoří hiearchii pojmů. Patří zde
   - hyponymie; X je hyponymum k Y, když X je druhem Y. Vyjařuje se vzorem *is-a*. $hyponym(\text{cat}, \text{animal})$.
   - hypernymie; inverzní relace k hyponymii.
3. extrakce netaxonomických relací; Relace mezi pojmy, jiné než taxonomické. Příkladem budiž vztah *hasCapital* mezi pojmy *state* a *city*.
4. extrakce axiomů; Extrahuje tvrzení, které musí platit pro instance pojmů. Příkladem budiž $\forall x.state(x) \Rightarrow \exists y.city(y) \wedge hasCapital(x, y)$

V této práci se budeme věnovat extrakci pojmů, taxonomických i netaxonomických relací.

Jako jazyk textu nad kterým budeme provádět analýzu byla zvolena angličtina, především z toho důvodu, že angličtina je pro analýzu jednodušší jazyk než čeština a existuje větší množství nástrojů pro zpracování anglického jazyka.

[1] Formal Concept Analysis

2.1 Zpracování přirozeného jazyka

Pro zpracování sémantiky textu je nutné nejprve porozumět jeho syntaktické stránce. Syntaxe vět má stromovou strukturu a jazyky obsahují rozličné prvky, jako jsou předložky, pády a slovosledy, jak převést tuto ideální podobu věty do relativně efektivní lineární struktury, kterou je pak možno vyslovit či zapsat. Tento převod je bohužel nejen velmi často mnohoznačný, a to na obě strany, ale také i obtížně formálně definovaný či vůbec definovatelný. Jazyky se totiž neustále vyvíjejí a jednotliví uživatelé ne vždy zcela přesně respektují autoritami definovaná pravidla. Přes to všechno samotní lidé nemívají s porozuměním ostatních problémy, a to i v případě, kdy se nejedná o syntakticky správné konstrukce. Toto je v jistém smyslu povzbuzující, neboť to naznačuje, že pro porozumění textu možná ani dokonalá syntaktická analýza není potřeba.

2.1.1 Extrakce textu

První fází zpracování textových zdrojů obecně je *předzpracování*. V první podfázi této fáze jde o extrakci samotného textu ze vstupních dokumentů, které mohu být uloženy v nejrozličnějších formátech. Naštěstí většina dnes použitelných formátů na webu i v kancelářského software jsou implementováný v jazyce XML (eXtensible Markup Language). Pro extrakci samotného textu tedy můžeme použít společné knihovny a nástroje pro práci s jazykem XML.

Zpracovat formáty generované programem obecně není velký problém. Ten nastává v případě, kdy je potřeba zpracovávat formáty, které mohou psát lidé ručně, jako je například jazyk HTML. Ačkoliv je to teoreticky jazyk nad XML či SGML, mnohé dokumenty obsahují různé syntaktické chyby, se kterými se při předzpracování musí počítat, a třebaže se nepodaří zrekonstruovat původně myšlený dokument, je žádoucí, aby byl analyzátor schopen se zotavit z takové syntaktické chyby a vyprodukoval tak použitelný výstup. Dostupné knihovny pro práci s chybným HTML kódem naštěstí existují.

Výstupem této fáze předzpracování je prostý text, tedy řetězec znaků v unicode.

2.1.2 Tokenizace

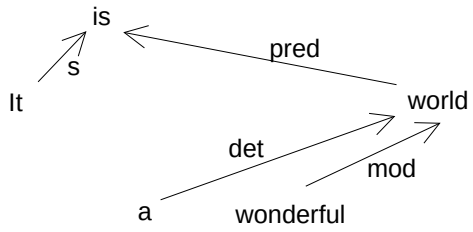
Základní jednotkou, kterou se při zpracování jazyka zabýváme, je slovo. Pro angličtinu není vnitřní struktura slova zvláště zajímavá, což neplatí pro některé jiné jazyky, jako je čeština či turečtina. Z praktických důvodů většina existujících systémů pro transkripci vět jazyka do číslicové formy kóduje slova řetězci znaků konečné abecedy. Při zpracování textu je nutné dekódovat původní slova, tedy převést řetězec znaků na řetězec slov.

Jednotlivá slova se v textu oddělují speciálním znakem (obvykle mezerou), případně interpunkcí. Jednoduchý tokenizátor můžeme vytvořit regulárním výrazem obsahujcící všechny známé slova-oddělující řetězce. Problém komplikují speciální typografické značky, jako jsou oddělovníky a spojovníky, či různé typy mezer. Například abeceda Unicode definuje více než 17 znaků pro mezeru. Problém tokenizace je navíc také obtížný pro některé jazyky, jako je například čínština, které obvykle slova v textu neoddělují.

Výstupem tokenizace je řetězec tokenů, kde tokeny mohou být slova či interpunkce. Samotné tokeny jsou pro nás v této fázi stále jen řetězce znaků.

s	Podmět
obj	Předmět
pred	Predikát klausule
mod	Modifikátor (např. přídavné jméno, které upravuje podstatné jméno)
det	člen

Tabulka 2.1: Některé důležitější závislostní vztahy používané programem MiniPar



Obrázek 2.2: Syntaktická analýza věty “It’s a wonderful world”

2.1.3 Rozdělení vět

V západních jazycích pužívající latinku je obvyklé začínat věty kapitalizací prvního písmene prvního slova věty a končit tečkou. Ve větě se však mohou vyskytnout i jiná slova psaná velkými počátečními písmeny, v češtině to může označovat vlastní jména, či výraz úcty, v němčině například jakékoliv podstatné jméno. Znak těčky může kromě konce věty označovat také zkratky.

Dělení vět je tedy specifické pro konkrétní jazyk, ačkoliv pro západní jazyky lze vyrobit jednoduchý nástroj na dělení vět jednoduchými regulárními výrazy na základě používaných zkratek v daném jazyce s uspokojivou přesností. Na dělení vět se také používá strojového učení natrénovaném na velkých korpusech.

Výstupem této fáze je seznam vět.

2.1.4 Slovní druhy a lematizace

Značkování slovních druhů – v angličtině part-of-speech (POS) – znamená přiřazení slovních druhů jednotlivým slovům.

Při příležitosti značkování je vhodné přiřadit slovům jejich lemma, tedy základní tvar slova. To je užitečné v jazycích s bohatou morfologií (i např. v českém jazyce), protože pro extrakci pojmů je pochopitelně dobré vědět, že slova označují stejné pojmy, přestože se vyskytují v různých tvarech.

2.1.5 Syntaktická analýza

Syntaktická analýza (angl. parsing) odhaluje syntaktickou strukturu věty. Na obrázku 2.2 je znázorněn závislostní graf věty “It’s a wonderful world.” získané syntaktickou analýzou. Hrany označují závislostní vztahy mezi slovy, jejich popis je v tabulce 2.1

2.1.6 Kategorizace pojmenovaných entit

Bohatou množinou slov v jazyce tvoří vlastní jména. Při učení ontologií nás obvykle konkrétní názvy nezajímají, zajímá nás pouze to, jakému pojmu tato jména odpovídají. Rozlišení

toho, zda dané jméno označuje osobu, město, stát, knihu, ... řeší metody kategorizace pojmenovaných entit (Named Entity Resolution)

2.1.7 Vyhodnocení koreferencí

Vyhodnocení koreferencí (Coreference resolution) znamená nalezení slov, která označují stejný objekt. Zde patří vyhodocení *anafor*, což je navázání zájmen na slova, na která odkazují.

2.2 Metody extrakce sémantických vztahů

Extrakce sémantických vztahů, ač sám o sobě široký pojem, má mnoho společného s řadou dalších pojmů. Uvedu malý přehled:

- Text Mining – Obecný pojem pro získávaní znalostí z textu. Kromě extrakce termů, pojmů a vztahů se za text mining dá považovat také klasifikace, shlukování dokumentů, či sumarizace dokumentů. Základem bývá použití metod pro data mining pro text [5]
- Učení ontologií – Především extrakce pojmů a vztahů mezi nimi, jak již bylo posáno v úvodní části této práce.
- Populace ontologií – Doplňování instancí do existující ontologie z textu.
- Extrakce informací – Extrakce entit a vztahů mezi nimi, obvykle z předem známé množiny vztahů. Příkladem může být extrakce informace o změnách vedení ve firmách z novinových článků[2]

Dále uvedu základní principy, ze kterých současné metody pro extrakci sémantických vztahů vycházejí. Konkrétní metody, které jsem jako součást této práce implementoval jsou pak popsány podrobněji v dalších kapitolách.

2.2.1 Statistické metody

Významnou třídou metod jsou metody založené na tzv. batohu slov (bag-of-words). Ten je založen na zjednodušujícím modelu jazyka, ve kterém pořadí slov v dokumentu není rozlišitelné. Tento model se úspěšně používá pro vyhledávání informací či pro klasifikaci dokumentů.

Příkladem použití modelu batohu slov pro extrakci nějakého typu sémantických vztahů uveďme použití analýzy spoluvýskytů podle autorů Sanderson a Croft[14]. Uvádějí metodu pro vztah zobecnění, tedy vztah mezi dvěma slovy, které označují pojmy, kde jeden pojem zahrnuje druhý, tedy že slovo x je obecnější výraz pro y:

> Slovo x zahrnuje slovo y, pokud množina dokumentů obsahující y je podmnožinou dokumentů obsahujících x.

Jak uvádí [3], záleží na výběru toho, co považujeme za dokument. Podle toho, zda za dokument považujeme okolí slov, články, odstavce, či jednotlivé věty, můžeme získat různé významy.

[2] Tato úloha byla tématem soutěže MUC-6

Statistiky o slovech můžeme počítat i na jiné úrovni než výskytech slov v dokumentech. Pro každé slovo můžeme nejprve extrahovat rozličné vlastnosti popisující chování slova v textu. Za takové vlastnosti můžeme považovat lingvistické vlastnosti, jako je slovní druh.

Připomeňme citát anglického lingvisty J. R. Firthe:

> You shall know a word by the company it keeps

Důležitou vlastností slov jsou tedy i výskyty slov okolních, kde okolí může znamenat různé věci, od prosté vzdálenosti slov v textu po syntaktické vztahy získané syntaktickou analýzou textu.

2.2.2 Metody založené na vzorech

Metoda použitelná pro extrakci hypernymických vztahů z rozsáhlých korpusů jsou vzory Hearstové. Jde o běžné vzory, které odhalují vztahy v textu explicitně uvedeny. Využívá při tom mělkou syntaktickou analýzu, při které stačí v textu identifikovat jmenné fráze (NP)

Všimněme si, že například věta

Such elements as gold or platinum.

vyhovuje vzoru

$$such\ NP\ as\ \{NP,\}^*\ \{(and|or)\}\ NP$$

a na základě toho můžeme odvodit, že slova *gold* a *platinum* jsou hyponyma slova *element*.

Mezi vzory navržené Hearstovou jsou tyto:

$$\begin{aligned} NP\ such\ as\ \{NP,\}^*\ \{(and|or)\}\ NP \\ such\ NP\ as\ \{NP,\}^*\ \{(and|or)\}\ NP \\ NP\ \{,NP\}^*\ \{,\}\ or\ other\ NP \\ NP\ \{,NP\}^*\ \{,\}\ and\ other\ NP \\ NP\ including\ \{NP,\}^*\ NP\ \{(and|or)\}\ NP \\ NP\ especially\ \{NP,\}^*\ \{(and|or)\}\ NP \end{aligned}$$

Definuje také metodu nalezení vzorů pro jiné typy vztahů

1. Získáme dvojice slov u kterých je známo, že vztah mezi nimi platí.
2. Nalezneme věty, kde se tyto dvojice vyskytují v syntaktickém vztahu.
3. Vytvoříme vzory zobecněním nalezených výrazů.
4. Použijeme nové vzory a pokračujeme od bodu 1

Vzory mohou mít různou podobu. Jednoduchým vzorem může být bag-of-words model, kde ve větách hledáme výskyt zajímavého slova (*triggering word*). Vzory mohou být tvořeny ručně lidskými experty, či je možné použití strojového učení pro automatickou tvorbu vzorů. Oblast použití strojového učení je zajímavá především pro úlohy extrakce informací a je příliš rozsáhlá a vydala by na samostatnou práci, proto se ji v této práci nezabývejme.

2.2.3 Zdroje dat

Pro všechny zmíněné metody jsou v první řadě potřeba data, tedy texty v daném jazyce. V této práci využívám především English Gigaword Coprpus, který obsahuje články zpravodajství několika agentur. Existují snahy využít jako zdroj dat síť WWWW, a to buď přímo procházením její struktury (crawling) [2] pro vytvoření vlastního korpusu, či využitím vyhledávačů, jako je Yahoo či Google (například [4])

Kromě textových dat, což jsou data nestrukturovaná, je potřeba zmínit také strukturované zdroje jazykových dat, jako jsou slovníky, thezaury a ontologie. WordNet[3] je lexikální databáze angličtiny. Obsahuje informace o slovech a jejich významech, kde mezi významy popisuje základní sémantické vztahy, jako je synonymie, hypernymie, či meronymie. Ačkoliv je možné využít databáze jako je WordNet pro extrakci sémantických vztahů, v této práci využijeme WordNet jen pro evaluaci výsledků.

2.3 Software

Zde uvedu existující softwarové nástroje, které mají souvislost s extrakcí sémantických vztahů a kterými jsem se v souvislosti s touto prací zabýval.

2.3.1 GATE

GATE (General Architecture for Text Engineering) je rámec pro tvorbu aplikací zpracování textu vyvíjený kolem Sheffieldské Univerzity. Obsahuje knihovnu v jazyce Java spolu s grafickým uživatelským rozhraním. Sjednocuje množství užitečných komponent pro zpracování přirozeného jazyka nad jednoduchou architekturou.

GATE definuje tři typy komponent (v GATE označované jako zdroje – Resources)

- Jazykové zdroje (Language Resources)– jazyková data jako jsou slovníky, korpusy a ontologie
- Procesní zdroje (Processing Resources) – kompontenty, které provádějí nějaké operace nad jazykovými zdroji (značkovače, syntaktický analyzátor, . . .)
- Obrázkové zdroje (Visual Resources) – komponenty tvořící uživatelské rozhraní

Základním objektem v GATE je Anotace. Anotace tvoří acyklický orientovaný graf, kde uzlem grafu je ukazatel v textu a hrana obsahuje libovolný objekt reprezentující popis oné části textu pod anotací.

2.3.2 ANNIE

ANNIE (A Nearly-New Information Extraction System) je kolekce komponent distribuovaných spolu s GATE, které obsahují základní nástroje pro zpracování přirozeného jazyka (především angličtiny).

[3] http://wordnet.princeton.edu/

2.3.3 text2onto

Text2Onto je nástroj pro učení ontologií. Kombinuje statistické metody spolu s jednoduchými metodami založenými na lingvistických znalostech, jako jsou lexikální vzory. Pro zpracování textu využívá GATE a obsahuje podporu angličtiny a španělštiny.

Mezi extrahované entity, které text2onto vytváří patří

- Pojmy a instance.
- Taxonomické vztahy mezi pojmy;
- Netaxonomické vztahy, jako jsou vztahy *Subclass-of*, *Subtopic-of*, či vztah podobnosti, a pojmenované vztahy pojmenované slovesem
- Axiomy mezi pojmy, konkrétně prozatím pouze disjunkce pojmů.

Každému extrahovanému vztahu je přiřazena míra jistoty a o každém vztahu lze odvodit vysvětlení jeho existence v textu.

Kapitola 3

Extrakce zajímavých termů

3.1 Metody extrakce zajímavých termů

Extrakce termů je proces, který z daného dokumentu vybere množinu termů. Nad touto množinou vytvoří uspořádání důležitosti.

Prvních N termů v uspořádání důležitosti by mělo vystihovat podstatné termy daného textu, a měli by tedy poskytovat informaci, o čem daný text je. Extrakce termů má své uplatnění například pro kategorizaci dokumentů nebo automatickou tvorbu rejstříků.

V této sekci představím metody založené na dobře známých statistických metodách (viz [10]). Pro čistě statistické metody považujeme za termy samotná slova. Dále uvedu způsob rozšíření těchto metod na dvojice slov.

3.1.1 Extrakce termů relativní četností

Hypotéza: Důležitá slova se v daném textu vyskytují častěji, než v obecném textu.

n	...	Absolutní četnost (počet výslytů) slova w v obecném korpusu T
N	...	Počet slov v obecném korpusu
n'	...	Absolutní četnost w ve zkoumaném textu T'
N'	...	Počet termů ve zkoumaném textu

Relativní četnosti jsou absolutní četnosti normalizovány celkovým počtem slov.

$$f = \frac{n}{N} \tag{3.1}$$

$$f' = \frac{n'}{N'} \tag{3.2}$$

Důležitost slova w na základě relativních četností je pak

$$D_w^{rf} = \frac{f'}{f} = \frac{n'N}{nN'} \tag{3.3}$$

3.1.2 Věrohodnostní test

Test poměru věrohodností (Likelihood ratio test) je statistický test pro srovnání dvou hypotéz.

$\Lambda = \frac{L_0}{L_1}$

Kde L_0 je maximum likelihood hypotézy 0 a L_1 maximum likelihood hypotézy 1.

Formulujme hypotézy:

$$H_0 \quad \ldots \quad \text{Slovo } w \text{ se vyskytuje se stejnou pravděpodobností } p_0 \text{ v } T \text{ i } T'. \tag{3.4}$$
$$H_1 \quad \ldots \quad \text{Slovo } w \text{ se vyskytuje s pstí } p \text{ v } T \text{ a s pstí } p' \text{ v } T' \tag{3.5}$$

Maximum likelihood odhad pravděpodobností pro p_0, p a p' je

$$p_0 = \frac{n + n'}{N + N'} \tag{3.6}$$
$$p = \frac{n}{N} \tag{3.7}$$
$$p' = \frac{n'}{N'} \tag{3.8}$$

Výsledné věrohodnosti L_0 a L_1 spočítáme jako

$$L_0 = \mathcal{L}(p_0|n, N).\mathcal{L}(p_0|n', N') \tag{3.9}$$
$$L_1 = \mathcal{L}(p|n, N).\mathcal{L}(p'|n', N') \tag{3.10}$$

kde

$$\mathcal{L}(p|n, y) = \binom{n}{y} p^y (1-p)^{n-y} \tag{3.11}$$

je věrohodnost binomického modelu.

Poměr hodnot, který vyjadřuje kolikrát je hypotéza 1 pravděpodobnější než hypotéza 0, se vyjádří jako

$$D_w^{lr} = -2\Lambda \tag{3.12}$$
$$\Lambda = \ell(n', N', p) + \ell(n, N, p) - \ell(n', N', p_1) - \ell(n, N, p_2) \tag{3.13}$$
$$\ell(n, N, p) = n \log(p) + (N - n) \log(1 - p) \tag{3.14}$$

3.1.3 Metoda tf-idf

Term frequency – Inverse document frequency (tf-idf) je míra založená na dvou předpokladech:

1. Slovo, který se vyskytuje v daném dokumentu často, je pro tento dokument důležité.
2. Slovo, které se vyskytuje v mnoha dokumentech, není pro konkrétní dokument důležité.

$n_{i,j}$ ... Absolutní četnost slova t_i v dokumentu d_j (3.15)

D ... Množina dokumentů (3.16)

d_j ... Dokument jako množina slov (3.17)

$tf_{i,j}$ je naše známá relativní četnost, tentokráte ovšem pro konkrétní dokument.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3.18}$$

idf_i udává bezvýznamnost slova na základě podílu dokumentů, ve kterých se slovo vyskytlo.

$$idf_i = log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \tag{3.19}$$

Výsledná míra $tfidf_{i,j}$ je pak součin

$$tfidf_{i,j} = tf_{i,j} idf_i \tag{3.20}$$

3.2 Bigramy

Předchozí metody můžeme snadno rozšířit na dvojice slov. n, respektive n' pak budou označovat četnosti dvojic slov $w_1 w_2$ v obecném korpusu, respektive ve zkoumaném textu.

Ve výsledcích vidíme, že tento přístup extrakce zajímavých bigramů produkuje mnoho dvojic slov, které nejsou pojmy a na celkové výsledky mají vliv negativní. Ve skutečnosti nás totiž zajímají jen kolokace. Pokusme se odfiltrovat ty dvojice, které nejsou kolokace.

3.2.1 Věrohodnostní test na kolokace

Použijeme opět věrohodnostní test (podle [10]):

H_0 ... $P(w^2|w^1) = p = P(w^2|\neg w^1)$ (3.21)

H_1 ... $P(w^2|w^1) = p_1 \neq p_2 = P(w^2|\neg w^1)$ (3.22)

n_{12} ... Absolutní četnost dvojice $w_1 w_2$ (3.23)

n_1 ... Absolutní četnost slov w_1 (3.24)

n_2 ... Absolutní četnost slov w_2 (3.25)

N ... Počet slov v korpusu (3.26)

Pak pro odhad pravděpodobností maximální věrohodností p, p_1 a p_2 platí

$$p = \frac{n_2}{N} \qquad p_1 = \frac{n_{12}}{n_1} \qquad p_2 = \frac{n_2 - n_{12}}{N - n_1} \tag{3.27}$$

Pro věrohodnosti hypotéz pak:

$$L_0 = \mathcal{L}(p|n_{12}, n_1).\mathcal{L}(p|n_2 - n_{12}, N - n_1) \quad (3.28)$$

$$L_1 = \mathcal{L}(p_1|n_{12}, n_1).\mathcal{L}(p_2|n_2 - n_{12}, N - n_1) \quad (3.29)$$

kde $\mathcal{L}$ je věrohodnost binomického koeficientu viz 3.11

$$D^c_{w_{12}} = -2\Lambda \quad (3.30)$$

$$\Lambda = \ell(n_{12}, n_1, p) + \ell(n_2 - n_{12}, N - n_1, p) - \\ -\ell(n_{12}, n_1, p_1) - \ell(n_2 - n_{12}, N - n_1, p_2) \quad (3.31)$$

3.2.2 Použití lingvistické informace

Čistě statistické metody můžeme vylepšit dodáním lingvistické informace. Jako pojmy obvykle chápeme podstatná jména. Z výsledků statistických metod odfiltrujme ty slova či bigramy, které nebudou odpovídat zvolenému vzoru slovních druhů.

Jako vzor použijme tento regulární výraz nad značkami značkovače TreeTagger:

$$(JJ|NN|NP|NNS|NPS)^*(NN|NP|NNS|NPS) \quad (3.32)$$

Značka	Význam
JJ	Adjektivum
NN, NNS	Podstatné jméno v singuláru, plurálu
NP, NPS	Vlastní jméno v singuláru, plurálu

3.3 Experiment

Pro experiment jsem použil 35 článků v anglickém jazyce z projektu LT4el[1] s ručne anotovanými klíčovými slovy. Pro každý článek se extrahuje 50 nejlepších výrazů. Na grafech je pak znázorněna neinterpolovaná přesnost.

Srovnání metod unigramů

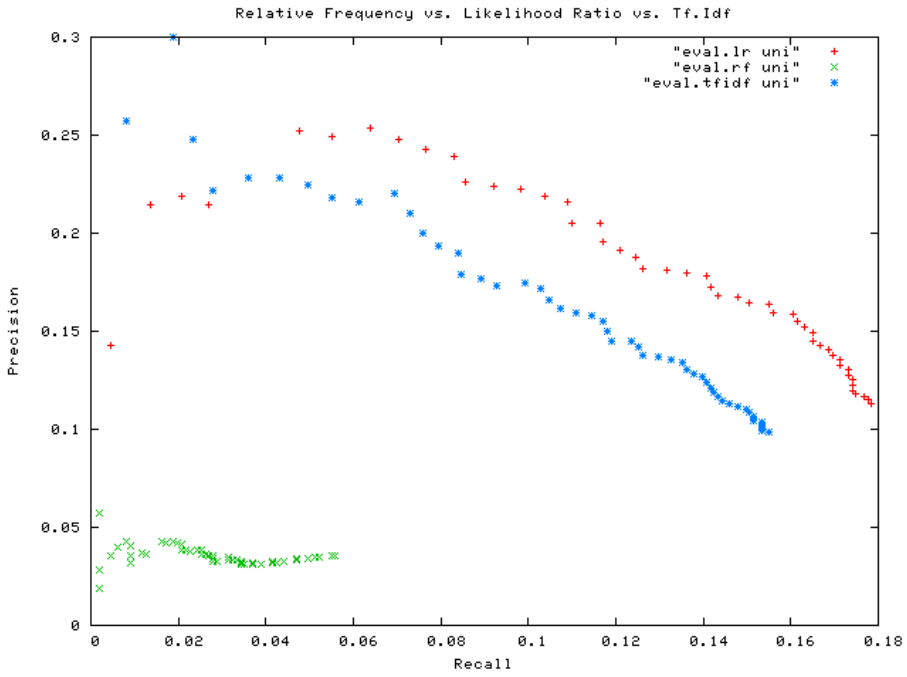
Jak vidíme z grafu 3.1, metoda relativních frekvencí se pro tento případ jeví jako nevhodná.

Přidání bigramů

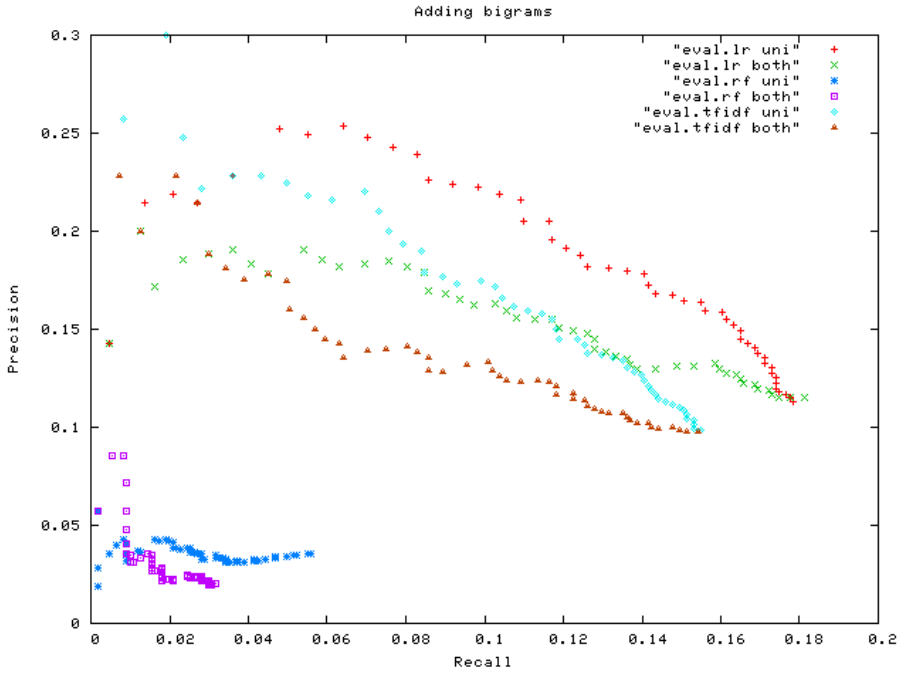
Přidání bigramů do výsledku (obrázek 3.2) má za následek horší přesnost systému. To je způsobeno tím, že jen málo z nalezených bigramů se dá považovat za dvojslovné pojmy.

Přidání bigramů s využitím lingvistické informace

Teprve odfiltrováním bigramů s využitím lingvistické informace získáváme lepší výsledky z bigramů (obrázek 3.3).



Obrázek 3.1: Srovnání metod na unigramech

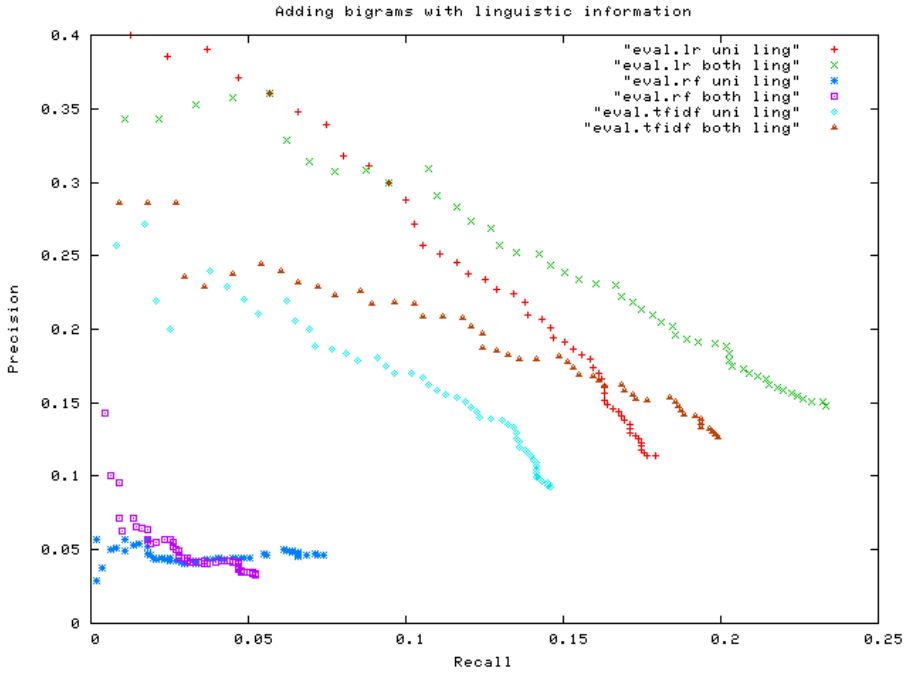


Obrázek 3.2: Vliv přidání bigramů

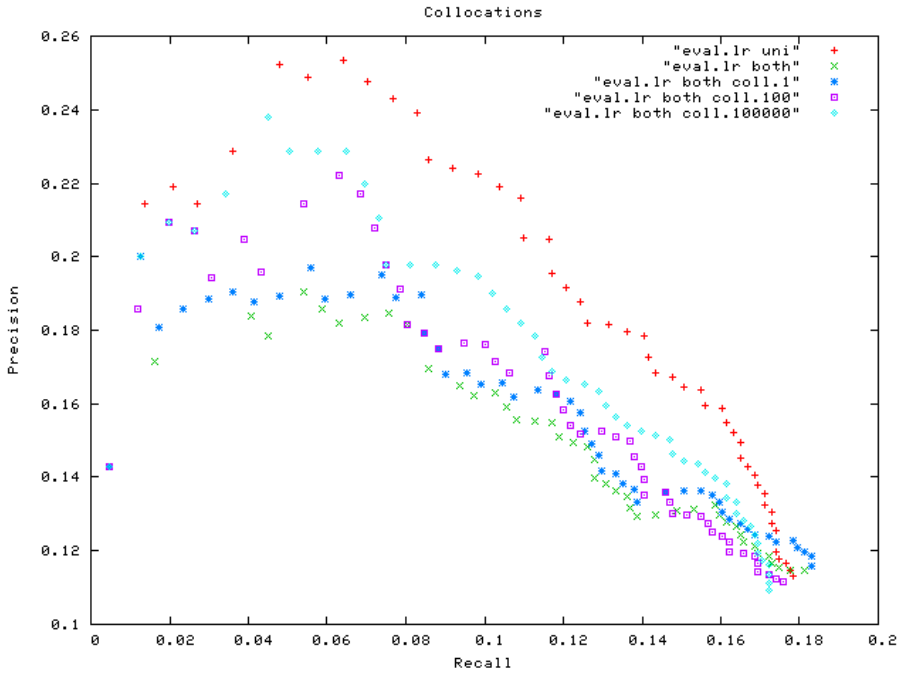
Práh testu kolokace

Vidíme (obrázek 3.4, že odfiltrováním bigramů, které neprojdou testem na kolokaci, můžeme marginálně zvýšit přesnost systému. Nicméně ne dost na to, aby překonala systém bez

[1] http://www.let.uu.nl/lt4el/



Obrázek 3.3: Vliv přidání bigramů s využitím lingvistické informace



Obrázek 3.4: Vliv nastavení prahu likelihoodu pro kolokace

bigramů. To by mohlo naznačovat, že se v ručně anotovaných datech mnoho dvojslovných pojmů nevyskytuje, ovšem metodou využitím lingvistické informace jsme již ukázali, že bigramy mohou zlepšit výsledek. V tomto případě se nám jeví využití značek jako vhodnější metoda pro určení bigramů než čistě statistický přístup.

Kapitola 4

Extrakce pojmů a vztahů

Za pojmy v této části práce budu považovat množinu termů. Pojem obsahuje termy, které jsou si navzájem významově podobné. Pro tento účel si nadefinujme vztah podobnosti mezi termy a nalezněme způsob, jak jej spočítat. Sémantický vztah podobnosti mezi termy se nazývá *synonymie*.

4.1 Linova míra podobnosti

Dekang Lin [8] definuje podobnost dvou objektů jako množství informace obsažené v tom, co mají objekty společné, děleno množstvím informace v popisu objektů.

Pro každý objekt je definována množina vlastností.

$$T(w) \quad \ldots \quad \text{množina vlastností slova } w \tag{4.1}$$
$$I(w, f) \quad \ldots \quad \text{Vzájemná informace mezi slovem } w \text{ a vlastností } f \tag{4.2}$$

Podobnost pak definuje jako:

$$sim_{lin}(w_1, w_2) = \frac{\sum_{f \in T(w_1) \cap T(w_2)} (I(w_1, f) + I(w_2, f))}{\sum_{f \in T(w_1)} I(w_1, f) + \sum_{f \in T(w_2)} I(w_2, f)} \tag{4.3}$$

pro nezáporná $I(w, f)$.

Pro použití na podobnost je potřeba nadefinovat vlastnosti, které použijeme pro popis objektů. Pro své experimenty jsem stejně jako v [8] použil přímé syntaktické vztahy.

4.1.1 Popis syntaktických vlastností

Slova mohou být s ostatními slovy v různých syntaktických vztazích. Myšlenka použití syntaktických vztahů pro míru podobnosti je založena na pozorování, že sémanticky podobná slova se často pojí se stejným slovesem, či jsou rozvíjena stejnými přídavnými jmény.

Jako vlastnost f slova w tedy definujeme

$$f \quad = \quad (r, w') \qquad \text{kde} \tag{4.4}$$
$$r \quad \ldots \quad \text{Typ sémantického vztahu} \tag{4.5}$$
$$w' \quad \ldots \quad \text{Slovo, které je se slovem } w \text{ ve vztahu } r \tag{4.6}$$

Zbývá vyjádřit vzájemnou informaci mezi w a f.

$$I(w, f) = \log \frac{P(w \cap f)}{P(w)P(f)} \tag{4.7}$$

Pravděpodobnosti budeme odhadovat maximální věrohodností (MLE). Proto si nadefinujeme četnosti:

$$\|w, r, w'\| \quad \ldots \quad \text{Absolutní četnost výskytu slov } w \text{ a } w' \text{ ve vztahu } r \tag{4.8}$$

$$\|*, r, w'\| \quad \equiv \quad \sum_{w} \|w, r, w'\| \tag{4.9}$$

$$\|w, r, *\|, \|*, r, *\| \quad \ldots \quad \text{obdobně} \tag{4.10}$$

Předpokládejme, že výskyt w a w' jsou na sobě nezávislé a oboje závisí na typu syntaktického vztahu r. Prakticky tedy počítáme s každým typem syntaktického vztahu zvlášť.

$$I(w, f) = I(w, (r, w')) = \log \frac{P(w \cap w'|r)}{P(w|r)P(w'|r)} \tag{4.11}$$

Pak můžeme použít MLE odhady:

$$P_{MLE}(w \cap w'|r) \quad = \quad \frac{\|w, r, w'\|}{\|*, r, *\|} \tag{4.12}$$

$$P_{MLE}(w|r) \quad = \quad \frac{\|w, r, *\|}{\|*, r, *\|} \tag{4.13}$$

$$P_{MLE}(w'|r) \quad = \quad \frac{\|*, r, w'\|}{\|*, r, *\|} \tag{4.14}$$

$$I(w, (r, w')) = \log \frac{\|w, r, w'\| \|*, r, *\|}{\|w, r, *\| \|*, r, w'\|} \tag{4.15}$$

4.2 Experiment

Pro experiment extrakce sémantického vztahu podobnosti jsem použil část gigakorpusu (afp). Korpus byl syntakticky analyzován a ze syntaktické analýzy byly vyextrahovány četnosti trojic $\|w, r, w'\|$.

slovo	podobnost
vice president	0.157
representative	0.144
secretary	0.141
secretary of state	0.129
speaker	0.126
senator	0.125
prince	0.118

Tabulka 4.1: Nejpodobnější slova k *president*

4.3 Shlukování termů

Pojmy můžeme chápat jako množinu termů, ve které mají všechny dvojice termů podobný význam. V předcházející části jsme si zavedli podobnost termů. Na základě této podobnosti můžeme shlukovat.

Použijeme hierarchické shlukování metodou zdola nahoru. Hierarchické shlukování vytváří strom shluků. Tomu odpovídá zobecňování pojmů, kde obecnější pojem vždy označuje nadmnožinu objektů pojmu konkrétnějšího.

Input: množina $X = \{x_1, \ldots, x_n\}$ objektů
funkce sim: $\mathcal{P}(X) \times \mathcal{P}(X) \to \mathbb{R}$
for $i := 1..n$ **do**
  $c_i := \{x_i\}$
end
$C := \{c_1, \ldots, c_n\}$
$j := n + 1$
while $|C| > 1$ **do**
  $(c_{n_1}, c_{n_2}) := \arg\max_{(c_u, c_v) \in C \times C} sim(c_u, c_v)$
  $c_j := c_{n_1} \cup c_{n_2}$
  $C := C \setminus \{c_{n_1}, c_{n_2}\} \cup \{c_j\}$
  $j := j + 1$
end

Algoritmus 1: Algoritmus shlukování zdola nahoru

4.3.1 Výsledky shlukování

Pro shlukování byla použita slova, která se v datech vyskytla alespoň 100 krát. Ukázka úseku výsledného dendrogramu je vidět na obrázku 4.1.

Výsledek by se dal označit za uspokojivý a odpovídá běžné představě o rozdílech mezi slovy jako například *president*, *minister* a slovy v druhé části shluku jako například *militant*, *separatist*, či *terrorist*.

Shlukování můžeme použít také pro nalezení významů u víceznačných slov, jak popisují [12]. Ukažme si zjednodušený postup:

1. Zvolíme víceznačné slovo w.
2. Vybereme seznam N slov nejpodobnějších k w podle funkce sim.
3. Shlukujeme seznam.

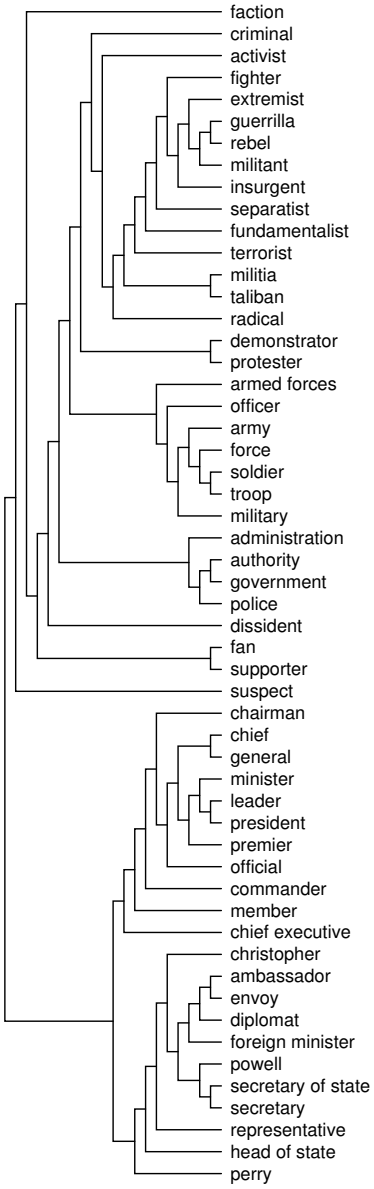
Na příklad pro slovo *suit*, což znamená jednak *oblek* a jednak *žaloba* vypadá shluk seznamu osmi nejbližších slov jako na obrázku 4.2.

4.4 Vlastnosti pojmů

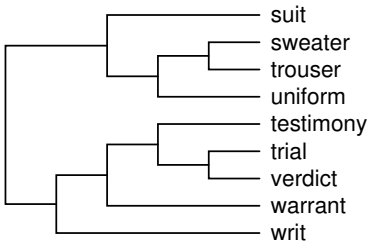
Shlukování podobných termů do pojmů děláme především pro to, abychom podchytili podstatu toho, co mají ony termy společného.

V této části prozkoumám způsob extrakce těchto podstatných rysů založených na statistice výskytů termů s jejich atributy, která je obdobná jako v [13].

Cílem tohoto snažení je dvojí:



Obrázek 4.1: Příklad shluku – Shluky obsahující slovo ‘terrorist’ a ’president’



Obrázek 4.2: Významy slova *suit*

- Nalezení intenzí daného pojmu – tedy nalezení takové množiny atributů, která vystihuje objekty označované pojmem. Příkladem budiž pojem ‘rose’, ‘tulip’, ‘violet’ a možné subjektivně přijatelné atributy, například ‘stinks’, ’is green’, ’is pretty’
- Nalezení pojmenování pojmu, tedy jednoslovného, či několika málo slovného, pojmenování vystihující množinu objektů označovanou pojmem. Například ‘flower’.

Jako atributy využijeme stejné atributy, které jsme použili pro shlukování v předcházející sekci.

Zavedeme funkci, která pro daný pojem a atribut bude udávat míru vhodnosti atributu k pojmu. Vhodnost zde definujeme subjektivně. Atributy můžeme rozdělit podle syntaktického vztahu. Například pro přídavná jména můžeme definovat vhodnost tak, že seřazením podle vhodnosti by sekvence atributů měla odpovídat odpovědi člověka na otázku: “Uveďte přídavná jména pojící se s tímto pojmem”.

Tuto funkci nazvěme F.

$$C \quad \ldots \quad \text{Pojem jako množina termů} \tag{4.16}$$
$$F(C, f) \quad \ldots \quad \text{Vhodnost atributu } f \text{ pro koncept } C \tag{4.17}$$

4.4.1 Podmíněná pravděpodobnost

Pro daný pojem C můžeme definovat vhodnost jako podmíněnou pravděpodobnost výskytu atributu f při výskytu pojmu C. Podmíněná pravděpodobnost odpovídá formálnímu vyjádření myšlenky, že vhodný atribut je takový, který se s pojmem pojí nejčastěji.

$$F_{cp}(C, f) = P(f|C) \tag{4.18}$$

Podmíněnou pravděpodobnost můžeme odhadnout odhadem maximální vhodností (MLE). Tu vypočteme z následujících četností:

$$\|w, f\| \quad \ldots \quad \text{Absolutní četnost spolu-výskytu termu } w \text{ s atributem } f \tag{4.19}$$
$$\|w, *\| \quad \ldots \quad \sum_f \|w, f\| \tag{4.20}$$
$$\|*, w\| \quad \ldots \quad \sum_w \|w, f\| \tag{4.21}$$

Za výskyt pojmu budeme považovat výskyt každého z termů tvořících pojem. Pak vyjádříme odhad podmíněné pravděpodobnosti jako:

$$F_{cp}(C, f) = P_{MLE}(f|C) = \frac{\sum_{w \in C} \|w, f\|}{\sum_{w \in C} \|w, *\|} \tag{4.22}$$

Jmenovatel výše uvedeného zlomku závisí jen na daném pojmu, Vhodnost atributu výpočtem podmíněné pravděpodobnosti tedy odpovídá atributu, který se s daným pojmem pojí nejčastěji.

	f	$\neg f$
C	O_{11}	O_{12}
$\neg C$	O_{21}	O_{22}

Tabulka 4.2: Kontingenční tabulka pro χ^2 test

4.4.2 χ^2 test

Pokud si data srovnáme do tabulky 4.2

Význam jednotlivých polí tabulky je následující:

$$O_{11} \quad \ldots \quad \text{Počet spoluvýskytů některého termu z konceptu } C \text{ s atributem } f \tag{4.23}$$
$$O_{12} \quad \ldots \quad \text{Počet spoluvýskytů konceptu } C \text{ s atributem jiným než } f \tag{4.24}$$
$$O_{21} \quad \ldots \quad \text{Počet spoluvýskytů atributu } f \text{ s termy, které netvoří } C \tag{4.25}$$
$$O_{22} \quad \ldots \quad \text{Počet spoluvýskytů termů } t \notin C \text{ s atributy jinými než } f \tag{4.26}$$

Obecně můžeme spočítat hodnotu χ^2 jako

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{4.27}$$

Kde O_{ij} jsou pozorované hodnoty a E_{ij} očekávané hodnoty. Pokud N je celkový počet všech dvojic spoluvýskytů, tak pro konkrétní případ tabulky 2×2 můžeme spočítat hodnotu χ^2, kterou budeme považovat za vhodnost:

$$F_{chi} = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})} \tag{4.28}$$

4.4.3 Test poměrem věrohodností

Dále vyzkoušíme test poměrem věrohodnosti. Formulujme hypotézy:

$$H_0 : \quad P(f|C) = p = P(f|\neg C) \tag{4.29}$$
$$H_1 : \quad P(f|C) = p_1 \neq p_2 = P(f|\neg C) \tag{4.30}$$

Odhadem maximální věrohodnosti odhadneme pravděpodobnosti p, p_1 a p_2. Za využití symbolů z předcházející sekce můžeme vyjádřit:

$$p = \frac{O_{11} + O_{21}}{N} \qquad p_1 = \frac{O_{11}}{O_{11} + O_{12}} \qquad p_2 = \frac{O_{21}}{O_{21} + O_{22}} \tag{4.31}$$

Pro výpočet věrohodnosti opět použijme binomický model.

$$L_0 = \mathcal{L}(p|O_{11}, O_{11} + O_{12}).\mathcal{L}(p|O_{21}, O_{21} + O_{22}) \tag{4.32}$$
$$L_1 = \mathcal{L}(p_1|O_{11}, O_{11} + O_{12}).\mathcal{L}(p_2|O_{21}, O_{21} + O_{22}) \tag{4.33}$$

kde $\mathcal{L}$ je věrohodnost binomického modelu 3.11. Poměr věrohodností pak určuje kolikrát je hypotéza 0 pravděpodobnější:

$$\lambda = \frac{L_0}{L_1} \tag{4.34}$$

Za vhodnost atributu budeme považovat $-2 \log \lambda$, protože tato hodnota je porovnatelná s χ^2.

$$F_{lr} = -2 \log \frac{L_0}{L_1} \tag{4.35}$$

4.4.4 Výpočet vzájemné informace

$$F_{mi} = \log \frac{P(f, C)}{P(f)P(C)} = \log \frac{\frac{O_{11}}{N}}{\frac{O_{11}+O_{21}}{N} \frac{O_{11}+O_{12}}{N}} \tag{4.36}$$

Jak upozorňuje Patrick Pantel v [13], hodnota vzájemné informace se klonění k termům s nízkou absolutní četností, proto tuto hodnotu násobí činitelem:

$$F_{midiscount} = \frac{O_{11}}{O_{11}+1} \frac{min(O_{11}+O_{21}, O_{11}+O_{12})}{min(O_{11}+O_{21}, O_{11}+O_{12})+1} \log \frac{\frac{O_{11}}{N}}{\frac{O_{11}+O_{21}}{N} \frac{O_{11}+O_{12}}{N}} \tag{4.37}$$

Pro nalezení vhodného pojmenování dále P. Pantel v [13] nepočítá hodnotu vzájemné informace mezi atributy pojmu, nýbrž počítá vzájemnou informaci mezi jednotlivými termy a jejich atributy a tuto hodnotu sečte přes všechny termy, které jsou součástí pojmu, tedy:

$$F_{midiscountsum}(C, f) = \sum_{t \in C} F_{midiscount}(\{t\}, f) \tag{4.38}$$

4.4.5 Experimenty

Cílem experimentů je zjistit, zda můžeme touto metodou nalézat atributy, které by se daly použít pro popis konceptů. Podívejme se nejprve, jak dopadne opačný pokus, tedy nalezení nejvhodnějších termů k danému atributu. Hledaným atributem pro tabulku 4.3 bylo přídavné jméno 'beautiful', hledáme tedy pojmy, které jsou krásné.

#	F_{cp}	F_{chi}	F_{lr}	$F_{midiscount}$
1	woman	scenery	woman	scenery
2	game	sea island cotton	game	piece of music
3	day	piece of music	scenery	christmas present
4	country	temptress	place	waterfall
5	place	omelette	girl	pin-up

Tabulka 4.3: Vhodné termy k atributu beautiful$_A \rightarrow_{mod} X$

Pro určení pojmenování jsou některé atributy důležitější než jiné. Každá z uvedených funkcí odhaluje slovo 'president', což subjektivně určíme jako správné pojmenování. U tohoto příkladu nás to nepřekvapí, když se podíváme na výsledky výpočtu podmíněné pravděpodobnosti, ze které plyne, že atribut $X \rightarrow_{person}$ president$_N$ je nejčastější.

To odpovídá vzoru, v datech častému, jako například:

#	F_{cp}	F_{chi}	F_{lr}	$F_{midiscountsum}$
1	$X \rightarrow_{person}$ president$_N$	$X \rightarrow_{person}$ president$_N$	$X \rightarrow_{conj}$ president$_N$	$X \rightarrow_{person}$ president$_N$
2	$X \rightarrow_{conj}$ president$_N$	predecessor$_N \rightarrow_{nn} X$	counterpart$_N \rightarrow_{nn} X$	ex-president$_N \rightarrow_{nn} X$
3	$X \rightarrow_{s}$ be$_{VBE}$	ex-president$_N \rightarrow_{nn} X$	predecessor$_N \rightarrow_{nn} X$	predecessor$_N \rightarrow_{nn} X$
4	US$_N \rightarrow_{nn} X$	laureate$_A \rightarrow_{mod} X$	laureate$_A \rightarrow_{mod} X$	$X \rightarrow_{under}$ vice president$_N$
5	counterpart$_N \rightarrow_{nn} X$	counterpart$_N \rightarrow_{nn} X$	ex-president$_N \rightarrow_{nn} X$	$X \rightarrow_{appo}$ predecessor$_N$

Tabulka 4.4: Vhodné atributy k pojmu {'Bill Clinton', 'George Bush', 'Jimmy Carter', 'Ronald Reagan'}

President Bill Clinton gave a speech at ...

Tento atribut chybí na prvních místech u výpočtu poměru věrohodnosti, z čehož bychom mohli usuzovat, že existují termy, pro které je tento atribut také významný, nebo-li že pojem, který bychom nazvali 'president' není úplný.

Podle [13] má význam použít jen určitou množinu syntaktickým vztahů a z nich pak sečíst skóre všech atributů sdílejících stejné slovo. Sčítat výsledné hodnoty funkcí F můžeme snadno udělat pro funkci F_{mi}, protože tato obsahuje hodnoty informace a sčítat informace má význam.

4.5 Hiearchie termů

Mezi podobnými slovy k termu 'jeep' nacházíme mimo jiné termy jako je 'van', 'lorry', či 'vehicle'. Žádné z nich nelze považovat za synonyma. Slovo 'vehicle' bychom mohli považovat za hypernymum, tedy slovo označující abstraktěnšjí pohled na slovo 'jeep'. Spolu s ostatními slovy z tohoto seznamu můžeme považovat množinu slov 'jeep', 'lorry' a 'van' za hyponyma slova 'vehicle'. Pokusme se zjistit, zda je toto uspořádání patrné z dat a zda je možné uspořádat tato slova automaticky.

4.5.1 Spoluvýskyt atributů

Sanderson a Croft [14] popisují metodu, jak se dá vyvodit, že slovo x je hypernymum slova y:

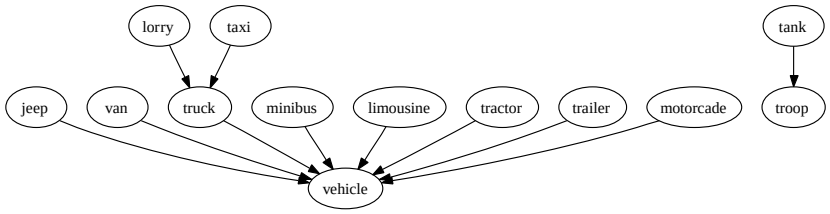
Slovo x zahrnuje slovo y, pokud dokumenty obsahující y jsou podmnožinou dokumentů obsahujících x, nebo-li:

$$P(x|y) = 1 \wedge P(y|x) < 1 \tag{4.39}$$

Tento principu zobecnili Fotzo a Gallinari [6], podle kterých je možné ze spoluvýskytu slov v dokumentech odhadnout následující:

Slovo x je hypernymum slova y právě tehdy když

$$P(x|y) \geq t \wedge P(y|x) < P(x|y) \tag{4.40}$$



Obrázek 4.3: Hiearchie slov podobných ke slovu ‘jeep’ ($t = 0.6$)

$$n(x,y) \quad \text{označuje počet dokumentů, kde se slova x i y vyskytly současně} \tag{4.41}$$

$$n(y) \quad \text{označuje počet dokumentů obsahující slovo y} \tag{4.42}$$

$$P(x|y) = \frac{n(x,y)}{n(y)} \tag{4.43}$$

$$\tag{4.44}$$

$$t \quad \textit{parametr} \tag{4.45}$$

Vyjdeme ze stejného vztahu, jako při výpočtu spoluvýskytu podle Fotzo a Gallinariho (viz rovnice 4.40). Na místo spoluvýskytu termů v dokumentech využijme vyextrahované syntaktické atributy.

Výsledkem bude binární relace nad termy H_c, kterou budeme chápat jako sémantický vztah *hypernymy*.

$$H_c = \{(x,y)|P(x|y) \geq t \wedge P(y|x) < P(x|y)\} \tag{4.46}$$

$$P(x|y) = \frac{\sum_{f \in T(x) \cap T(y)} min(\|x,f\|, \|y,f\|)}{\sum_{f \in T(y)} \|y,f\|} \tag{4.47}$$

kde $T(x)$ je množina atributů termu x a $\|x,f\|$ je absolutní četnost spoluvýskytu termu x s atributem f. t je hodnota prahu, kterou je potřeba odhadnout experimentálně.

Pro daný term x můžeme efektivně vyhledat jeho hypernymum tím, že spočítáme hodnoty $P(x|y)$ a $P(y|x)$ pro všechny termy y, které jsou podobné termu x na základě funkce *sim* a na nich vypočítáme relaci H_c

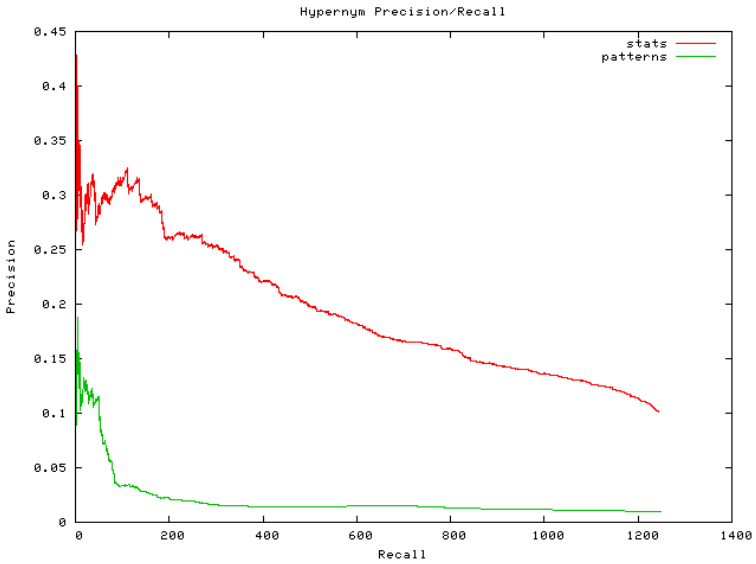
Pro nejbližších 15 termů ke slovu ‘jeep’ bylo tímto způsobem vytvořena hiearchie, kterou lze vidět na obrázku 4.3. Slovo ‘vehicle’ správně vychází jako hypernymum. Chybně vychází například slovo ‘taxi’ jako hyponymum slova ‘truck’.

Tato metoda na experimentálních datech z Gigacorpusu správně nalezne například:

$$\text{Clinton, Bush} \rightarrow \text{president}$$

Ovšem mnoho chyb je podobných jako:

$$\begin{aligned} \text{six, five, four} &\rightarrow \text{three} \\ \text{Germany, France, Japan} &\rightarrow \text{United States} \end{aligned}$$



Obrázek 4.4: Vyhodnocení extrakce vztahu hypernym na WordNetu

Zde dochází k tomu, že obecný pojem pro tyto termy se buď nevyskytuje vůbec, nebo se nevyskutuje se stejnými syntaktickými vztahy. Z množiny podobných slov je pak vybráno slovo, které je *kohyponymem*. Kohyponyma jsou slova, které sdílejí stejné hypernymum. Srovnejme například běžné fráze:

Six people died in . . .

Large number of people are . . .

Obecné pojmenování 'number' se vyskytuje v jiném syntaktickém vztahu než použití konkrétních čísel.

Pro vyhodnocení použijme WordNet. Úlohu převedeme na problém hlednání dvojic slov x a y, kde x je hyponymum slova y. Můžeme tak vyhodnotit přesnost, kde za pravdivou dvojici slov budemem považovat takovou, pro kterou ve WordNetu existuje dvojice synsetů, které jsou v tranzitivním uzávěru relace HYPERNYM. Nastavením hodnoty prahu t můžeme dosáhnout různé poměry přesnosti proti pokrytí. Výsledek závislosti přesnosti na pokrytí je na obrázku 4.4 čárou označenou jako stats.

Výsledek je do jisté míry podhodnocen, protože mnoho dvojic, které se subjektivně zdají být v pořádku se nevyskytují ve WordNetu v hledaném vztahu. Při náhodně zvoleném vzorku 20 extrahovaných dvojic bylo subjektivně určena přesnost 0.5, při čemž dvě třetiny chybných dvojic by se dala označit za kohyponyma[1].

4.6 Vztahy mezi pojmy

Budeme se zabývat extrakcí sémantických vztahů z korpusu na základě výskytu dvou termů ve větě v určitém syntaktickém vztahu.

[1]Tento výsledek nemá pro zjevnou zaujatost žádnou praktickou hodnotu a uvádím jej zde jen pro zajímavost.

4.6.1 Pojmenování relace na základě sloves

Vojtěch Svátek a Martin Kavalec si ve své práci [7] všímají toho, že informace o relaci je obvykle nesena slovesem. Vybírají tedy slovesa vyskytující se v okolí obou konceptů, které mají tvořit vztah. Definují tedy míru AE (Above Expectation), která je tím větši, čím významnější je výskyt trojice (v, c_1, c_2) oproti předpokládanému nezávislému výskytu.

$$P(c_1 \wedge c_2|v) = \frac{|\{t_i|v, c_1, c_2 \in t_i\}|}{|\{t_i|v \in t_i\}|} \tag{4.48}$$

$$AE(c_1 \wedge c_2|v) = \frac{P(c_1 \wedge c_2|v)}{P(c_1|v).P(c_2|v)} \tag{4.49}$$

t_i je množina označovaná jako *transakce* a obsahuje v, c_1 a c_2, pokud se c_1 i c_2 vyskytly v blízkosti slovesa v.

Stejný princip můžeme použít pro náš případ, kde s využitím syntaktické analýzy můžeme pracovat přímo s argumenty sloves.

4.6.2 Slovesné argumenty

Snadno interpretovatelným syntaktickým vztahem mezi dvěma termy je vztah dvou termů jako argumentů jednoho slovesa. Jako příklad si uveďme jednoduchou větu:

John works for IBM

Termy 'John' a 'IBM' jsou zde v syntaktickém vztahu $John \rightarrow_s work \leftarrow_{for} IBM$

Extrahovat informaci, že John pracuje u IBM by byla úloha pro metody extrakci informací. Z hlediska určování sémantickýh vztahů nám jde jen o vztah, který říká, že společnosti zaměstnávají osoby. Podobně, jak jsme definovali pojem jako množinu termů, můžeme definovat sémantický vztah jako množinu syntaktických vztahů.

Shlukováním sloves se zabývá například [16].

Oproti syntaktickým atributům je dat, ze kterých bychom mohli počítat statistiku, relativně málo. Omezujeme na argumenty sloves, címž přehlížíme další slova ve větě. Pro získání informace je také potřeba, aby se pojmy, mezi kterými hledáme vztah, spoluvyskytovaly jako argumenty stejného slovesa.

$$\|C_1, l_1, v, l_2, C2\| \quad \ldots \quad \text{četnost syntaktického vztahu } C_1 \rightarrow_{l_1} v \leftarrow_{l_2} C_2 \tag{4.50}$$

$$v \quad \ldots \quad \text{sloveso} \tag{4.51}$$

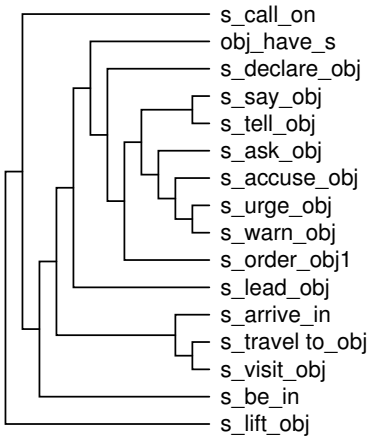
$$C_1, C_2 \quad \ldots \quad \text{pojmy} \tag{4.52}$$

$$l_1, l_2 \quad \ldots \quad \text{typ syntaktického vztahu mezi slovesem a termem} \tag{4.53}$$

Použijme následující algoritmus:

1. Vypočtěme N nejčetnějších vztahů mezi C_1 a C_2.
2. Shlukujme slovesa z těchto vztahů na základě funkce *sim*

Výsledkem může být například obrázek 4.5, který označuje shluk nalezených vztahů mezi pojmem obsahující term 'president' a pojmem obsahující term 'country'. Pro tvorbu pojmů bylo použito nejbližších 16 termů.



Obrázek 4.5: Vztahy mezi ‘president a ‘country’

4.6.3 Cesty v závislostním stromu

Výše popsaný přístup extrakce na základě slovesných argumentů můžeme zobecnit. Slovesné argumenty dávají do souvislosti jen slova, která jsou v přímé syntaktické závislosti na slovese. Do vztahu ale můžeme dát všechny dvojice slov ve větě. Každá dvojice slov ve větě je provázána syntaktickou cestou, tedy cestou v syntaktickém závislostním stromu.

Mezi cestami v závislostním stromu můžeme také definovat míru podobnosti podle [9]. Hledáním podobných syntaktických vztahů získáváme *parafráze*. Parafráze jsou fráze, které vyjadřují stejný sémantický vztah mezi slovy jiným syntaktickým prostředkem. Takovými parafrázemi může být například:

$$X \textit{ solves } Y$$
$$Y \textit{ is solved by } X$$

Při hledání významu frází narážíme na stejný problém jako při hledání pojmů. Ačkoliv shluk parafrází bychom mohli považovat za sémantiký vztah, rádi bychom takovému shluku přiřadili jednoduché a jasné pojmenování. Na místo hledání podobnosti mezi frázemi hledejme mezi podobnými frázemi takové, které jsme schopni snadno interpretovat. Takovými frázemi jsou vztahy mezi slovesnými argumenty.

Autoři [9] pro výpočet podobnosti používají obdobný princip jako rovnice 4.3. Každý výskyt syntaktického vztahu dosazuje slova do *slotů*, *slotX* a *slotY*. Pak vypočte podobnost zvlášť pro slotX a pro slotY:

$$sim_x(p_1, p_2) = \frac{\sum_{w \in T_x(p_1) \cap T_x(p_2,s)} I(p_1, w) + I(p_2, w)}{\sum_{w \in T_x(p1)} I(p_1, w) + \sum_{w \in T_x(p2)} I(p_2, w)} \tag{4.54}$$

Obdobně pro sim_y. Výsledná podobnost je pak geometrickým průměrem

$$S(p_1, p_2) = \sqrt{sim_x(p_1, p_2) sim_y(p_1, p_2)} \tag{4.55}$$

Pro hledání významu konkrétního vztahu hledejme nejpodobnější vztah z množiny vztahů tvaru $x \rightarrow_{l1} v \leftarrow_{l2} y$.

Příklad výsledku hledání parafrází pro frázi x solves y je v tabulce 4.5.

Rank	Pattern
1	$X \rightarrow_s cause_V \leftarrow_{obj} Y$
2	$Y \rightarrow_s concern_V \leftarrow_{obj} X$
3	$Y \rightarrow_s result_N \leftarrow_{of} X$
4	$X \rightarrow_s provoke_V \leftarrow_{obj} Y$
5	$X \rightarrow_s concern_V \leftarrow_{obj} Y$
6	$Y \rightarrow_s be_{VBE} \leftarrow_{pred} result_N \leftarrow_{of} X$

Tabulka 4.5: Parafráze k $X \rightarrow_s solve_V \leftarrow_{obj} Y$

4.7 Lexiko-syntaktické vzory

Pro daný sémantický vztah se můžeme pokusit o získání syntaktických vzorů. Využijme princip, který je popsán o vzorech Hearstové a konkrétně se jej pokusme aplikovat na závislostní strom.

1. Shromážděme dvojice pojmů, o kterých víme, že daný vztah platí.
2. Najděme věty, ve kterých se vyskytla taková dvojice.
3. Definujme vzor jako nejkratší cestu v závislostním grafu věty mezi oběma pojmy.

Pro vzory definujeme standardní míry přesnosti a pokrytí:

tp ... Počet nalezených dvojic, které jsou v daném sémantickém vztahu.
fp ... Počet nalezených dvojic, které nejsou v daném sémantickém vztahu.
fn ... Počet dvojic, které jsou v daném sémantickém vztahu, ale nebyly nalezeny, ačkoliv se v textu vyskytly.

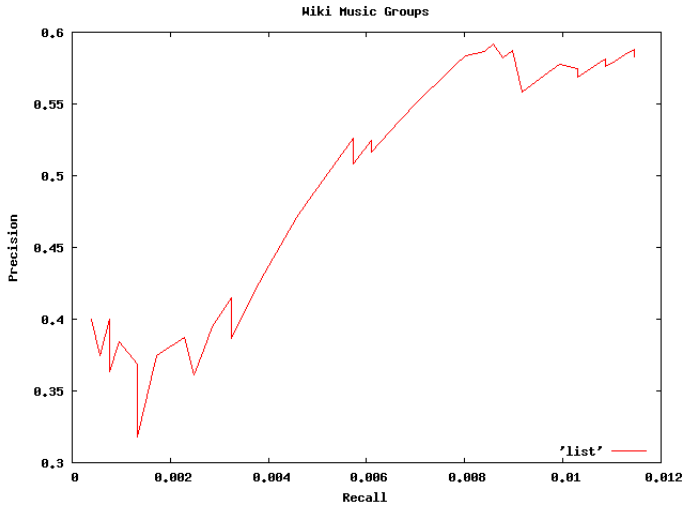
$$\text{přesnost} = \frac{tp}{tp + fn} \tag{4.56}$$

$$\text{pokrytí} = \frac{tp}{tp + fp} \tag{4.57}$$

Dopouštíme se zde troufalého předpokladu, že vybrané věty skutečně popisují náš sémantický vztah a že tedy výskyt dvojice slov splňující vztah v dané větě není jen shoda okolností, či například výskyt jiného významu stejného slova ve větě.

Nalezené vzory seřaďme podle přesnosti na trénovacích datech a definujme parametr t. Jako výsledné vzory pak považujme jen ty, které dosáhly přesnosti větší než hodnota parametru t.

Jako první experiment byl zvolen vztah hypernymum. Články gigacorpusu byly rozděleny na trénovací a testovací část. Pro určení, zda mezi dvěma slovy platí vztah hypernymie byl použit WordNet, konkrétně tranzitivní uzávěr relace HYPERNYM. Z trénovacích dat se pak vzaly jen ty vzory, pro které byl nalezen vztah alespoň pětkrát. Na základě přesnosti na trénovacích datech byly vzory seřazeny. Výsledný graf přesnosti na pokrytí je vidět na grafu 4.4 čárou označenou jako patterns. Ukazuje se, že tato metoda není příliš vhodná pro zvolená data.



Obrázek 4.6: Přesnost a pokrytí

Uvedenou metodu lze teoreticky použít na jakýkoliv sémantický vztah. Použijme jej tedy pro nějaký netaxonomický vztah:

Pro experiment byly vzaty články z anglické verze Wikipedie o hudebních skuipnách. Trénovací sada obsahuje 2054 dokumentů, testovací 2054 jiných dokumentů. Jako relace byla vybrána relace has_album, což je relace mezi hudební skupinou a názvem alba. Tyto relace byly získány z hudební databáze musicbrainz[2]. Jako výskyt relace bylo považováno to, že se vyskytl název skupiny a jejího alba v jedné větě.

Výsledné hodnoty testu jsou na obrázku 4.6.

[2]http://musicbrainz.org/

Kapitola 5

Architektura, návrh a implementace

Metody extrakce sémantických vztahů a k nim provedené experimenty popsané v předchozí části jsou jistě zajímavé samy o sobě. Pro lepší orientaci a snad i využití výsledků navrhněme aplikaci, která umožní pohodlné zobrazení výsledků, či bude možné ji použít pro dolování textových dat z dokumentů.

5.1 Případy užití

5.1.1 Thesaurus

Thezaury jsou slovníky obsahující sémantické vztahy, především synonyma. Program by měl nabízet možnost výběru slova a zobrazit slova v daném sémantickém vztahu.

5.1.2 Extrakce klíčových slov

Pro danou množinu dokumentů by měl program automaticky vytvořit seznam klíčových slov.

5.1.3 Tvorba konceptuálních map

Z dané množny dokumentů by měl program napomáhat tvorbě konceptuální mapy.

- Extrakce zajímavých pojmů
- Extrakce vztahů mezi pojmy

5.2 Architektura

Jazyková data mohou být velice rozsáhlá. Bylo by nepraktické, aby uživatel musel mít tato data na své pracovní stanici. Proto bude lepší, aby architektura byla založena na modelu klient/server. Klient se serverem budou komunikovat jasně definovaným protokolem.

5.3 Návrh komunikačního protokolu

Komunikačním protokol je protokol pro komunikaci mezi klientem a serverem. Protokol musí být:

- použitelný v síti internet;
- rozšiřitelný; Přidání nových funkcí do klienta či serveru nesmí znamenat kompletní změnu protokolu.
- otevřený; Protokol musí být snadno implementovatelný jakoukoliv třetí stranou, například jinými klienty.

Nechť základem protokolu je N3 nad HTTP (text/rdf+n3). Takový protokol bude splňovat požadavky.

5.4 Návrh datových struktur

5.4.1 Reprezentace závislostního stromu

Výsledkem syntaktické analýzy je závislostní strom. Požadavky na reprezentaci tak jsou následující:

- Uzly stromu jsou slova.
- Hranami stromu jsou syntaktické závislosti mezi slovy.
- Uzly mohou nést libovolnou informaci a přenášet tak sémantickou informaci získanou z různých fázích předzpracování.
- Musí být snadné a rychlé vyhledat předem zadané podstromy.

XML je běžný nástroj pro práci s dokumenty, které mají stromovu strukturu. Má definované programové rozhraní (Document Object Model) i dotazovací jazyk (XPath, XQuery). Pokusme se tedy využít XML technologií pro práci se závislostními stromy.

Základem budiž elementy dvou typů:

- *node*; Uzel reprezentující slovo. Jeho atributy jsou
  - *category*; Slovní druh.
  - *text*; Slovní lemma.
- *link*; Reprezentuje závislost mezi slovy. Jeho atributy:
  - *type*; Vztah mezi rodičovským *node* elementem a následnickým *node* elementem.

Elementy typu node budou obsahovat následníky typu link, a ty zase následníky typu node, což vytváří stromovu strukturu.

5.4.2 Návrh databáze

Vyextrahovaná data je potřeba vhodně uložit. I hodně dat se vhodně dá uložit pomocí programů, kterým se lidově říká SQL databáze [1].

Databáze bude tvořena ze dvou částí:

1. databáze pozadí; Data extrahovaná z velkého korpusu považována za pozadí (viz kapitolu *extrakce zajímavých termů*).
2. databáze dokumentů; Data extrahovaná z dokumentů (viz případ užití *tvorba konceptuálních map*.

Základními entitami budiž tyto:

1. Slovo; Každé slovo s informací o slovním druhu.
2. Term; Označuje potenciálně víceslovný termín. O každém termu je potřeba vědět, ve kterém dokumentu a kde se vyskytl. Na rozdíl od slova je term zajímavý jen pro databázi dokumentů.
3. Relace; Pojmenované vztahy mezi termy.

5.4.3 XQuery jako dotazovací jazyk nad stromy

Jazyk XQuery je standardní jazyk pro výběr obsahu XML dokumentů. Pro popis cest využívá jazyk XPath.

Například pro výběr všech podstatných jmen na základě formátu popsaného výše stačí jednoduchý výraz v jazyce XPath:

```
//node[@category='N']
```

Složitější příklad: Všechny podměty slovesa 'play' (hrát) které se vyskytnou v dokumentu vybereme následujícím výrazem:

```
//node[@category='V' and @text='play']/link[@type='s']/node
```

Pro lexiko-syntaktické vzory však nebude stačit hledat jeden typ uzlů podle nějakého kontextu, ale dávat do souvislosti více uzlů. Dotaz v jazyce XQuery, který demonstruje použití pro hledání trojice podmět-sloveso-předmět v dané množině dokumentů *collection* následuje.

```
for $v in collection//node[@category="V"]
for $s in $v/link[@type="s"]/node[@category="N"]
for $obj in $v/link[@type="obj"]/node[@category="N"]
return
    <svo>
        <v>distinct-values($v/@text)</v>
        <obj>distinct-values($obj/@text)</obj>
        <s>distinct-values($s/@text)</s>
    </svo>
```

[1] Ale kterým my říkáme "systémy pro řízení báze dat"

5.5 Návrh serveru

Úkolem serverové části budiž odpovídání na dotazy klienta. Předpokládá se, že většina informací z textu bude předzpracována ještě před spuštěním serveru. Zpracování rozsáhlých textů může být paměťově i časově náročná operace. Na druhou stranu nelze předpočítat veškeré možné dotazy předem, protože paměťová náročnost uložení všech možných odpovědí je příliž velká. Bude proto potřeba nalézt vhodnou mez, jak s ohledem na dobu vyřízení požadavku, tak i s ohledem na diskový prostor serveru, přičemž oba ohledy jsou vzájemně protichůdné.

Jak již bylo v úvodu této kapitoly naznačeno, server by měl pracovat jednak s pozadím, což bude obvykle velký korpus, a s jednotlivými množinami dokumentů, kterou nazveme popředí. Dokumenty popředí jsou ty objekty, ze kterých chceme extrahovat informaci, nazvěme je tedy MO (Mining Objects). Předpokládejme, že server by měl umět ukládat více MO najednou a všechny budou sdílet jedno pozadí. Pro identifikaci MO použijme jednoznačných identifikátorů, které nazvěme MO_{URI}.

5.5.1 Návrh funkcí

Návrh funkcí rozdělíme na funkce pro dotazování na informace extrahované z pozadí a na ty pracující s dokumenty popředí. Funkce pracující s pozadím dávají informace o slovech (Word). Funkce pracující nad extrahovanými dokumenty popředí pracují s termy (Term).

getSimilarWords :: Set(Word) → Set (Word, Similarity)

Pro danou množinu slov vyhledá množinu podobných slov.

getRelatedWords :: Set(Word) → Set (Word, Relation, Similarity)

Pro danou množinu slov vyhledá množinu pravděpodobných vztahů s jinými slovy.

getHypernym :: Word → Set (Word, Score)

Pro dané slovo vyhledá pravděpodobná hypernyma.

getHyponyms :: Word → Set (Word, Score)

Pro dané slovo vyhledá pravděpodobná hyponyma.

getImportantTerms :: MO_{URI} **→ Set (Term, Score)**

Vrátí nejzajímavější termy z dokumentů popředí daného MO_{URI}.

getSimilarTerms :: (MO_{URI}**, Term) → Set (Term, Score)**

Vrátí množinu termů podobných k danému termu.

getWordsSimilarToTerm :: (MO_{URI}**, Term) → Set (Word, Score)**

Vrátí množinu slov z pozadí podobných danému termu.

getRelatedTerms :: (MO_{URI}, Term) → Set (Term, Relation, Score)

Pro daný term vrátí množinu pravděpodobných vztahů s jinými termy.

5.6 Návrh klienta

Hlavní úlohou klientské části je poskytování uživatelského rozhraní k službám serveru.

5.7 Implementace

5.7.1 Server

Serverová část byla implementována v jazyce Python s využitím knihoven:

- NLTK (Natural Language ToolKit)[2]; Pro přístup k WordNetu a pro rozdělení textu na věty.
- Twisted.Web[3]; Jako jednoduchý HTTP server.
- RDFlib[4]; Pro zpracování a tvorbu zpráv protokolu mezi serverem a klientem.
- sqlite; Pro uložení dat.
- bsddb; Pro uložení slovníků (id slova na text slova).
- CMPH[5]; Pro uložení slovníků perfektní hashovací funkcí (text slova na id). Rozhraní pro jazyk Python bylo vytvořeno pomocí nástroje swig.
- lxml[6]; Pro zpracování XML formátu závislostního stromu.
- PyLucene[7]; Pro indexování vět při hledání termů.

5.7.2 Předzpracování

Samostatnou komponentou serveru je předzpracovací část. Vstupem jsou jednotlivé dokumenty ve formátu prostého textu. Výstupem jsou statistiky a předpočítané hodnoty.

Jednotlivé fáze předzpracování můžeme zjednodušeně rozepsat:

1. rozdělení dokumentů na jednotlivé věty;
2. zpracování vět syntaktickým analyzátorem MiniPar;
3. převedení výsledku programu MiniPar do XML;
4. provedení XSLT transofrmací pro převedení předložek z uzlů do hran; Tato operace převede například kus stromu *solution* $\rightarrow_{mod}$ *to* $\rightarrow_{pcomp}$ *problem* na *solution* $\rightarrow_{to}$ *problem*.

[2] http://nltk.sourceforge.net/
[3] http://twistedmatrix.com/trac/wiki/TwistedWeb
[4] http://code.google.com/p/rdflib/
[5] http://cmph.sourceforge.net/
[6] http://codespeak.net/lxml/
[7] http://pylucene.osafoundation.org/

5. identifikace možných termů; Jako uzly typu N (Noun) případně rozvité.

6. extrakce vlastností pro každý term; Jako vlastnosti jsou použity syntaktické vlastnosti, tzn. okolní hrany a uzly v syntaktickém závislostním stromu.

7. nalezení výskytů syntaktických vzorů; Základním vzorem je vzor pro hledání sloves a jejich argumentů.

8. předpočítání důležitosti všech nalezených termů (metodou vzájemné informace s využitím četností získaných z pozadí) a vzájemnou podobnost na základě podobných vlastností (metodou Linovy podobnosti);

Metoda nalezení termů nemusí nalézt všechny důležité termy. Uživatel má možnost zadat jako term libovolný řetězec. V takovém případě se využije fulltextové vyhledávání vět obsahující hledaný textový řetězec. Následně se vyhledá množina uzlů v syntaktickém stromu zahrnujících hledaná slova. Z této množiny A se odhadne hlava fráze jako taková množina uzlů B, jejímiž předky nejsou jiné uzly z množiny A. Určení hlavy je důležité pro sběr syntaktických vlastností a pro použití syntaktických vzorů.

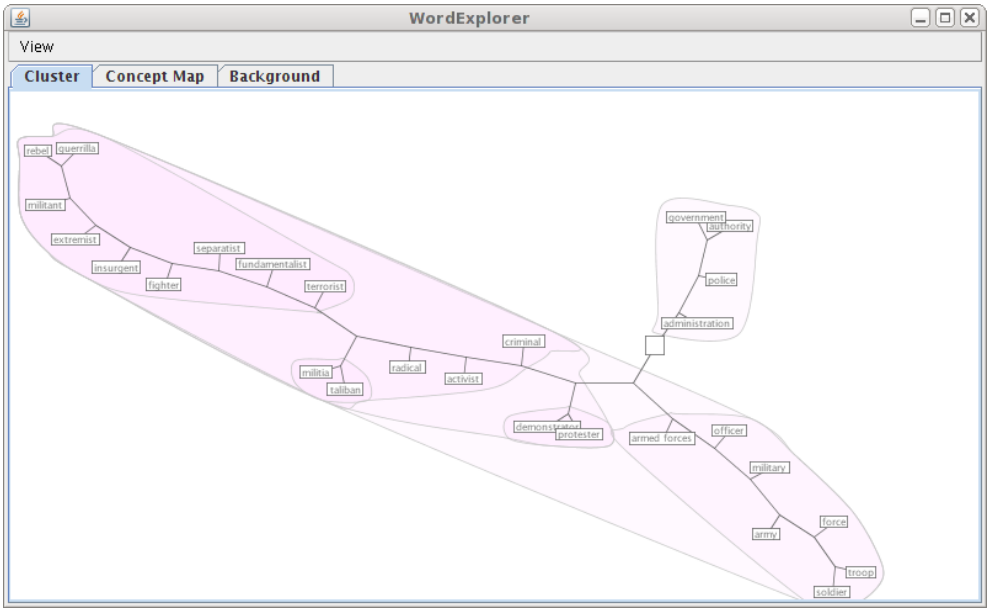
5.7.3 Klient

Klientská část byla implementována v jazyce Java s využitím knihoven:

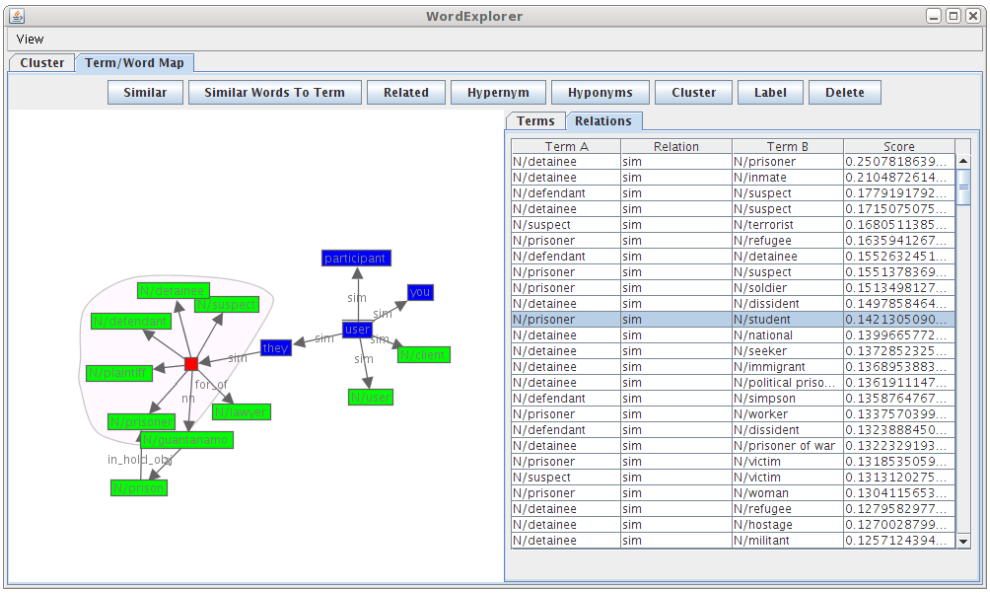
- Prefuse[8]; Pro vizualizaci grafů.
- Apache Jakarta Commons HttpClient[9]; Pro HTTP spojení se serverem.
- Jena[10]; Pro zpracování a tvorbu zpráv protokolu.

Ukázka snímku obrazovky z klienta je na obrázku 5.1 zobrazující prohlížeč slovního shluku. Druhý snímek ukazuje mapu termů a slov 5.2.

[8] http://prefuse.org/
[9] http://hc.apache.org/httpclient-3.x/
[10] http://jena.sourceforge.net/



Obrázek 5.1: Snímek prohlížeče shluků



Obrázek 5.2: Snímek mapy termů a slov

Kapitola 6

Závěr

Oblast extrakce sémantických vztahů z textu je velmi široká. Tato práce se především zaměřila na oblast související s učením ontologií, což ovlivnilo záběr zkoumaných metod.

Bylo implementováno a v rámci možností vyhodnoceno několik metod pro extrakci termů a různých typů sémantických vztahů. Na základě nich byl pak vytvořen systém pro extrakci sémantických vztahů z textu a souvisejícího uživatelského rozhraní.

Tato práce nepřichází s žádnou novou metodou extrakce sémantických vztahů. Přínos této práce spočívá spíše v aplikaci známých metod. Velkým problémem pro statistické metody zpracování přirozeného jazyka je potřeba velkého množství dat. Přínosem této práce je snaha o využití dat z velkého korpusu jako pozadí pro extrakci vztahů z popředí.

Možností dalšího vývoje projektu je mnoho. Nabízí se především použití dalších metod, které v této práci nebyly zmíněny, jako je celá třída metod založených na strojovém učení a metody extrakce informací.

Literatura

[1] *WordNet*. The MIT Press, 1998, ISBN 0-262-06197-X.
URL http://wordnet.princeton.edu

[2] Baroni, M.; Bernardini, S.: BootCaT: Bootstrapping Corpora and Terms from the Web.

[3] Cimiano, P.: *Ontology Learning an Population from Text*. Springer, 2006, ISBN 978-0-387-30632-2.

[4] Cimiano, P.; Pivk, A.; Schmidt-Thieme, L.; aj.: Learning taxonomic relations from heterogeneous sources. 2004.

[5] Feldman, R.; Sanger, J.: *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, December 2006, ISBN 0521836573.

[6] Fotzo; Gallinari: Learning Generalization/Specialization Relations between Concepts – Application for Automatically Building Thematic Document Hierarchies. 2004.

[7] Kavalec, M.; Svátek, V.: A Study on Automated Relation Labelling in Ontology Learning. In *Ontology Learning from Text: Methods, Evaluation and Applications*, 2005.

[8] Lin, D.: Automatic Retrieval and Clustering of Similar Words. In *COLING-ACL*, 1998, s. 768–774.

[9] Lin, D.; Pantel, P.: DIRT – discovery of inference rules from text. In *Knowledge Discovery and Data Mining*, 2001, s. 323–328.
URL citeseer.ist.psu.edu/lin01dirt.html

[10] Manning, C. D.; Schütze, H.: *Foundations of Statistical Natural Language Processing*. The MIT Press, June 1999, ISBN 0262133601.

[11] Ogden, C. K.; Richards, I. A.: The Meaning of Meaning. 1923.

[12] Pantel, P.; Lin, D.: Discovering Word Senses from Text. 2002.
URL http://www.patrickpantel.com/Download/Papers/2002/kdd02.pdf

[13] Pantel, P.; Ravichandran, D.: Automatically Labeling Semantic Classes. In *Human Language Technology / North American Association for Computational Linguistics (HLT/NAACL-04)*, 2004, s. 321–328.

[14] Sanderson; Croft: Deriving concept hierarchies from text. 1999.

[15] Schmidt, M.: How to Search the Future Web. In *Sborník prací konference a soutěže Student EEICT*, 2008.

[16] Walde, S.: Clustering Verbs Semantically According to their Alternation Behaviour. 2000.
URL citeseer.ist.psu.edu/article/walde00clustering.html