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

ANALÝZA KVALITY PŘEVODU ELEKTRONICKÝCH SLOVNÍKŮ

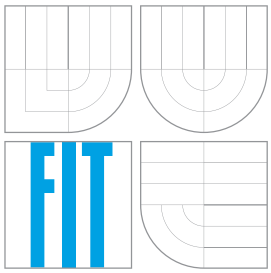
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETRA STEHLÍKOVÁ

BRNO 2013

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

ANALÝZA KVALITY PŘEVODU ELEKTRONICKÝCH SLOVNÍKŮ

QUALITY ANALYSIS OF ELECTRONIC DICTIONARIES TRANSFORMATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PETRA STEHLÍKOVÁ
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAN KOUŘIL
SUPERVISOR

BRNO 2013

Abstrakt

Práce se zabývá elektronickými slovníky, jejich formáty a analýzou správnosti jejich převodu z jiných formátů. Práce popisuje podrobněji formát Lexical Markup Framework. Dále se věnuje možnostem analýzy převodů (především latentní sémantické analýze) a nástrojům pro ni použitých. Na základě těchto teoretických znalostí byly vytvořeny skripty v jazyce Python pro analýzu slovníků ve formátu Lexical Markup Framework.

Abstract

The bachelor's thesis deals with electronic dictionaries, their formats and quality analysis of their conversions. The thesis describes Lexical Markup Framework format in detail. It also discusses the capabilities of advanced algorithms such as LSA for conversion quality analysis and the tools that can be used for the analysis. Based on this theoretical knowledge the scripts in Python language were created to analyze dictionaries in Lexical Markup Framework format.

Klíčová slova

elektronické slovníky, Lexical Markup Framework, kosinová podobnost, latentní sémantická analýza, singulární dekompozice

Keywords

electronic dictionaries, Lexical Markup Framework, cosine similarity, latent semantic analysis, singular value decomposition

Citace

Petra Stehlíková: Analýza kvality převodu elektronických slovníků, bakalářská práce, Brno, FIT VUT v Brně, 2013

Analýza kvality převodu elektronických slovníků

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Jana Kouřila a uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....
Petra Stehlíková
15. května 2013

Poděkování

Ráda bych poděkovala Ing. Janu Kouřilovi za cenné rady a vedení mé práce.

Obsah

Kapitola 1

Úvod

Při převodu slovníků mezi různými formáty vznikají chyby. Jedná se zejména o pravopisné chyby způsobené špatným rozpoznáním znaku při převodu tištěných slovníků do elektronické podoby. Dále se mohou objevit chyby sémantické, kdy jsou jednotlivým významům slova špatně přiřazeny doplňující příklady či vysvětlení. Tato práce se snaží slovníky převedené do formátu Leximal Markup Framework analyzovat a tyto chyby odhalit.

V kapitole 2 budou představeny některé formáty elektronických slovníků založené na Extensible Markup Language, především formát Lexical Markup Framework. Kapitola 3 se zabývá nástroji, schématy a metrikami, které lze použít pro syntaktickou i sémantickou analýzu informací obsažených ve slovnících. V kapitole 4 bude nastíněna implementace jednotlivých analýz a v poslední kapitole (5) budou představeny výsledky.

Kapitola 2

Formáty elektronických slovníků

V této kapitole budou popsány některé formáty elektronických slovníků založené na XML (Extensible Markup Framework). Podrobněji se kapitola věnuje standardu Lexical Markup Framework – LMF (viz 2.2), zmíněny jsou pak formáty Open Lexicon Interchange Format – OLIF (viz 2.3) a Extensible Dictionary Exchange Format – XDXF (viz 2.4). Příklady slovníků v jednotlivých formátech jsou uvedeny v příloze B.

2.1 Definice termínů

morfém nejmenší, mluvnicky nedělitelná část slova, která je nositelem věcného nebo mluvnického významu [13] (předpona, vpona, přípona, koncovka, kořen slova)

lexém formálně významová jednotka lexikální zásoby [13]

tvar sekvence morfémů [2]

intenze souhrn podstatných charakteristik či vlastností určité množiny entit [12]

extenze třída (množina) entit, které mají shodné vlastnosti [12]

2.2 Lexical Markup Framework

Text této kapitoly je založen na [2].

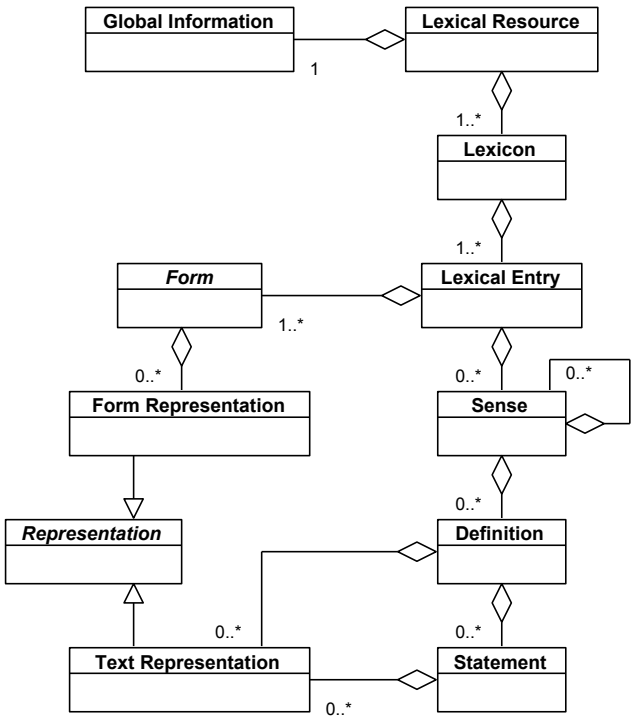
Lexical Markup Framework (LMF) je standard Mezinárodní organizace pro normalizaci (International Organization for Standardization) ISO 24613:2008 [2] pro zpracování přirozeného jazyka (*Natural Language Processing, NLP*) a strojově čitelné slovníkové zdroje (*Machine-Readable Dictionary Lexicons, MRD*) založený na XML.

Jedná se o abstraktní metamodel, který poskytuje obecný, standardizovaný systém pro konstrukci elektronických slovníků zohledňující morfologické, syntaktické a sémantické aspekty.

Lexical Markup Framework se skládá ze základního balíku (*LMF core package*) a rozšiřujících balíků (*LMF extensions*).

2.2.1 LMF core package

Základní balík (*core package*) je metamodel poskytující flexibilní základ pro vytváření LMF modelů a rozšíření. Jeho struktura je zobrazena v diagramu na obrázku 2.1. Tento balík



Obrázek 2.1: LMF Core package

poskytuje následující třídy:

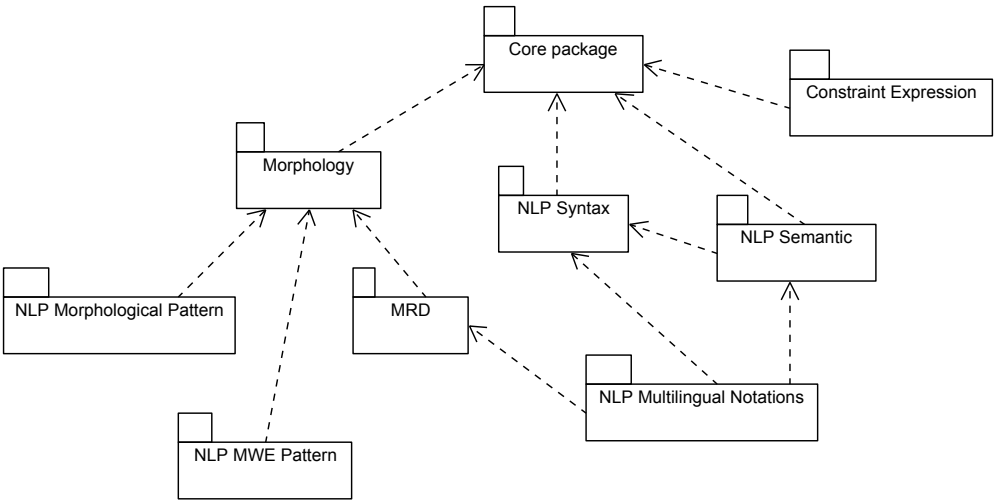
- *Lexical Resource* – reprezentuje celý lexikální zdroj, obsahuje jeden nebo více slovníků
- *Global Information* – obsahuje administrativní a obecné informace o celém lexikálním zdroji; musí obsahovat atribut *language coding*, který specifikuje standard kódování názvů jazyků; dále může obsahovat atributy *script coding* (specifikace standardu kódování názvů písem) a *character coding* (verze Unicode)
- *Lexicon* – obsahuje všechny lexikální záznamy daného jazyka
- *Lexical Entry* – reprezentuje lexém daného jazyka, spravuje vztahy mezi tvary (třída *Form*) a významy (třída *Sense*) tohoto lexému
- *Form* – abstraktní třída reprezentující tvar: lexém, morfologickou variantu lexému nebo morfém
- *Form Representation* – reprezentuje jednu pravopisnou variantu tvaru (řetězec Unicode)
- *Representation* – abstraktní třída pro Unicode řetězec a příp. atributy popisující specifický jazyk, písmo a pravopis
- *Sense* – představuje jeden význam lexému (*Lexical Entry*), umožňuje významy hierarchicky vnořovat (specifičtější významy)

- *Definition* – reprezentuje slovní popis významu čitelný pro lidi, není určen pro počítačové zpracování
- *Statement* – reprezentuje slovní popis, vylepšuje a doplňuje popis v *Definition*
- *Text Representation* – reprezentuje textový obsah *Definition* nebo *Statement*

Příklad slovníku ve formátu LMF je v příloze B.1. Tento slovník využívá rozšíření Morphology a Machine Readable Dictionary.

2.2.2 LMF extensions

Lexical Markup Framework obsahuje vedle základního balíku také několik rozšiřujících balíků (*extensions*). Jejich vztah k základnímu balíku je zobrazen v diagramu na obrázku 2.2.



Obrázek 2.2: Závisloti rozšiřujících balíků na základním balíku LMF

Rozšiřující balíky nabízejí tyto třídy[1]:

Rozšíření Morphology

Tento balík slouží k popisu morfologie lexikálních záznamů z hlediska extenze. Poskytuje dvě kategorie podtříd abstraktní třídy *Form*. První reprezentuje množiny gramatických variant, které doplňují abstraktní lexém, druhá poskytuje podtřídy pro informace související s tvary v jiném lexikálním záznamu (v jiné instanci *Lexical Entry*).

První skupinu tvoří třídy:

- *Lemma* – reprezentuje slovní tvar vybraný podle konvence k označení lexikálního záznamu (*Lexical Entry*), konvence výběru se může lišit podle jazyka, jazykové rodiny nebo může být zvolena editorem

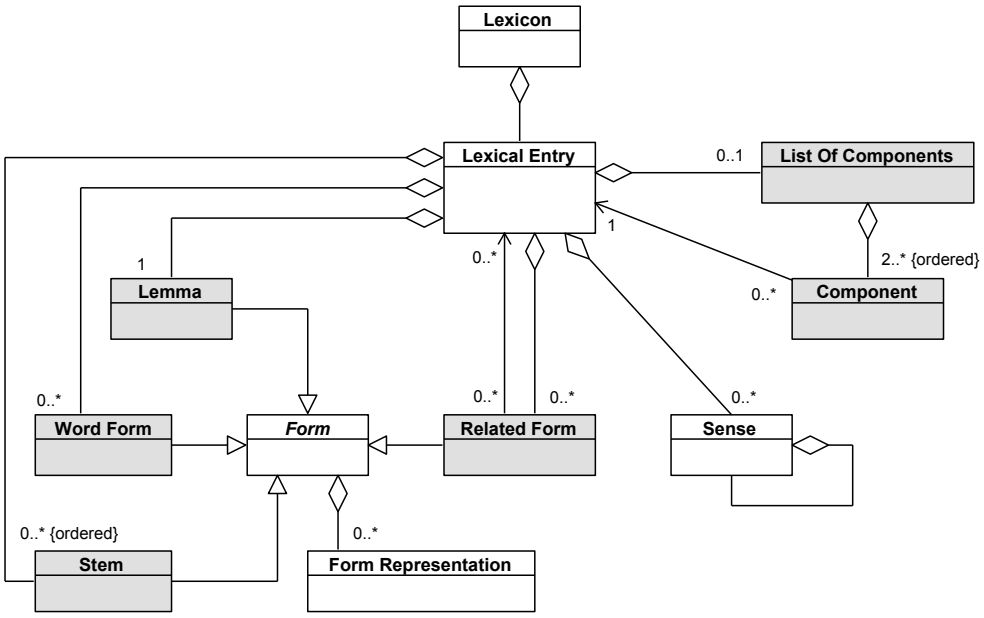
[1] Podrobněji popisuji pouze balíky použité v analyzovaných slovnících.

- *Word Form* – tvar lexému používaný ve větě nebo frázi (holé lexémy, složeniny, víceslovné výrazy)
- *Stem* – reprezentuje morfém

Druhou skupinu představuje třída *Related Form*, která reprezentuje slovní tvar nebo morfém, který může různě souviset s lexikálním záznamem (např. odvozováním, kořenem).

Toto rozšíření dále poskytuje třídy *List Of Components* a *Component*, které reprezentují celkový pohled na víceslovné výrazy.

Vzájemné vztahy mezi třídami tohoto rozšíření a s třídami základního balíku jsou znázorněny v diagramu na obrázku 2.3.



Obrázek 2.3: Rozšíření Morphology

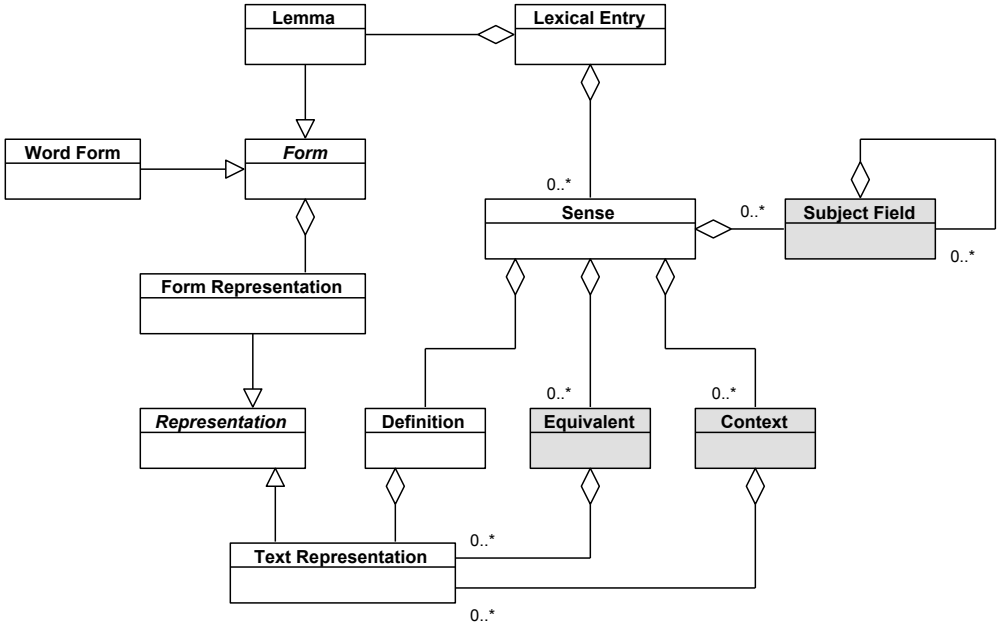
Rozšíření Machine Readable Dictionary

Rozšíření Machine Readable Dictionary poskytuje metamodel pro data uložená ve strojově čitelných slovnících. Podporuje jak možnost přístupu ke slovníku pro člověka, tak jeho strojové zpracování.

Tento balík obsahuje tři třídy:

- *Equivalent* – ve dvojjazyčných slovnících reprezentuje překlad slovního tvaru uvedeného v *Lemma*
- *Context* – reprezentuje řetězec poskytující kontext užití slovního tvaru uvedeného v *Lemma*
- *Subject Field* – reprezentuje řetězec poskytující informace o oboru

Vztahy mezi těmito třídami a třídami základního balíku a rozšiřujícího morfologického balíku jsou znázorněny v diagramu na obrázku 2.4.



Obrázek 2.4: Rozšíření Machine Readable Dictionary

Rozšíření NLP syntax

Pomocí toho balíku lze popsat vlastnosti lexému kombinovaného s jinými lexémy ve větě.

Rozšíření NLP semantics

Toto rozšíření umožňuje popsat jeden význam a jeho vztah s jinými významy v rámci jednoho jazyka. Poskytuje také spojení se syntaxí kvůli složitému vztahu syntaxe a sémantiky v mnoha jazycích.

Rozšíření NLP multilingual notation

Třídy tohoto balíku popisují reprezentaci ekvivalentů pro instance tříd *Sense* a *Syntactic Behaviour* (třída rozšíření NLP syntax) mezi dvěma a více jazyky.

Rozšíření NLP morphological patterns

Rozšíření NLP morphological patterns poskytuje třídy pro popis morfologie daného jazyka z hlediska intenze.

Rozšíření NLP multiword expression patterns

Toto rozšíření poskytuje třídy pro reprezentaci interní struktury víceslovných výrazů daného jazyka.

Rozšíření Constraint expression

Toto rozšíření umožňuje popsat omezení dvojic atribut–hodnota platná v rámci třídy *Lexicon*.

2.3 Open Lexicon Interchange Format

Open Lexicon Interchange Format (OLIF) je otevřený standard pro kódování lexikálních dat založený na formátu XML nabízející podporu pro výměnu a reprezentaci jazykových dat. Formát OLIF byl navržen a vytvořen jako oficiální standard OLIF2 Konsorciem. [3]

Slovník ve formátu OLIF se skládá ze tří částí [15]:

- *header* – obsahuje data společná pro všechny záznamy slovníku,
- *body* – obsahuje jednotlivé lexikální záznamy,
- *shared resources* – obsahuje dodatečná data (např. bibliografické informace).

Jednotlivé záznamy slovníku v části *body* jsou obaleny značkou `entry`. Data záznamu jsou rozdělena do tří hlavních skupin:

- *monolingual* (značka `mono`) – obsahuje kanonický tvar, jazyk, slovní druh, obor a sémantickou třídu,
- *cross-reference* (značka `crossRefer`) – definuje vztahy mezi daným záznamem a jinými záznamy ve slovníku stejného jazyka,
- *transfer* (značka `transfer`) – definuje vztahy mezi daným záznamem a záznamy v jiném jazyce.

Příklad jednoduchého slovníku ve formátu OLIF je v příloze B.2.

2.4 Extensible Dictionary Exchange Format

Extensible Dictionary Exchange Format (XDXF) je další formát pro uchovávání elektronických slovníků založený na XML. Vznikl pro účely převodu existujících (tištěných) slovníků do elektronické podoby pro zjednodušení úprav a vyhledávání. [6]

Struktura slovníku ve formátu XDXF je rozdělena do dvou částí:

- značka `meta_info` obaluje veškeré informace o slovníku (např. jeho název nebo autora),
- značka `lexicon` potom obsahuje jednotlivé záznamy slovníku.

Záznamy jsou uvedeny ve značce `ar` (*article*). Jeden záznam musí obsahovat alespoň jednu značku `k` (*key phrase*), která obsahuje klíčovou frázi – vlastní heslo, podle kterého lze vyhledávat abecedně ve slovníku. Značka `def` obaluje ostatní informace o záznamu. Může být vnořena a obsahovat jednotlivé definice klíčové fráze.[2] [5]

Příklad jednoduchého slovníku ve formátu XDXF je v příloze B.3.

[2] Definice formátu XDXF obsahuje celou řadu dalších značek pro podrobnější informace o záznamu. Uvedla jsem pouze ty základní, použité v příkladu slovníku v příloze B.3.

Kapitola 3

Návrh řešení

Tato kapitola se věnuje nástrojům (3.1), schématům a metrikám (3.2) použitým pro analýzu slovníků.

3.1 Použité nástroje

3.1.1 iconv

Program iconv slouží pro převod jednoho kódování textového souboru na jiné. LMF slovníky analyzované v rámci této práce jsou uloženy s kódováním UTF-8. Některé nástroje (např. knihovna libma) ovšem pracují v kódování ISO-8859-2, proto bude nutné slovníky (nebo jejich části) převést do tohoto kódování.

3.1.2 GNU Aspell

GNU Aspell je Open Source aplikace pro kontrolu pravopisu. [9] Lze ji využít při analýze tištěných slovníků převedených do elektronické podoby (formát LMF) pomocí optického rozpoznávání znaků (Optical Character Recognition, OCR). K dispozici jsou slovníky angličtiny, němčiny, francouzštiny i češtiny. Pro kontrolu češtiny bude ale použita knihovna libma (viz 3.1.4).

3.1.3 Modul SAX

SAX (Simple API for XML) je modul určený pro zpracování XML souborů. Jedná se o proudové rozhraní, aplikace tedy dostává informace z XML průběžně, bez možnosti návratu nebo navigace v XML, díky čemuž je zpracování XML souboru nenáročné na paměť. [7]

3.1.4 Knihovna libma

Knihovna libma je rozhraní k morfologickému analyzátoru češtiny. [18] Stejně jako Aspell může být tato knihovna využita pro analýzu tištěných slovníků převedených pomocí OCR. Pomocí knihovny budou analyzovány české části slovníků. Dále lze knihovnu využít pro lemmatizaci, tedy převedení slov do jejich základního tvaru pro snazší překlad.

3.1.5 Knihovna gensim

Gensim je knihovna umožňující sémantickou analýzu textů. Obsahuje algoritmy a schémata pro převod dokumentů do vektorového prostoru a jejich následné zpracování – Tf (*term frequency*), Tf-Idf (*term frequency – inverse document frequency*), LSA (*latent semantic analysis*) a další. [19]

3.1.6 Knihovna NLTK

Knihovna NLTK (*Natural Language Toolkit*) je rozsáhlá knihovna určená pro zpracování přirozeného jazyka. [8] Z této knihovny bude využit nástroj pro lemmatizaci pracující s databází WordNet. Bude použit pro lemmatizaci anglických slov pro přesnější sémantickou analýzu.

3.2 Schémata a metriky

3.2.1 Vector Space Model

Model vektorového prostoru (Vector Space Model) je algebraický model pro reprezentaci textových dokumentů jako vektorů. [4] Dokumenty a dotazy jsou reprezentovány vektory:

$$\begin{aligned} d_j &= (w_{1j}, w_{2j}, \cdots, w_{tj}) \\ q &= (w_{1q}, w_{2q}, \cdots, w_{tq}) \end{aligned}$$

Každá dimenze vektoru odpovídá jednomu termu – pokud se term vyskytuje v dokumentu (příp. dotazu), pak je daná hodnota (jeho váha) nenulová. Existuje několik schémat pro stanovení těchto vah, jedním z nejlepších je Tf-Idf. Jako termy jsou obvykle označována jednotlivá slova, klíčová slova nebo slovní spojení. V případě, kdy jsou za termy zvolena jednotlivá slova, je počet dimenzí t roven počtu různých slov vyskytujících se v souboru dokumentů. [4]

Pro stanovení podobnosti dvou takových vektorů lze použít např. **kosinovou podobnost**. Ta je podle [16] definována takto:

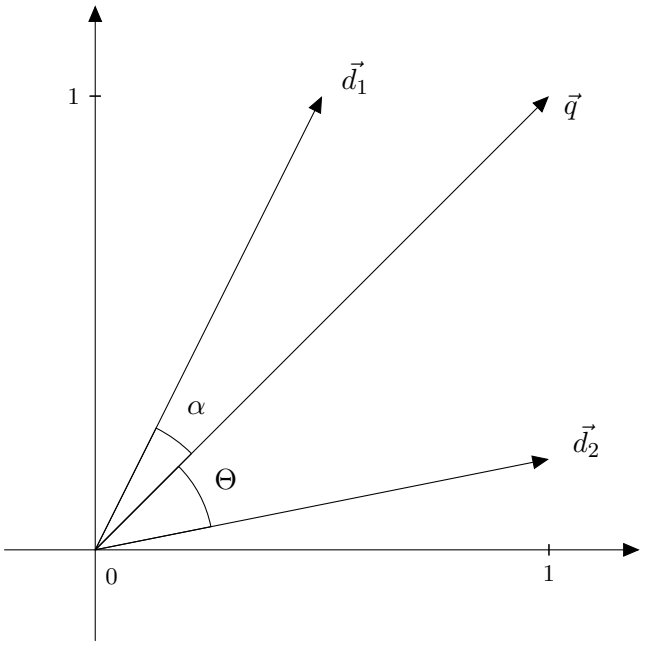
$$similarity(\vec{x}, \vec{y}) = \cos(\theta) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| * \|\vec{y}\|}$$

Stanovení kosinové podobnosti se rovná nalezení kosinu úhlu mezi dvěma vektory. Pokud je výsledek kosinové podobnosti roven 0, pak dokumenty nesdílejí žádné atributy, naopak kosinus úhlu roven 1 znamená shodnost obou dokumentů.

Podobnost dvou dokumentů d_1 a d_2 s dotazem q ilustruje obrázek 3.1. Vektor dokumentu $\vec{d_1}$ svírá s vektorem dotazu $\vec{q}$ menší úhel než vektor dokumentu $\vec{d_2}$. Kosinus úhlu α bude tedy větší než kosinus úhlu Θ, což znamená, že dotaz je více podobný dokumentu d_1 než dokumentu d_2.

3.2.2 Latentní sémantická analýza

Latentní sémantická analýza (*Latent semantic analysis, LSA*) je metoda výpočtu podobnosti dvou dokumentů.



Obrázek 3.1: Kosinová podobnost

Tato metoda řeší problémy modelu vektorového prostoru, kterými jsou synonymnost (více různých slov má stejný význam), kdy je výsledná podobnost podhodnocena, a mnohoznačnost (jedno slovo má více významů), v jejímž případě je výsledná podobnost nadhodnocena. [14]

Latentní sémantická analýza dále využívá dekompozice matice term-dokument pomocí *singular value decomposition* (SVD) ke snížení počtu dimenzí vektorového prostoru.

Term frequency

Pro sémantické analýzy slovníků je potřeba sestavit tzv. vektory dokumentů, kde dokumenty budou texty vysvětlující význam daného slova či doplňující kontext jeho užití. Tyto dokumenty mohou vypadat např. takto (příklad převzat z [11]):

1. *Shipment of gold damaged in a fire.*
2. *Delivery of silver arrived in a silver truck.*
3. *Shipment of gold arrived in a truck.*

Nejjednodušším způsobem, jak lze vytvořit vektory těchto dokumentů, je spočítat četnost jednotlivých slov v dokumentech (*term frequency*) značenou jako $tf_{t,d}$, kde t, d v indexu značí slovo (term) a dokument:

$$
\begin{array}{l}
\text{a} \\
\text{arrived} \\
\text{damaged} \\
\text{delivery} \\
\text{fire} \\
\text{gold} \\
\text{in} \\
\text{of} \\
\text{shipment} \\
\text{silver} \\
\text{truck}
\end{array}
\quad \vec{d_1} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}
\quad \vec{d_2} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 2 \\ 1 \end{bmatrix}
\quad \vec{d_3} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}
$$

Term frequency – Inverse document frequency

Ohodnocení podle schématu *term frequency* (Tf) ovšem předpokládá, že slova v dokumentu jsou všechna stejně důležitá. Ve většině případů je tohle ale nežádoucí. Pro relevantnější ohodnocení lze podle [14] použít tzv. *četnost dokumentů* (*document frequency*) označovanou jako df_t. Ta je definována jako počet dokumentů v kolekci, které obsahují slovo t. Příklad je uveden v tabulce 3.1.

t	df_t	idf_t
a	3	0
arrived	2	0,1761
damaged	1	0,4771
delivery	1	0,4771
fire	1	0,4771
gold	2	0,1761
in	3	0
of	3	0
shipment	2	0,1761
silver	1	0,4771
truck	2	0,1761

Tabulka 3.1: Četnost dokumentů df_t a inverzní četnost dokumentů idf_t

Četnost dokumentů lze převést na váhu označovanou jako *inverzní četnost dokumentů* (*inverse document frequency*) podle vzorce [14]

$$idf_t = \log \frac{N}{df_t},$$

kde N je celkový počet dokumentů v kolekci.

Hodnoty idf_t pro příklad jsou opět uvedeny v tabulce 3.1. Pomocí četnosti slov a inverzní četnosti dokumentů lze sestavit konečnou váhu každého slova v každém dokumentu nazývanou *Term frequency – Inverse document frequency* (Tf-Idf):

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t$$

Pak vektory dokumentů z příkladu budou:

$$
\begin{array}{l|ccc}
\text{a} & 0 & 0 & 0 \\
\text{arrived} & 0 & 0{,}1761 & 0{,}1761 \\
\text{damaged} & 0{,}4771 & 0 & 0 \\
\text{delivery} & 0 & 0{,}4771 & 0 \\
\text{fire} & 0{,}4771 & 0 & 0 \\
\text{gold} & 0{,}1761 & 0 & 0{,}1761 \\
\text{in} & 0 & 0 & 0 \\
\text{of} & 0 & 0 & 0 \\
\text{shipment} & 0{,}1761 & 0 & 0{,}1761 \\
\text{silver} & 0 & 0{,}9542 & 0 \\
\text{truck} & 0 & 0{,}1761 & 0{,}1761
\end{array}
$$

$\vec{d_1}$ $\vec{d_2}$ $\vec{d_3}$

Podle [14] je pak každému slovu přiřazena váha v dokumentu takto:

1. Nejvyšší váhu mají slova, která se často vyskytují v malém počtu dokumentů.
2. Nižší váhu mají slova, která se vyskytují méně často, nebo se vyskytují v mnoha dokumentech.
3. Nejnižší váhu mají slova vyskytující se prakticky ve všech dokumentech.

Kosinová podobnost

K příkladu je přidán dotaz (*query*) do kolekce dokumentů: *gold silver truck* (příklad převzat z [11]). Vektor tohoto dotazu sestavený pomocí váhového schématu Tf-Idf z předchozí kapitoly bude:

$$
\vec{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0{,}1761 \\ 0 \\ 0 \\ 0 \\ 0{,}4771 \\ 0{,}1761 \end{bmatrix}
$$

Podobnosti jednotlivých dokumentů ze začátku kapitoly s dotazem q jsou uvedeny v tabulce 3.2.

dokument	**kosinová podobnost**
d_1	0,0801
d_2	0,8247
d_3	0,3272

Tabulka 3.2: Kosinová podobnost dokumentů d_1, d_2 a d_3 s dotazem q

Z tabulky vyplývá, že nejpodobnější dotazu je dokument 2 (*Delivery of silver arrived in a silver truck.*).

Latentní sémantická analýza

Pro analýzu pomocí latentní sémantické analýzy je třeba sestavit tzv. *term-dokument* matici, kde každý řádek reprezentuje jednotlivá slova a každý sloupec jednotlivé dokumenty v kolekci. Pro příklad bude matice term-dokument při použití Tf-Idf pro určení vah slov v dokumentech vypadat následovně:

$$C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0{,}1761 & 0{,}1761 \\ 0{,}4771 & 0 & 0 \\ 0 & 0{,}4771 & 0 \\ 0{,}4771 & 0 & 0 \\ 0{,}1761 & 0 & 0{,}1761 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0{,}1761 & 0 & 0{,}1761 \\ 0 & 0{,}9542 & 0 \\ 0 & 0{,}1761 & 0{,}1761 \end{bmatrix}$$

Singular Value Decomposition

Singulární rozklad (*singular value decomposition*) je [10]:

1. metoda pro transformaci korelačních proměnných do množiny nekorelačních proměnných, která lépe ukazuje různé vztahy mezi originálními daty
2. metoda pro identifikaci a uspořádání dimenzí, při kterém data vykazují největší rozdíly
3. metoda pro nalezení nejlepší aproximace originálních dat při použití méně dimenzí

Singulární rozklad je podle [10] definován:

Definice 1 (Singulární rozklad)

$$C = USV^T,$$

kde

- $U^TU = I$, $V^TV = I$,
- *sloupce matice* U *jsou ortonormální vlastní vektory matice* CC^T,
- *sloupce matice* V *jsou ortonormální vlastní vektory matice* C^TC,
- S *je diagonální matice obsahující druhé odmocniny vlastních čísel matic* CC^T *a* C^TC *v sestupném pořadí.*

Následující příklad je převzat z [10]:
Pro matici

$$C = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

je

$$CC^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

Pro vlastní hodnoty λ matice CC^T platí rovnice $|CC^T - \lambda I| = 0$:

$$CC^T - \lambda I = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{bmatrix}$$

$$\begin{vmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{vmatrix} = 0$$

Vyřešením rovnice získáme $\lambda_1 = 12$ a $\lambda_2 = 10$.
Pro vlastní vektory platí rovnice $CC^T\vec{x} = \lambda\vec{x}$. Z toho:

$$\begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Přepsáno jako soustava rovnic:

$$11x_1 + x_2 = \lambda x_1 \tag{3.1}$$
$$x_1 + 11x_2 = \lambda x_2 \tag{3.2}$$

Dosazením λ_1 do rovnice 3.1 získáme rovnici:

$$x_1 = x_2,$$

z níž lze stanovit vlastní vektor $[1;1]$. Obdobně pro λ_2 se spočítá vlastní vektor $[1;-1]$. Sestavením těchto vektorů jako sloupců matice (podle hodnot odpovídajících vlastních čísel sestupně) získáváme matici

$$X = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Na závěr je třeba matici převést na ortogonální pomocí Gram-Schmidt ortonormalizačního procesu:

$$\vec{u_1} = \frac{\vec{x_1}}{|\vec{x_1}|} = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$$

$$\vec{w_2} = \vec{x_2} - \vec{u_1} \cdot \vec{x_2} * \vec{u_1} = [1, -1]$$

$$\vec{u_2} = \frac{\vec{w_2}}{|\vec{w_2}|} = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right]$$

Matice U je pak rovna:

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Podobným způsobem se spočítá i matice V:

$$V = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{30}} \\ \frac{1}{\sqrt{6}} & 0 & -\frac{5}{\sqrt{30}} \end{bmatrix}$$

Pro singulární rozklad je potřeba matici V transponovat:

$$V^T = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & -\frac{5}{\sqrt{30}} \end{bmatrix}$$

$M \times N$ matice S je diagonální matice, jejímiž prvky jsou druhé odmocniny vlastních čísel matic CC^T a C^TC uspořádané sestupně podle velikosti:

$$S = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}$$

Singulární rozklad matice C je pak:

$$C = USV^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & -\frac{5}{\sqrt{30}} \end{bmatrix}$$

Matice term-dokument vytvořená k příkladu v této kapitole má následující singulární rozklad (rozklad byl vypočten za pomoci on-line kalkulátoru BlueBit Matrix Calculator [1]):

$$U = \begin{bmatrix} 0 & 0 & 0 \\ -0{,}1695 & 0{,}0331 & -0{,}4919 \\ -0{,}0023 & 0{,}6499 & 0{,}2158 \\ -0{,}4341 & -0{,}0091 & 0{,}0805 \\ -0{,}0023 & 0{,}6499 & 0{,}2158 \\ -0{,}0101 & 0{,}2763 & -0{,}4419 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -0{,}0101 & 0{,}2763 & -0{,}4419 \\ -0{,}8683 & -0{,}0182 & 0{,}1609 \\ -0{,}1695 & 0{,}0331 & -0{,}4919 \end{bmatrix}$$

$$S = \begin{bmatrix} 1{,}0971 & 0 & 0 \\ 0 & 0{,}7257 & 0 \\ 0 & 0 & 0{,}3332 \end{bmatrix} \quad V^T = \begin{bmatrix} -0{,}0052 & -0{,}9983 & -0{,}0576 \\ 0{,}9886 & -0{,}0138 & 0{,}1501 \\ 0{,}1507 & 0{,}0562 & -0{,}9870 \end{bmatrix}$$

Výpočet podobnosti

Po rozkladu matice term-dokument je možné snížit počet dimenzí vektorového prostoru, jejichž počet se v praxi pohybuje v desítkách tisíc. Výsledkem jsou řádově stovky dimenzí. [14] Matice U, S a V^T se upraví na k dimenzí následovně:

- V matici U zůstane prvních k sloupců.

- V matici S zůstane prvních k sloupců a prvních k řádků.
- V matici V^T zůstane prvních k řádků.

Takto upravené matice jsou označeny U_k, S_k a V_k^T.
Dále je třeba upravit vektor dotazu podle rovnice [11]:

$$\vec{q_k} = \vec{q^T} U_k S_k^{-1}$$

Pro příklad ze začátku této kapitoly a $k = 2$ jsou matice a vektor dotazu následující:

$$U_k = \begin{bmatrix} 0 & 0 \\ -0{,}1695 & 0{,}0331 \\ -0{,}0023 & 0{,}6499 \\ -0{,}4341 & -0{,}0091 \\ -0{,}0023 & 0{,}6499 \\ -0{,}0101 & 0{,}2763 \\ 0 & 0 \\ 0 & 0 \\ -0{,}0101 & 0{,}2763 \\ -0{,}8683 & -0{,}0182 \\ -0{,}1695 & 0{,}0331 \end{bmatrix} \quad S_k = \begin{bmatrix} 1{,}0971 & 0 \\ 0 & 0{,}7257 \end{bmatrix}$$

$$V_k^T = \begin{bmatrix} -0{,}0052 & -0{,}9983 & -0{,}0576 \\ 0{,}9886 & -0{,}0138 & 0{,}1501 \end{bmatrix} \quad \vec{q_k} = \begin{bmatrix} -0{,}4065 & 0{,}0631 \end{bmatrix}$$

Sloupce matice V_k^T jsou souřadnice dokumentů v novém k-dimenzionálním vektorovém prostoru.

Podobnost dotazu s dokumenty se nyní spočítá např. pomocí kosinové podobnosti. Výsledné hodnoty pro příklad (pro srovnání spolu s hodnotami spočítanými bez převodu pomocí LSA) jsou uvedeny v tabulce 3.3.

	kosinová podobnost	
dokument	**bez LSA**	**s LSA**
d_1	0,0801	0,1586
d_2	0,8247	0,9860
d_3	0,3272	0,4972

Tabulka 3.3: Podobnost dotazu q s dokumenty d_1, d_2 a d_3

Kapitola 4

Implementace

Tato kapitola se věnuje samotné implementaci skriptů pro analýzu slovníků.

4.1 Implementační jazyk

Pro implementaci skriptů byl zvolen jazyk Python (verze 2.7.2 nainstalovaná na serveru `merlin` a verze 2.7.3 na serveru `knot01`), protože poskytuje velké množství knihoven a modulů pro zpracování přirozeného jazyka. Všechny výstupní soubory skriptů jsou kódovány v ISO-8859-2. Všechny skripty vyžadují rozhraní pylibma a pylibma_ext knihovny libma.

4.2 Zpracování XML souboru

Všechny analyzované slovníky byly uloženy v XML formátu Lexical Markup Framework (LMF). Vzhledem k velikosti souborů, která se pohybuje od 30 do 150 MB, byl pro zpracování XML souborů zvolen modul SAX (The Simple API for XML, viz kapitolu 3.1.3), který nenačítá celý XML soubor do paměti, ale analyzuje jej postupně a je tak nenáročný na paměť.

Pro obsluhu analyzátoru SAX je třeba vytvořit třídu rozhraní `ContentHandler` implementující metody `startElement(name, attrs)` a `endElement(name)`, kde atribut `name` je jméno elementu a atribut `attrs` představuje atributy daného elementu. Tyto metody určují události provedené podle jména aktuálního elementu. Většinou se jedná o uložení informací z atributů do interní struktury, u koncových elementů `LexicalEntry` a `Lexicon` jsou volány metody implementující samotné analýzy těchto informací.

Pro uložení informací ze slovníku jsou vytvořeny třídy odpovídající standardu LMF a struktuře analyzovaného slovníku.

4.3 Analýza českých částí slovníků

Pro analýzu českých částí slovníků byl vytvořen skript `dbxml3_cz.py`, který pomocí knihovny libma analyzuje slova v českých řetězcích. Analyzované slovníky byly uloženy v kódování UTF-8, pro analýzu knihovnou libma je bylo třeba převést do kódování ISO-8859-2. Výstupem skriptu je soubor se záznamy ve formátu:

```
<neznámé slovo><tabulátor><XML značka><tabulátor><kontext><znak nového řádku>
```

XML značkou je míněna nejbližší značka, která obsahuje dané slovo, kontextem pak řetězec, v němž se neznámé slovo ve slovníku vyskytuje. XML značky mohou být:

```
Lemma
WordForm
Sense
Equivalent
TextRepresentation
Definition
```

Skript lze spustit na serverech `merlin` a `knot01`.

4.4 Analýza německo-českého slovníku

Německo-český slovník byl převeden pomocí OCR do elektronické podoby a v rámci projektu `ocr2lmf` převeden do formátu Lexical Markup Framework. Během těchto převodů vznikly ve slovníku chyby způsobující nevalidní XML soubor, který nelze zpracovat pomocí modulů určených pro zpracování XML. Jednalo se o uvozovky a apostrofy uvnitř řetězců.

4.4.1 Oprava chyb

Pro opravu chyb ve slovníku byl vytvořen skript `ncs-lmf_correct.py`, jehož výstupem je soubor `ncs-lmf-corrected.xml`, ve kterém jsou nahrazeny uvozovky a apostrofy odpovídajícími XML entitami (`&quot;` a `&apos;`). Oprava byla implementována za použití regulárních výrazů v několika průchodech.

4.4.2 Analýza

Pro analýzu opraveného slovníku byl vytvořen skript `dbxml3_ncs.py`, který vyextrahuje všechny textové řetězce, převede je pomocí iconv (3.1.1) z kódování UTF-8 do kódování ISO-8859-2 a rozdělí je na jednotlivá slova. Vzhledem k tomu, že v řetězcích se vyskytují společně jak německá, tak česká slova, není možné je rozdělit do dvou skupin a tyto analyzovat samostatně. Slova jsou nejprve všechna analyzována pomocí GNU Aspell (3.1.2) jako německá, nenalezená slova potom dále pomocí libma (3.1.4) jako slova česká. Slova, která projdou i druhým sítem v podobě analýzy knihovnou libma, jsou uložena do souboru `ncs_unknown` v tomto formátu:

```
Lexical Entry id: <id>
<seznam nenalezených slov dané lexical entry oddělených znakem nového řádku>
<prázdný řádek>
```

Oba skripty lze spustit na serveru `merlin`.

4.5 Sémantická analýza česko-anglického slovníku

Pro sémantickou analýzu česko-anglického slovníku byl vytvořen skript `dbxml3_lsa.py`. Tento skript porovnává pomocí metrik popsaných v kapitole 3 řetězce představující anglické ekvivalenty českých hesel a komentáře k nim s příklady uvedenými v části kontextu daného významu hesla.

Na příkladu je možné vidět tyto informace zapsané v LMF:

```
<LexicalEntry id="e31918g">
    <Lemma>
        <feat att="writtenForm" val="koruna"/>
    </Lemma>
    <Sense>
        <Equivalent>
            <feat att="language" val="eng"/>
            <feat att="gloss" val="čelenka"/>
            <feat att="writtenForm" val="diadem"/>
        </Equivalent>
        <Equivalent>
            <feat att="language" val="eng"/>
            <feat att="gloss" val="královská ap."/>
            <feat att="writtenForm" val="crown"/>
        </Equivalent>
        <Context>
            <TextRepresentation>
                <feat att="languageIdentifier" val="ces"/>
                <feat att="text" val="královská koruna"/>
            </TextRepresentation>
            <TextRepresentation>
                <feat att="languageIdentifier" val="eng"/>
                <feat att="text" val="royal crown"/>
            </TextRepresentation>
        </Context>
    </Sense>
</LexicalEntry>
```

Anglické ekvivalenty jsou uvedeny ve značce `Equivalent` v části `writtenForm`, české komentáře k těmto ekvivalentům se zapisují do části `gloss`. Příklady užití nebo širší kontext užití ekvivalentu je uveden ve značce `Context` pomocí třídy `TextRepresentation` v části `text`. Tyto jsou uvedeny v českém i anglickém jazyce, pro analýzu jsou vybírány pouze anglické varianty.

Všechny řetězce vyextrahované z XML jsou rozděleny na jednotlivá slova.

Slova anglických ekvivalentů jsou převedena na základní tvar pomocí lemmatizátoru knihovny NLTK.

České komentáře bylo třeba pro další analýzu přeložit do angličtiny. Pro jejich překlad byl využit slovník uložený v souboru `slovnik_data.txt` [17], který je kvůli rychlejšímu vyhledávání nahrán do paměti do datového struktury *slovník*. Komentáře jsou pak přeloženy slovo od slova do všech variant, které použitý slovník nabízí. Vytvoření těchto variant bylo implementováno pomocí modulu `itertools` metodou `itertools.product()`, která vytvoří kartézský součin seznamů překladů jednotlivých slov.

České komentáře přeložené do angličtiny jsou spojeny s odpovídajícími anglickými ekvivalenty významu do jednotlivých seznamů, které představují dotazy. Kolekce dokumentů je pak tvořena anglickými texty uvedenými ve značce `Context` daného významu (`Sense`).

Takto připravené kolekce dokumentů a dotazů jsou zpracovány pomocí knihovny gensim (viz kapitolu 3.1.5) následovně:

1. Metodou `doc2bow` je vytvořena matice term-dokument (jako seznam seznamů `corpus`)

– jako váhy jsou použity četnosti slov v dokumentech Tf.

2. Tato matice je pak převedena na matici `corpus_tfidf` s váhami podle Tf-Idf metodou `models.TfidfModel()`.

3. Následně jsou matice `corpus` a `corpus_tfidf` zpracovány metodou `models.LsiModel()`[1], která implementuje latentní sémantickou analýzu. Výsledkem jsou matice `corpus_lsi` a `corpus_tf_lsi`.

4. Metodou `similarities.MatrixSimilarity()` je provedena inicializace podobnostních dotazů pro všechny čtyři matice.

5. Každý vytvořený dotaz je pak převeden podle bodů 1 až 3 do vektorových prostorů Tf, Tf-Idf a LSA a je proveden výpočet kosinové podobnosti mezi dotazem a všemi dokumenty v kolekci.

Ze všech podobností spočítaných v rámci jedné kolekce dokumentů a odpovídající kolekce dotazů je vybrána ta nejvyšší. Pokud je tato podobnost menší, než jakou udává stanovený limit (lze nastavit pomocí parametru skriptu, implicitně je 0,5), pak je do výstupního souboru zapsán záznam ve formátu:

```
<LexicalEntryID> <SenseNumber><nový řádek>
<seznam dotazů oddělených znakem nového řádku>
<seznam dokumentů s podlimitní podobností oddělených znakem nového řádku>
<prázdný řádek>
```

`LexicalEntryID` je identifikátor záznamu, `SenseNumber` identifikátor významu daného hesla (záznamu).

Jednotlivé dotazy jsou uvozeny znaky `Q:`, jednotlivé dokumenty znaky `D:`. Za textem dokumentu je mezerou oddělená nejvyšší spočítaná podobnost.

Zvolila jsem čtyři způsoby vyhodnocení podobnosti:

1. Schéma Tf-Idf a kosinová podobnost.

2. Schéma Tf-Idf, latentní sémantická analýza a kosinová podobnost.

3. Schéma Tf, latentní sémantická analýza a kosinová podobnost.

4. Schéma Tf a kosinová podobnost.

Skript lze spustit na serveru `knot01`.

[1] Latentní sémantická analýza (Latent Semantic Analysis, LSA) je v některých zdrojích označovaná jako latentní sémantická indexace (Latent Semantic Indexing, LSI)

Kapitola 5

Výsledky

5.1 Analýza české části slovníků

Analyzované slovníky: slovníky obsahující češtinu z adresáře `dicts2lmf/dictform01/lmf_dicts/` a všechny slovníky z adresáře `dicts2lmf/dictform02/lmf_dicts/`

České části slovníků byly analyzovány za účelem získání slov, která nejsou ve slovnících knihovny libma. Tabulky 5.1 a 5.2 uvádějí procenta neznámých českých slov ve výše uvedených slovnících.

Slovník	Neznámá slova
czen-lmf.xml	4,25 %
czfr-lmf.xml	1,94 %
czge-lmf.xml	3,46 %
czru-lmf.xml	1,88 %
czsp-lmf.xml	1,33 %
encz-lmf.xml	2,96 %
frcz-lmf.xml	2,14 %
gecz-lmf.xml	1,54 %
rucz-lmf.xml	1,01 %
spcz-lmf.xml	2,02 %

Tabulka 5.1: Výsledky analýzy české části slovníků pomocí `libma` – `dicts2lmf/dictform01/lmf_dicts/`

Slovník	Neznámá slova
encz-lmf.xml	3,55 %
frcz-lmf.xml	2,24 %
gecz-lmf.xml	1,16 %
spcz-lmf.xml	6,11 %

Tabulka 5.2: Výsledky analýzy české části slovníků pomocí `libma` – `dicts2lmf/dictform02/lmf_dicts/`

Záznamy ve výstupních souborech (viz příloha A) vypadají např. takto[1]

```
třezalkovitých  Sense   pojmenování různých druhů rostlin z čeledi
    třezalkovitých a lomikamenovitých
koroně  Equivalent  v sluneční koroně při úplném zatmění slunce
zdrojembýt  Equivalent  zdrojembýt hlavním dodavatelem
```

Poslední příklad ukazuje, že ve slovnících se vyskytují také překlepy, v tomto případě v podobě vynechané mezery.

5.2 Analýza německo-českého slovníku

Analyzovaný slovník: `ocr2lmf/dictform02/lmf_dicts/ncs-lmf.xml`

Nalezeno nebylo celkem 79 602 slov, což je asi 5,6 % všech analyzovaných slov. Nejčastější chybou u českých slov bylo chybné rozpoznání znaku „á“ jako znaku „ä“, znaku „í“ jako znaku „f“, případně zcela chybějící interpunkce. V německé části se vykytovaly podobné chyby (např. mezi znaky „i“ a „l“), další chyby byly způsobeny malým počátečním písmenem u podstatných jmen nebo chybějícím písmenem „e“ na konci slova. Tyto chyby jsou viditelné v následujícím textu vyňatém z výstupního souboru skriptu.

```
Lexical Entry id: S 331
fass

Lexical Entry id: S 332
Sag

Lexical Entry id: I 29
necf
zäjmy
rozdilne
nekdo
mä
rozdilne
zäjmy
```

5.3 Sémantická analýza česko-anglického slovníku

Analyzovaný slovník: `dicts2lmf/dictform01/lmf_dicts/czen-lmf.xml`

Slovník byl analyzován za účelem nalezení kontextů nesprávně přiřazených k významu slova.

Ze čtyř implementovaných způsobů stanovování podobnosti (viz 4.5) se jako nejlepší ukázal způsob předposlední, tedy vynechání převodu do prostoru Tf-Idf). Pro velmi krátké dokumenty (dokumenty o délce jednoho až dvou slov) byl přidán způsob poslední, tj. spočítání kosinové podobnosti vektorů s váhami v podobě četnosti slov v dokumentu.

[1] Jedná se o několik řádků z výsledků analýzy slovníku `dicts2lmf/dictform02/lmf_dicts/encz-lmf.xml`.

Limit podobnosti jsem nejprve nastavila na hodnotu 0,5. Po analýze výstupu skriptu s takto nastaveným limitem podobnosti jsem zjistila, že za podobné (a tedy správně přiřazené) mohu považovat všechny dokumenty, které mají s alespoň jedním dotazem alespoň jedním způsobem stanovenou podobnost větší než nula (hranici jsem následně nastavila na 0,01).

Přestože je kombinace výše jmenovaných způsobů výpočtu podobnosti poměrně úspěšný, stále se ve výsledcích objevují podobné dokumenty s nulovou podobností. Důvodem je především skutečnost, že kolekce dokumentů jsou velmi malé (některé obsahují pouze jeden nebo dva dokumenty).

Výsledky mají tuto podobu:

```
e5g 1
Q: ah (potěšení, bolest)
Q: aah
D: Oh, that one. 0.0068229

e117g 3
Q: absence (neexistence)
Q: absence of sth (neexistence)
D: tone deafness 0.0

e129g 1
Q: absolutism
Q: despotism (krutý)
Q: absolute rule
Q: arbitrary rule (svévolný)
D: absolutist 0.0
```

Ve výstupním souboru jsou uvedeny vždy jen ty dokumenty, které mají největší podobnost s odpovídajícími dotazy menší, než je stanovená hranice (v tomto případě byla hranice podobnosti stanovena na 0,01). Dokumentů mohlo být v dané kolekci více, což je případ prvního příkladu, kde bylo v kolekci šest následujících dokumentů:

- *Aah, that's better!*
- *Hey! That's a good idea!*
- *Ah, that feels good!*
- *Ah, I see!*
- *Oh, that one.*
- *Say aah!*

Podobnost pátého dokumentu byla tak nízká vzhledem k tomu, že dokument neobsahuje žádné ze slov dotazů. Nebyla ale ani nulová, protože dokument obsahuje slovo *that*, které se vyskytuje i v ostatních dokumentech kolekce.

Ve druhém případě se jedná o významově podobné texty (*tone deafness* znamená *absence hudebního sluchu*). Dokument však neobsahuje žádné ze slov dotazů a kolekce dokumentů obsahovala pouze tento jeden dokument.

Poslední příklad ilustruje případ rozdílných tvarů slov a zároveň malého počtu dokumentů.

Kapitola 6

Závěr

Cílem této práce bylo analyzovat slovníky ve formátu Lexical Markup Framework za účelem odhalení chyb vzniklých převodem do tohoto formátu.

Byl vytvořen skript pro opravu nevalidního XML souboru se slovníkem převedeným z tištěné podoby a skript pro zjištění úspěšnosti převedení tištěného textu do elektronické podoby. Analýzou bylo zjištěno, že pouhých 5,6 % slov nebylo správně rozpoznáno.

Hlavní částí práce byl sémantický analyzátor česko-anglického slovníku. Tento měl odhalit nesprávně přiřazené příklady užití (či kontext užití) slova k jeho významu. Výsledky jsou vzhledem k malému množství textů dostupných pro porovnávání málo přesné. Přesto se na základě výstupu skriptu pro tuto analýzu domnívám, že takové chyby se v analyzovaném slovníku nenacházejí. Některé dvojice porovnávaných textů sice analyzátor vyhodnotil jako nepodobné, nicméně jedná se buď o případ, kdy jsou ve dvou textech podobná slova (např. přídavné jméno a sloveso se stejným základem), nebo o případ, kdy jsou dva texty syntakticky skutečně zcela rozdílné, nicméně významově jsou si blízké, ale vzhledem k malému množství dat nebylo možné tyto sémantické podobnosti odhalit.

Dále byl vytvořen skript pro získání českých slov ze slovníků, která nejsou obsažena ve slovnících knihovny libma. Na základě výstupů tohoto skriptu bude možné slovníky této knihovny o neznámá slova rozšířit. Množství neznámých slov v LMF slovnících přesáhlo jen u čtyř ze 14 slovníků 3 % všech analyzovaných slov, přičemž poměrově nejvíce jich bylo ve španělsko-českém slovníku (6,11 %).

Literatura

[1] BlueBit Matrix Calculator. [online], 2003 [cit. 25. 4. 2013].
URL http://www.bluebit.gr/matrix-calculator/

[2] ISO 24613:2008 Language Resource Management – Lexical Markup Framework (LMF). 2008.

[3] The open XML language data standard. [online], 2008 [cit. 25. 4. 2013].
URL http://www.olif.net/index.htm

[4] Vector Space Model. [online], 2013 [cit. 6. 5. 2013].
URL http://en.wikipedia.org/wiki/Vector_space_model

[5] Description. [online], [cit. 11. 5. 2013].
URL https://github.com/soshial/xdxf_makedict/blob/master/format_standard/xdxf_description.md

[6] XDXF Manual. [online], [cit. 25. 4. 2013].
URL http://web.archive.org/web/20120206021908/http://xdxf.revdanica.com/drafts/logical/05a/XDXF_manual.html#XMLEditors

[7] Simple API for XML. [online], [cit. 26. 4. 2013].
URL http://www.megginson.com/downloads/SAX/

[8] Natural Language Toolkit. [online], [cit. 9. 5. 2013].
URL http://nltk.org/

[9] ATKINSON, K.: GNU Aspell. [online], 2004 [cit. 26. 4. 2013].
URL http://aspell.net/

[10] BAKER, K.: *Singular Value Decomposition Tutorial*. Březen 2005.
URL http://www.ling.ohio-state.edu/~kbaker/pubs/Singular_Value_Decomposition_Tutorial.pdf

[11] GARCIA, E.: *SVD and LSI Tutorial4: Latent Semantic Indexing (LSI) How-to Calculations*.
URL http://www.na.iac.cnr.it/~bdv/SVD%20and%20LSI%20Tutorial%204.pdf

[12] JUHAŇÁK, L.: Lingvistické a sémiotické pojmy a problémy související se selekčními jazyky. [online], Říjen 2008 [cit. 24. 4. 2013].
URL http://www.inflow.cz/lingvisticke-semioticke-pojmy-problemy-souvisejici-se-sj

[13] Kolektiv autorů: *Slovník spisovné češtiny pro školu a veřejnost*. Praha: Academia, druhé vydání, 2001.

[14] MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H.: *An Introduction to Information Retrieval*. Cambridge, Anglie: Cambridge University Press, Duben 2009, 544 s.
URL http://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf

[15] McCORMICK, S.: The Structure and Content of the Body of an OLIF v.2.0/2.1 File. [online], [cit. 11. 5. 2013].
URL http://www.olif.net/documents/NewOLIFstruct&content.pdf

[16] PHILIP, S.: Discussion of Similarity Metrics: Cosine Similarity. [online], [cit. 26. 4. 2013].
URL http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/sphilip/cos.html

[17] SVOBODA, M.: GNU/FDL Anglicko-Český slovník. [online], 2004 [cit. 7. 5. 2013].
URL http://slovnik.zcu.cz/download.php

[18] ČERNÝ, S.: Dokumentace knihovny libma. Technická zpráva, Fakulta informačních technologií, Vysoké učení technické v Brně, 2010.

[19] ŘEHŮŘEK, R.: Introduction. [online], Duben 2013 [cit. 30. 4. 2013].
URL http://radimrehurek.com/gensim/intro.html

Příloha A

Obsah CD

Přiložené CD obsahuje:

- Zdrojové kódy písemné zprávy včetně obrázků
- Písemnou zprávu ve formátu PDF
- Skript pro opravu chyb v německo-českém slovníku `ncs-lmf_correct.py` a jeho výstup `ncs-lmf-corrected.xml`
- Skripty pro analýzu a jejich výsledky:

Skript	**Analýza**	**Soubory s výsledky**
`dbxml3_cz.py`	české části slovníků	`dbxml3_cz/dictform01/*` `dbxml3_cz/dictform02/*`
`dbxml3_ncs.py`	německo-český slovník (OCR)	`ncs_unknown`
`dbxml3_lsa.py`	sémantická analýza	`lsa_output`

Tabulka A.1: Skripty a soubory s jejich výsledky

- Soubor **dicts/slovnik_data.txt**, který obsahuje anglicko-český slovník použitý pro překlady v sémantickém analyzátoru.

Všechny skripty a jejich výstupy jsou umístěny také v adresáři projektu `/mnt/minerva1/nlp/projects/dbxml3`.

Příloha B

Příklady slovníků ve formátech LMF, OLIF a XDXF

B.1 Lexical Markup Framework

```
<LexicalResource dtdVersion='16'>
    <GlobalInformation>
        <feat att='languageCoding' val='ISO 639-3'/>
    </GlobalInformation>
    <Lexicon>
        <feat att='language' val='ces'/>
        <LexicalEntry id='e31918g'>
            <feat att='partOfSpeech' val='noun'/>
            <Lemma>
                <feat att='writtenForm' val='koruna'/>
            </Lemma>
            <WordForm>
                <feat att='grammaticalFeature' val='ž'/>
                <feat att='writtenForm' val='koruna'/>
            </WordForm>
            <Sense>
                <Equivalent>
                    <feat att='language' val='eng'/>
                    <feat att='gloss' val='čelenka'/>
                    <feat att='writtenForm' val='diadem'/>
                </Equivalent>
                <Equivalent>
                    <feat att='language' val='eng'/>
                    <feat att='gloss' val='královská ap.'/>
                    <feat att='writtenForm' val='crown'/>
                </Equivalent>
                <Context>
                    <TextRepresentation>
                        <feat att='languageIdentifier' val='ces'/>
                        <feat att='text' val='královská koruna'/>
```

```
                    </TextRepresentation>
                    <TextRepresentation>
                        <feat att='languageIdentifier' val='eng'/>
                        <feat att='text' val='royal crown'/>
                    </TextRepresentation>
                </Context>
            </Sense>
        </LexicalEntry>
    </Lexicon>
</LexicalResource>
```

B.2 Open Lexicon Interchange Format

```
<olif OlifVersion='2.0'>
    <header>
        ...
    </header>
    <body>
        <entry lemmaUserId='e31918g'>
            <mono>
                <keyDC>
                    <canForm>koruna</canForm>
                    <language>cs</language>
                    <ptOfSpeech>noun</ptOfSpeech>
                    <subjectField>general</subjectField>
                </keyDC>
                <monoDC>
                    <monoMorph>
                        <gender>f</gender>
                    </monoMorph>
                </monoDC>
            </mono>
            <transfer>
                <keyDC>
                    <canForm>diadem</canForm>
                    <language>en</language>
                    <ptOfSpeech>noun</ptOfSpeech>
                    <subjectField>general</subjectField>
                </keyDC>
            </transfer>
            <transfer>
                <keyDC>
                    <canForm>crown</canForm>
                    <language>en</language>
                    <ptOfSpeech>noun</ptOfSpeech>
                    <subjectField>general</subjectField>
                </keyDC>
            </transfer>
```

```
        </entry>
    </body>
</olif>
```

B.3 Extensible Dictionary Exchange Format

```
<xdxf lang_from='CES' lang_to='ENG'>
<meta_info>
    ...
</meta_info>
    <lexicon>
        <ar>
            <k>koruna</k>
            <def>
                diadem
                <def>čelenka</def>
                crown
                <def>královská koruna</def>
            </def>
        </ar>
    </lexicon>
</xdxf>
```