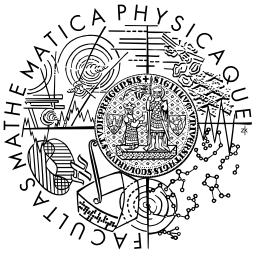


MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

**DIPLOMOVÁ PRÁCE**

Petr Fanta

# Kontextově závislý slovník pro překladatele

Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: RNDr. Ondřej Bojar, Ph.D.
Studijní program: Informatika
Studijní obor: Softwarové systémy

Praha 2016

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Kontextově závislý slovník pro překladatele
Autor: Petr Fanta
Ústav: Ústav formální a aplikované lingvistiky
Vedoucí diplomové práce: RNDr. Ondřej Bojar, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt: Při ručním překládání krátkých textů, jaké se vyskytují například na sociálních sítích, či mikroblozích (Twitter a podobně), je překladatel často nucen dohledávat doplňující informace v různých zdrojích. Může se jednat o méně běžná slova, o specifické termíny z neznámé domény, či o různé zkratky.

V této práci se zabýváme návrhem a implementací systému, který pro danou krátkou textovou zprávu automaticky sestaví minimální kontextově závislý slovník. Systém v překládaném textu vybírá vhodná hesla do slovníku a vyhledává k nim definice, překlady a příklady v otevřených zdrojích, či je automaticky extrahuje z paralelního korpusu. Získaný slovníček v ideálním případě bude pro překladatele již dostačujícím podkladem, aby překládanou zprávu s jistotou pochopil a zvolil odpovídající překladové ekvivalenty včetně odborných termínů. Empirické vyhodnocení se opírá o statistiky sledující, jak často byli uživatelé s navrženými hesly spokojeni, jak často byla hesla chybná a do jaké míry systém správně určil relevanci pro daný vstupní text.

Klíčová slova: kontextově závislý slovník desambiguace významu slov připojování významu pojmenovaným entitám výkladový slovník překladový slovník

Title: Context-Dependent Dictionary for Translators
Author: Petr Fanta
Institute: Institute of Formal and Applied Linguistics
Supervisor: RNDr. Ondřej Bojar, Ph.D., Institute of Formal and Applied Linguistics

Abstract: During a manual translation of short texts, such as texts occurring on social networks or microblogs (e.g., Twitter), translators are often forced to gather additional information from various sources. These can include less common words, domain-specific terms, or numerous abbreviations.

The aim of this thesis is to design and implement a system which automatically creates a minimal context-dependent dictionary for the given short message. The system identifies suitable dictionary entries in the translated text and searches for their definitions, translations, and examples from available open sources, or extracts them automatically from a parallel corpus. The resulted dictionary is ideally sufficient for human translators to understand the message, and to choose appropriate translation equivalent (including technical terms). An empirical evaluation is based on statistics which tracks how often users were satisfied with the proposed entries, how often the entries were incorrect and to what extent the system correctly identified the relevance for the input text.

Keywords: context-dependent dictionary word sense disambiguation entity linking explanatory dictionary translation dictionary

Chtěl bych poděkovat vedoucímu práce RNDr. Ondřejovi Bojarovi, Ph.D. za jeho cenné rady, připomínky, nápady a v neposlední řadě za čas a trpělivost, kterou mi věnoval.

Chtěl bych také poděkovat všem účastníkům provedených experimentů, že byli ochotni obětovat několik hodin a možná i dní svého života kvůli této práci.

Poděkování patří i mé rodině a přátelům, protože bez nich by tato práce nikdy nevznikla.

# Obsah

# Kapitola 1

# Úvod

Ačkoliv se v současnosti mnoho vědeckých prací zaměřuje na strojový překlad, překlad prováděný lidmi je stále nenahraditelný a neméně složitý. Ruční překlad klade velké nároky na znalosti překladatele, jenž přímo ovlivňují kvalitu překladu. Překladatel musí znát nejen zdrojový a cílový jazyk, ale i obecné poznatky v oblasti, kterou se překládaný text zabývá, aby dokázal textu porozumět a správně ho přeložit.

Jako příklad uveďme zprávu ze sociální sítě Twitter[1] (tweet) týkající se Ukrajinské krize: *Not that more proof is needed: Russian T-72B at Yenakievo checkpoint and Russian soldier.* Pokud se překladatel nezajímá o Ukrajinu, pravděpodobně nebude vědět, že *Yenakievo* je město v Doněcké oblasti na východě Ukrajiny, která je ovládána proruskými separatisty, a jak případně tento název přeložit. Podobný problém pro něj nejspíše bude i označení ruského tanku *T-72B*. Dohledání těchto informací sice není v dnešní době složité, ale výrazně snižuje výkonnost překladatele.

Daný problém se v současné době snaží řešit různé CAT (computer-assisted translation) nástroje pomocí překladových pamětí a terminologických databází, které jsou sestavovány z dříve provedených překladů, popřípadě ručně. Nevýhody takového přístupu jsou zřejmé:

- sestavování terminologických databází je náročné, a proto jsou zpravidla omezené na určitou doménu,
- překladová paměť poskytne nápovědu až při opakovaném výskytu překládaného slova, či fráze.

V této práci se zabýváme návrhem a vyhodnocením systému, který uživatelům poskytne podobné informace jako zmíněné CAT nástroje, ale tyto informace budou sestavovány zcela automaticky z různých zdrojů nezávisle na překladech provedených v minulosti, tedy bez znalosti řešení podobného problému řešeného dříve. Takový systém bude vhodným doplňkem současných CAT nástrojů, zejména při překladu založeném na crowdsourcingu (Šubert a Bojar, 2014), kdy překladatelé často naráží na texty z velmi odlišných domén a o to důležitější je poskytnout jim základní znalosti z pro ně nové oblasti.

[1] `https://twitter.com/`

## 1.1 Cíl práce

Cílem práce je vytvoření systému, který poskytne uživateli relevantní doplňující informace, jenž mohou být užitečné při překladu. Nelze přesně určit, jaké informace to jsou kvůli různým potřebám překladatelů, proto jsme se rozhodli omezit výstup systému na následující druhy informací:

- definice významů pojmenovaných entit v daném kontextu,
- definice významů slov v daném kontextu,
- překlady slov, či frází vhodné pro daný kontext,
- příklady použití pojmenovaných entit a konceptů ve stejném významu ve větách.

Myslíme si, že tyto informace jsou stěžejní pro vytvoření dobrého překladu. Na druhou stranu velké množství předkládaných informací může být pro uživatele nepříjemné až matoucí, proto se práce zabývá jednak vhodným výběrem informací, které jsou předkládány uživateli, jednak jejich filtrováním.

Jelikož prací zabývajících se podobným cílem existuje minimum (viz sekce 2.5), součástí textu je i návrh způsobu vyhodnocení takového systému. Navrhovaná metoda je založena na lidské anotaci výstupů systému, která je zaměřena na ověření správnosti a užitečnosti předkládaných informací a vyhodnocení výběru hesel, pro která jsou informace dohledávány.

## 1.2 Struktura práce

Text práce je rozdělen do osmi kapitol. První kapitola vysvětluje důvody k vytvoření této práce a popis jejich cílů. V druhé kapitole je problém představen z pohledu zpracování přirozených jazyků (NLP) a zbytek této kapitoly obsahuje přehled souvisejících prací a metod z NLP stěžejních pro naše řešení daného problému.

Kapitola 3 představuje jednotlivé zdroje informací, které jsou využity v našem řešení, jenž je teoreticky popsané v kapitole 4, popřípadě v kapitole 5, která obsahuje popis konkrétní implementace. Šestá kapitola obsahuje popis vyhodnocení našeho sytému.

V kapitole 7 navrhujeme možnosti vylepšení a směr, kterým by se mohl ubírat budoucí vývoj podobných systémů a poslední kapitola obsahuje shrnutí poznatků z této práce a diskusi.

# Kapitola 2

# Analýza

V této kapitole nejprve přiblížíme naší motivaci k vytvoření systému, který automaticky sestavuje minimální kontextově závislý slovník a pokusíme se nastínit typické problémy, které by měl takový systém řešit, a současné způsoby jejich řešení. Dále v této kapitole shrneme poznatky ze zpracování přirozeného jazyka počítači (natural language processing, NLP), které souvisí s naším řešením dané úlohy, a v závěru kapitoly představíme jiné systémy, které se zabývají podobným tématem jako tato práce, nebo mají podobné rysy jako naše řešení.

## 2.1 Twitter Crowd Translation

Prvotní motivací pro výzkum automatického sestavování kontextově závislých slovníků byl v našem případě systém Twitter Crowd Translation (TCT)[1] (Šubert a Bojar, 2014), proto ho i zde použijeme k demonstraci problémů, při kterých by takový slovník měl překladatelům pomoci. Nejprve však systém TCT krátce představíme.

TCT je infrastruktura pro rychlý překlad zpráv ze sociálních sítí, konkrétně Twitteru, založená na crowdsourcingu. Systém sbírá příspěvky od vybraných uživatelů Twitteru a distribuuje je pomocí e-mailu mezi registrované dobrovolníky s žádostí o překlad do určitého jazyka (viz výpis 2.1). Překlady jsou zasílány zpět do systému jednoduše jako odpovědi na tyto e-maily. Alternativně lze použít pro překlad webové rozhraní TCT. Získané překlady jsou shromažďovány, ohodnoceny a následně je nejlepší z nich opět vystaven na Twitteru.

```
Please translate the following post to language: Czech

#BREAKING IS confirms number two Omar al-Shishani killed in Iraq

ID:b3fc40b1d068699d6a2a7edb7a3aad31
```

Výpis 2.1: Ukázka e-mailu s žádostí o překlad.

Protože je délka tweetů je omezena pouze na 140 znaků, nemohou nést dostatečné množství informací. Jinak řečeno, kontext daného sdělení nemusí být z textu

[1] `http://tweeslate.com`

zcela zřejmý. Nicméně, historie tweetů jednoho uživatele je známá, dokonce s dalšími informacemi jako je čas odeslání, či místo odeslání, a lze předpokládat, že se některé tweety budou dotýkat stejného tématu, ale spoléhat na to nelze.

Naneštěstí TCT rozesílá jednotlivé tweety překladatelům naprosto izolované od jakéhokoliv kontextu, a proto mají pouze malou šanci textu plně porozumět. Mimoto se očekává, že překladatel pochopí význam tweetu bez užití nějakých pomocných nástrojů, jako jsou internetové vyhledávače, aby se neztrácela výhoda získaná prostřednictvím e-mailového rozhraní.

Prostředí sociálních sítí navíc přináší další specifické zvláštnosti, například velké množství překlepů a neustálených zkratek, náhodné zkracování slov, shluky slov netvořící věty, či špatný slovosled ve větách, hashtagy, odkazy na jiné uživatele sociální sítě a podobně.

Uveďme několik příkladů:

- *John Kerry: Russia must stop to support the separatists militarily.* `http://t.co/6GzKzBYsMa` — Některým tweetům lze snadno porozumět, ale je to porozumění dostatečné, když nevíme, kdo je *John Kerry*?

- *What Put said was in fact: yes, we were ready 2 use nukes 2 attack #Ukraine who gave away her nukes in exchange for our security guarantees.* — Některé tweety používají zkracování, které je možné považovat za nejasné, či nepřesné.

- *Today SMM accompanying workers/humanitarian convoy to hardhit Trokhizbenka (45km NW Luhansk): ppl without electricity/water/heat 4 many days* — Mnoho tweetů pak vyžaduje spíše specifické nebo místní znalosti: Co je *SMM*? Je *Trokhizbenka* město, vesnice, nebo nějaká oblast? Jak přeložit *hardhit*? I takové problémy se objevili v získaných překladech.

- *Welcome to the #Russian_invaders' "silence mode"! #Mortars instead of #Grad/s ...* `https://t.co/DvP3af8pXE` *#Ukraine* `http://t.co/HmbPM51lnO` — Jak přeložit slovo *mortar*? Co je *Grad*?

Jelikož je překlad v TCT založen na dobrovolnících, nemůžeme očekávat, že překladatelé budou experti na východní Evropu, či vojenskou techniku, nebo jakékoliv jiné téma, které se na Twitteru může kdykoliv objevit. Navíc překladatelé většinou nebudou rodilí mluvčí ve zdrojovém jazyce, zejména při překladu do méně rozšířených jazyků, jako je čeština. Lze tedy předpokládat, že pro ně budou výhodné další informace ať ze slovníků, nebo otevřených zdrojů jako je Wikipedie, která nám například prozradí, že *Grad* je typ Ruského raketometu, nebo že *mortar* je minomet, malta, hmoždíř, či vesnice v Indii. Poznamenejme však, že hypertextové odkazy obsažené v tweetech odkazující na obrázky, videa, či novinové články objasňující význam často i úplně nesmyslných sdělení.

Ačkoliv jsou zmíněné problémy vhledem k tomuto systému výraznější než v jiných situacích, věříme, že podobné problémy se mohou objevit i při překladu jiných, především krátkých, textů, proto se ve zbytku této práce nebudeme zabývat konkretně překladem Tweetů, ale překladem obecně.

## 2.2 Slovníky

Jak bylo řečeno dříve, v této práci se zaměřujeme převážně na způsob poskytnutí informací, které překladatel potřebuje hlavně při překladu textů z pro něj neznámé domény. V takové situaci předpokládáme, že překladatel zná gramatiku zdrojového i cílového jazyka, ale překládaný text může obsahovat pro překladatele neznámá slova a výrazy, nebo slova použitá v jiném významu. Běžným řešením tohoto problému je použití nějakého slovníku, který v ideálním případě poskytne všechny informace potřebné pro porozumění těmto slovům a jeho překlad. Poznamenejme, že všechna hesla ve slovnících jsou uvedena v nějakém základním tvaru, pro jednoslovná hesla je to typicky lemma.

Existuje více druhů slovníků vhodných pro podporu překladu, ale nejpoužívanější jsou asi tyto:

**Výkladové a encyklopedické slovníky** Výkladový slovník je jednojazyčný slovník, který obsahuje pro každé slovo seznam definic jeho významů. Tento seznam je často seřazen podle četnosti výskytů daného významu. Kromě definic hesla také často obsahují různé další informace o výslovnosti, či původu slov a podobně.

Podobně jako výkladové slovníky, encyklopedické slovníky obsahují definice významu různých podstatných jmen, zejména pojmenovaných entit (viz sekce 2.3).

**Překladové slovníky** Překladové slovníky jsou specializované slovníky používané pro překlad slov, či frází z jednoho jazyka do jiného. Tyto slovníky standardně poskytují překladové ekvivalenty pro jednotlivá slova a fráze ze vstupního jazyka bez ohledu na jejich význam. Problematická jsou v tomto přístupu víceznačná slova, tedy slova, která ve vstupním jazyce pokrývají několik významů, ale cílovém jazyce pro ně neexistuje jediné slovo, které popisuje stejné významy.

V takovém případě bývají překladové ekvivalenty doplněny o glosy, nebo krátké příklady upřesňující jejich význam. Jiným řešením daného problému je zpětné vyhledání ekvivalentu ve vstupním jazyce ke zvolenému překladu.

**Tezaury** Tezaurus je referenční příručka, či druh slovníku, který uživatelům nabízí slova seskupena podle podobného významu, tedy poskytuje seznam synonym, někdy i antonym. Hlavním smyslem takové příručky je nalezení slova, či slov, které se nejlépe hodí v dané situaci. Ačkoliv tezaurus obsahuje synonyma, neměl by být brán jako kompletní seznam synonym pro dané slovo. Na rozdíl od výkladových slovníků, hesla tezauru typicky neobsahují definice slov.

Problémem slovníků je jejich rozsah a aktuálnost. Vzhledem k velikosti slovní zásoby jednotlivých přirozených jazyků a rychlosti jejich vývoje a také vzhledem k rozsahu současných slovníků je takřka nemožné udržovat je kompletní a aktuální, proto se v řadě případů, kdy překladatel slovník potřebuje, může stát, že požadované informace nenalezne.

## 2.3 Pojmenované entity

Definice pojmenovaných entit je poněkud problematická a v různých zdrojích se výrazně liší. Často se pojmenované entity definují jako úseky textu, které lze zařadit do nějaké kategorie určené výčtem (například Ševčíková a kol. (2007)). Mohou to být například jména osob, geografické názvy, názvy společností či výrobků a podobně. Vyjmenování všech těchto kategorii je vždy velmi náročné a často dochází ke sporům, co ještě typ pojmenovaných entit je a co už nikoliv, proto se používají i obecnější definice — za pojmenovanou entitu budeme považovat vše, co v rámci textu označuje konkrétní osobu, věc, místo, nebo událost, kde nic z toho nemusí fyzicky existovat, ale daný text to jednoznačně určuje.

Význam pojmenovaných entit z pohledu této práce je zejména v propojení překládaného textu s širším kontextem. Jako příklad uveďme následující věty:

**Věta 1** President was assassinated.

**Věta 2** President J. F. Kennedy was assassinated.

Věta 1 nám toho mnoho neříká, pouze se dovídáme, že nějaký prezident byl zavražděn. Nicméně prezidentů je mnoho, jednak prezidenti různých států, jednak prezidenti různých korporací. Abychom větě porozuměli, potřebujeme získat kontext jiným způsobem, například slyšíme podobnou větu v rádiu, předpokládáme, že se jedná o prezidenta státu, ze kterého rádio vysílá. V případě 2. vety již jasně víme, kdo byl zabit, pokud známe presidenta J. F. Kennedyho. Z tohoto důvodu lze tvrdit, že pojmenované entity jsou klíčové pro porozumění textu. Protože význam pojmenovaných entit není většinou obecně známý, je vhodné do kontextově závislého slovníku zařadit definice jejich významu, zejména v případě krátkých textu, kde není možné vysvětlit všechny souvislosti.

Prvním krokem při zpracování pojmenovaných entit je jejich identifikace v textu a případně zařazení do nějaké kategorie. K tomu účelu slouží rozpoznávače pojmenovávaných entit (named entity recognizer, NER). Rozpoznávačů pojmenovaných entit existuje velké množství, jednak založených na ručně sestavovaných pravidlech, jednak na strojovém učení (Nadeau a Sekine, 2007), a dosahují na běžných textech velké spěšnosti – případě angličtiny až 90% F-míry (Straková a kol., 2013).

## 2.4 Desambiguace významu slov a připojování významu entit

Přirozené jazyky jsou typicky víceznačné, proto základním problémem při sestavování minimálního slovníku je zjednoznačnění významu slov, či pojmenovaných entit v daném kontextu. Tyto úlohy se v anglické literatuře označují jako word sense disambiguation (WSD) a entity linking (EL).[2] Ačkoliv byly v minulosti zkoumány odděleně, ukazuje se, že způsoby jejich řešení i jejich problémy jsou velmi podobné a je výhodné zkoumat je společně (Moro a Navigli, 2015; Moro a kol., 2014b). Obě úlohy lze rozdělit na následující kroky:

[2]Kvůli nedostatku vhodného českého názvosloví dále v textu používáme buď anglické názvy, nebo české překlady: desambiguace významu slov a připojování významu entit.

1. identifikace zmínek v textu,

2. vyhledání všech významů pro každou zmínku,

3. výběr nejvhodnějšího významu pro danou zmínku.

Hlavní rozdíly mezi WSD a EL jsou v použitých zdrojích (viz kapitola 3), které popisují významy slov (sense inventory) nebo pojmenovaných entit (knowledge base), a ve výběru zmínek v textu. Postup výběru významu a následného zjednoznačnění je pro obě úlohy stejný.

### 2.4.1 Identifikace zmínek

Při identifikaci zmínek jsou vybrány úseky analyzovaného textu, jejichž význam bude zjednoznačňován. Způsob výběru úseků textu je závislý na problému k jehož řešení je využita desambiguace významu slov, nebo připojování významu entit.

V případě WSD identifikace spočívá ve výběru jednotlivých slov, či úseků slov. Ta mohou být filtrována například podle slovních druhů a podobně.

Pro připojování významu entit je nutné nejprve v textu tyto entity vyhledat. Nejobecnějším řešením je využití nástrojů pro rozpoznávání pojmenovaných entit (viz sekce 2.3). Další možností je využití báze znalostí obsahující významy entit. Jelikož tato báze obsahuje všechny entity, které lze s jejím využitím zjednoznačnit, může systém s její pomocí entity i vyhledávat (například Guo a kol., 2013; Mihalcea a Csomai, 2007; Procházka a Smrž). V takovém případě je nutné nějakým způsobem vygenerovat, nebo jinak získat všechny tvary názvu entity a jiná její označení, které jsou následně vyhledány ve vstupním textu. Tento způsob je výhodný, pokud z nějakého důvodu selhávají jiné metody pro detekci zmínek, například pokud je vstupní text příliš krátký, nebo jinak nevhodný pro použití metod založených na strojovém učení (Guo a kol., 2013).

### 2.4.2 Zjednoznačnění významu

Výzkum desambiguace významu slov je velmi rozsáhlý (Navigli, 2009) a bylo prozkoumáno mnoho přístupů k tomuto problému. Jednotlivé přístupy lze rozdělit jednak podle reprezentace významu slov, jednak podle přístupu k řešení problému: od metod založených na slovnících (dictionary-based), či znalostech (knowledge-based), které využívají znalostí obsažených v lexikálních zdrojích, přes metody založené na učení s učitelem (supervised), ve kterých je používán pro každé slovo klasifikátor trénovaný na korpusu, jenž je ručně anotován významy slov, po metody založené na učení bez učitele (unsupervised, nebo také word sense discrimination), které shlukují výskyty slov, čímž reprezentují jejich význam. Všechny tyto přístupy většinou shodně definují kontext slova jako okno $N$ okolních slov, který vyžívají k zjednoznační významu slova.

Každý přístup má své výhody a nevýhody. Nejlepších výsledků dle různých vyhodnocení[3] dosahují algoritmy založené na učení s učitelem (Zhong a Ng, 2010;

[3]Vyhodnocením WSD systémů se zabývá workshop SemEval (například Moro a Navigli, 2015), dříve Senseval.

Shen a kol., 2013). Nicméně, tyto algoritmy vyžadují velké množství ručně anotovaných trénovacích dat, a proto je složité je přizpůsobit pro novou doménu, nebo jazyk (tzv. knowledge acquisition bottleneck).

Naopak unsupervised přístup zatím dosahuje nejhorších výsledků, ale tyto výsledky mohou být velmi ovlivněny reprezentací významů slov, která bývá při vyhodnocování mapována na nějaký známý slovník významů slov. Problém reprezentace významu je asi největším problémem těchto metod, i když se částečně týká i ostatních. Hlavní výhodou je pak nezávislost na jakékoliv manuální přípravě dat. Tento přístup tedy jako jediný neovlivňuje knowledge acquisition bottleneck a mohl by ho vyřešit.

Jelikož ručně sestavované slovníky, tezaury, lexikální databáze, případně ručně anotované korpusy výrazně ovlivňují úspěšnost metod založených na znalostech a učení s učitelem, lze tento problém považovat za největší problém desambiguace významu slov na základě jejich kontextu.

### 2.4.3 Desambiguace založená na znalostech

Metody založené na znalostech (knowledge-based nebo dictionary-based) využívají báze znalostí k určení významu slov v daném kontextu. Tyto metody obvykle dosahují menší úspěšnosti než přístupy založené na učení s učitelem, ale většinou mají lepší pokrytí významů díky velikosti dostupných dat. Nicméně se ukazuje, že použití vhodných vědomostních bází, či sémantických sítí (graph-based algoritmy) může výrazně zlepšit úspěšnost těchto algoritmů, dokonce mohou konkurovat supervised metodám (Ponzetto a Navigli, 2010).

Další výhodou tohoto přístupu je větší dostupnost vhodných zdrojů a nepodléhá tedy tak výrazně již zmiňovanému problému s ručně sestavovanými zdroji informací, proto lze tyto algoritmy snadněji adaptovat na různé domény a jazyky.

#### Leskův algoritmus a jeho varianty

Leskův algoritmus (Lesk, 1986) je základní knowledge-based algoritmus. Je založen na hypotéze, že existuje vztah mezi slovy, která se vyskytují společně v textu, a že tento vztah lze pozorovat i v definicích významů těchto slov. Algoritmus ve své původní verzi spočívá v porovnání definic, či glos, významů slova se všemi definicemi významů slov v daném kontextu. Slovu je pak přiřazen význam, jehož definice má nejvíce společných slov s definicemi okolních slov. Například ve frázi *time flies like an arrow* algoritmus porovná glosy pro *time* se všemi glosami slov *fly* a *arrow*, následně algoritmus porovná glosy *fly* s glosami *time* a *arrow* a tak dále. Algoritmus začíná pro každé slovo nanovo a nijak neuvažuje dříve přiřazené významy.

Tato varianta algoritmu vyžaduje velmi mnoho porovnání, pokud uvažujeme kontext $n$ slov, pak je potřeba spočítat $\prod_{i=1}^{n} |Senses(w_i)|$ překryvů pro každé slovo, kde $Senses(w)$ je množina všech významů slova $w$. Kvůli tomu byla navržena zjednodušená varianta tohoto algoritmu (Kilgarriff a Rosenzweig, 2000), která k určení významu slova využívá pouze překryv definic jeho významů a jeho kontext.

Porovnání těchto variant (Vasilescu a kol., 2004) ukazuje, že zjednodušená varianta překonává původní Leskův algoritmus jak v přesnosti (precision), tak

ve výtěžnosti (recall). Naneštěstí, přesnost obou variant je velmi závislá na přesném znění definic, takže absence určitého slova výrazně ovlivní výsledky. Dalším nedostatkem algoritmů je, že při výpočtu překryvu uvažují jen glosy popisující významy zjednoznačňovaného slova. Tyto glosy bývají často příliš krátké a neposkytují dostatečnou slovní zásobu k rozlišení významů.

To se snaží vyřešit rozšířený Leskův algoritmus (Banerjee a Pedersen, 2002). Tento algoritmus využívá při výpočtu překryvu rozšíření glos porovnávaných významů pomocí glos slov, která mají nějaký vztah s tímto významem v sémantické síti jako je WordNet (například slova nadřazené, podřazená; viz sekce 3.2), či BabelNet (Navigli a Ponzetto, 2010) a také zavadí jiný výpočet ohodnocení překryvů, který zvýhodňuje delší sekvence, jenž jsou shodné. Výpočet skóre pro nějaký význam $S$ slova $w$ lze formálně zapsat jako:[4]

$$score_{ExtLesk}(S) = \sum_{S':S \xrightarrow{rel} S' \vee S' \equiv S} |context(w) \cap gloss(S')|$$

kde $context(w)$ je množina slov tvořící kontext zjednoznačňovaného slova $w$ a $gloss(S')$ je množina slov z definice významu $S'$, který je buď ohodnocovaný význam samotný, nebo význam související pomocí relace $rel$.

Autoři ukázali , že využití definic ze souvisejících konceptů velmi pomáhá desambiguaci, dokonce algoritmus může při využití vhodné báze znalostí konkurovat nejlepším systémům (Ponzetto a Navigli, 2010).

### Desambiguace pomocí vyhledávání informací

Desambiguace významu slov je tradičně chápána jako mezistupeň ve zpracování přirozených jazyků, který lze využít ke zlepšení jiných úloh jako je například vyhledávání informací (information retrieval (IR)). Existují však systémy, které se snaží o využití technik pro vyhledávání informací k zjednoznačnění významu slov. Například námi navrhovaný systém (Fanta a kol., 2015) je založen na předpokladu, že lze pomocí vhodného dotazu vyhledat dokument, který popisuje význam daného slova.

Za dokument je v tomto přístupu považován záznam z nějakého inventáře významů slov (viz kapitola 3). Tento záznam popisuje jeden význam slova, typicky obsahuje identifikátor významu, slovo, glosu, či definici významu, případně i další informace, jako jsou definice souvisejících konceptů ze sémantických sítí, nebo morfologické značky, v závislosti na použitých zdrojích. Pro tyto dokumenty je vytvořen fulltextový index pomocí  nástroje pro vyhledávání informací.

Dotazy používané k desambiguaci jsou závislé na informacích, které jsou obsaženy v indexech, a nelze tedy popsat konkrétní podobu. Nicméně, vyhledávané dokumenty musí popisovat slovo, či pojmenovanou entitu, jehož význam je určován. Další části dotazu typicky vyhledávají slova, která tvoří kontext zjednoznačňovaného slova, v definici významu, případně v definicích souvisejících konceptů, podobně jako zjednodušený a rozšířený Leskův algoritmus (viz sekce 2.4.3). Výsledný význam slova je určen podle ohodnocení dokumentů popisujících významy nástrojem pro vyhledávání informací.

[4]Poznamenejme, že prezentovaná funkce je jednodušší variantou funkce z původního článku. Autoři původního článku používají porovnání s definicí, nikoliv s kontextem, podobně jako základní Leskův algoritmus.

### 2.4.4 Problémy desambiguace významu slov

Desambiguace významu slov i připojování významu entit je velmi složitý problém nejen pro počítače, ale i pro lidi. Pokud budeme předpokládat, že lidé dokáží vždy správně určit význam slova v určitém kontextu, nemůžeme očekávat lepší výsledky od automatické desambiguace, úspěšnost lidí můžeme tedy brát jako horní hranici úspěšnosti WSD systémů. Naneštěstí se ani lidé často nedokáží na významu slova shodnout. Pokud lidé mají za úkol přiřadit slovům v anglických textech významy definované ve WordNetu (viz sekce 3.2), shoda mezi anotátory je přibližně 72,5% (Snyder a Palmer, 2004).

Jako příklad uveďme věty obsahující pojmenovanou entitu *George Bush*:

**Věta 1** „George Bush fainted at a banquet hosted by the Prime Minister of Japan.“ (Leden 1992)

**Věta 2** „George Bush faiths, then falls after choking on pretzel.“ (Leden 2002)

V těchto větách nejde určit její význam bez dalšího kontextu, můžeme jen hádat zda se jedná o jednoho z amerických prezidentů, amerického biblistu, mladého politika (syna Jeba Bushe), či řidiče NASCAR.

Důvodem muže být nejasná definice pojmu významu slova (Kilgarriff, 1997). Význam slova je v zásadě nekonečně variabilní a závislý na kontextu a není tedy snadné významy rozdělit do různých nebo diskrétních dílčích významů. Přesnost přiřazování významů slovům je proto velmi závislá na granularitě jejich definic. Zatímco při hrubozrnném rozdělení jsou lidé schopni rozdíly dobře rozeznat, ale při jemném rozdělení tomu tak není, jelikož významy mohou být příliš volné, nebo se mohou překrývat.

Tento problém se týká i slovníků, jakožto inventářů významů. Nejenže slovníky mohou poskytovat naprosto rozdílné rozdělení slov podle významů, některé zdánlivě stejné významy se mohou lišit.

## 2.5 Podobné práce

Prací zabývajících se automatickým získáváním doplňkových informací pro překladatele je poměrně málo, nebo jsou špatně vyhledatelné kvůli neustálenému názvosloví. Jako příklad takové práce uveďme alespoň systém NERITS (Nebhi a kol., 2013).

Cílem tohoto systému je doplnění strojového překladu o informace o pojmenovaných entitách, jako jsou jména osob, organizací a míst, které pomohou uživateli s porozuměním. Systém nejprve text přeloží pomocí strojového překladače a následně se v cílovém jazyce pokusí vyhledat pojmenované entity a přiřadit k nim informace z DBpedie(Auer a kol., 2007).

Na druhou stranu existují i systémy, které nejsou primárně zaměřeny na překlad, a poskytují informace částečně splňující naší definici kontextově závislého slovníku. Příkladem takových systémů je systém Babelfy (Moro a kol., 2014a).

Tento systém provede desambiguaci významu slov a připojí význam k pojmenovaným ze vstupního textu a poté předloží uživateli text s vyznačenými koncepty a pojmenovanými entitami s jejich významy. Významy jsou překládány ve formě krátké glosy, která ho popisuje.

Ačkoliv byl systém vytvořen kvůli demonstraci možností WSD a EL, z našeho pohledu sestavuje jednoduchý významový slovník. Navíc systém umí pracovat s různými jazyky a to jak se vstupními texty, tak s glosami. To ještě více ukazuje vhodnost tohoto systému jako podpůrného nástroje pro překlad.

# Kapitola 3

# Zdroje znalostí

Kvalita automaticky sestavovaného slovníku velmi závisí na použitých zdrojích informací, které musí jednak poskytovat všechny druhy informací potřebné pro sestavení slovníku (viz sekce 1.1), jednak musí být vhodné pro postupy použité při jejich výběru, v našem případě zjednoznačňovaní významu slov a připojování významu pojmenovaným entitám (viz kapitola 4).

V dnešní době existuje mnoho různých zdrojů dat vhodných pro desambiguaci významu slov a připojovaní významu entit, jedná se zejména o ontologie, sémantické sítě, významové slovníky a jiné báze znalostí. Jako příklad uveďme BabelNet (Navigli a Ponzetto, 2010), či DBPedii (Auer a kol., 2007), ale existují i experimenty s použitím Google Search jako zdroje významů pro koncepty, či pojmenované entity (Sudarikov a Bojar, 2015). Tyto zdroje však často obsahují velmi málo textů a nejsou tedy vhodné pro získávání informací, které lze použít ve slovníku.

V našich experimentech jsme proto zvolili dnes již klasické zdroje informací pro WSD a EL: WordNet a Wikipedii[1]. Výhodou těchto zdrojů je, že obsahují mnoho informací, které lze přímo použít ve výkladovém slovníku (viz sekce 4.3.4) a jsou vhodné pro námi zvolenou metodu určování významů.

Posledním zdrojem použitým v našich experimentech je paralelní korpus CzEng. Tento zdroj se od ostatních výrazně liší svou strukturou a účelem. Zatímco Wikipedie a WordNet lze považovat za sémantické sítě, které jsou vhodné pro desmabiguaci významu, CzEng je z pohledu WSD a EL pouze prostý text. Automaticky sestavovaný paralelní korpus je však vhodným zdrojem překladových ekvivalentů. Získávání překladů z paralelních korpusů může zmírnit problémy ručně sestavovaných překladových slovníků, jako je aktuálnost a rozsah (viz sekce 2.2). Navíc paralelní korpusy bývají děleny po větách, které každému slovu poskytují kontext, a je tedy možné vybrat překlad vhodný v určité situaci.

Všechny použité zdroje jsou blíže představeny v následujících sekcích této kapitoly.

[1]Připojování významu entitám používající Wikipedii jako bázi znalostí se často označuje jako wikifikace podle systému *Wikify!* (Mihalcea a Csomai, 2007).

## 3.1 Wikipedie

Wikipedie[2] je mnohojazyčná webová encyklopedie, na jejíž tvorbě spolupracují dobrovolníci, je volně dostupná a poskytuje široké spektrum encyklopedických znalostí. Každý článek ve Wikipedii je reprezentován jako stránka, Wikipage, která obsahuje informace o konkrétním konceptu (například *Statue*), či pojmenované entitě (například *Statue of Liberty*).[3]

Titulek stánky je složen z lemma konceptu, který stránka popisuje, a případně upřesnění významu v závorkách, pokud je lemma víceznačné, například *Play (activity)* a *Play (theatre)*. Text stránky je částečně strukturovaný pomocí značkovacího jazyka, wikitextu, který umožňuje rozdělení článku do sekcí, vkládání tabulek a podobně.

Kromě toho Wikipedie zachycuje různé vztahy mezi stránkami. Jsou to:

**Přesměrování** Přesměrování je speciální druh Wikipage. Tyto stránky nemají žádný obsah kromě názvu stránky, která je cílem přesměrování, a slouží pro zachycení alternativních názvů konceptů, jenž je možné považovat za synonyma. Například stránka *USA* je přesměrování na stránku *United States*.

**Rozcestníky** Tyto stránky shromažďují odkazy na stránky se stejným, či podobným názvem a mohou v názvu obsahovat označení *(disambiguation)*, ale není to pravidlem. Rozcestníky zachycují homonymii a polysémii. Jako příklad uveďme stránku *Mercury*, která odkazuje na *Mercury (element)*, *Mercury (planet)*, či *Mercury (mythology)*, ale i na stránku *Freddie Mercury* a mnoho dalších článků.

**Interní odkazy** Stránky obsahují hypertextové odkazy, které propojují příbuzné koncepty v rámci Wikipedie. Například stránka *Aircraft* obsahuje odkazy *machine*, *fly* a *air* vedoucí na stránky *Machine*, *Flight*, *Atmosphere of Earth* a podobně.

**Mezijazykové odkazy** Mezijazykové odkazy propojují články a další stránky v jiných jazykových verzích Wikipedie. Naneštěstí nejsou všechny verze stejně obsáhlé a proto nemusí odkazy existovat, nebo mohou odkazovat pouze na část článku.

**Kategorie** Stránky Wikipedie mohou být zařazeny do jedné nebo více kategorii. Kategorie se mohou dále dělit na různé podkategorie. Například kategorie *Capitals in Europe* obsahuje články popisující evropská hlavní města, ale také podkategorii *Capital cities in the United Kingdom*, která obsahuje hlavní města ve Velké Británii.

Díky těmto vztahům lze Wikipedii považovat za graf, či sémantickou síť, kde jednotlivé stránky Wikipedie tvoří vrcholy a hrany jsou tvořeny různými vztahy mezi stránkami.

Wikipedie je velmi rozsáhlý zdroj informací, pravděpodobně největší volně dostupný zdroj encyklopedických znalostí. Její anglická verze[4] obsahuje téměř

[2] `https://www.wikipedia.org/`

[3] Dále v této sekci termín koncept kvůli zjednodušení označuje i pojmenované entity.

[4] Statistiky byly získány z výpisu databáze anglické a české Wikipedie z května 2015 poskytované v rámci projektu Kiwix (`http://www.kiwix.org/`)

6712209 stránek obsahujících text (článků, přesměrování a kategorií), z toho je 4853568 článků. Graf vytvořený z této verze obsahuje přes 231 milionů neorientovaných hran, pokud uvažujeme pouze interní odkazy. Jiné verze jsou však výrazně menší, například česká verze obsahuje pouze 320870 článků.

## 3.2 WordNet

WordNet (Miller, 1995; Fellbaum, 1998) je velká lexikální databáze nebo také sémantická síť pro anglický jazyk, která seskupuje slova do synonymických řad zvaných synsety, z nichž každý popisuje jiný koncept. Jako příklad uveďme synsety obsahující slovo *bike*: $\{motorcycle^1, bike^1\}_n$, $\{bicycle^1, bike^2, wheel^7, cycle^6\}_n$ a $\{bicycle^1, cycle^4, bike^1, pedal^1, wheel^4\}_v$, kde dolní index obsahuje slovní druh a horní index číslo významu v rámci jednoho slovního druhu, a tedy zjednoznačňuje víceznačná slova. Významy konceptu jsou číslovány podle počtu výskytu (nejčastější význam má nejnižší číslo).

Pro každý synset WordNet obsahuje krátkou obecnou definici, či glosu. Synset také může obsahovat jednu nebo více vět, které ukazují použití konceptů. Například synset $\{house^1\}_n$ má definici „*a dwelling that serves as living quarters for one or more families*“ a ukázkovou větu „*he has a house on Cape Cod*“.

Jednotlivé synsety ve WordNetu jsou propojeny pomocí lexikálních a sémantických vztahů. Zatím co sémantické vztahy mohou být pouze mezi synsety, lexikální vztahy mohou propojovat i jednotlivá slova a zachytit tak například vztah antonymie. Druhy sémantických vztahů se liší podle slovního druhu synsetu, například pro podstatná jména to jsou tyto vztahy:

**Is-a** Nejčastějším druhem relace mezi synsety je vztah nadřazený-podřazený. Do táto kategorie lze zařadit například slova nadřazená (hyperonyma) a slova podřazená (hyponyma).

**Instance-of** Tento druh relací vyjadřuje vztah mezi pojmenovanou entitou a jejím druhem. Například synset podstatných jmen obsahující koncept *Obama*[1] je instancí synsetu $\{President\ of\ the\ United\ States^1, United\ States\ President^1, President^2, \ldots\}_n$.

**Part-of** Posledním druhem sémantických vztahů jsou relace celek - část (holonyma) a část - celek (meronyma).

WordNet ve verzi 3.1 obsahuje 206941 významů rozdělených do 177659 synsetů pro 155287 slov.

## 3.3 CzEng

CzEng (Bojar a kol., 2012) je česko-anglický paralelní korpus, který je volně dostupný pro nekomerční využití. Ve verzi 1.0 obsahuje přibližně 15 milionů párů vět. Korpus je automaticky zarovnán na úrovni vět i slov. Korpus je dostupný v několika verzích, které obsahují různé stupně automatické analýzy. Jednotlivé verze obsahují od prostého textu přes automaticky určené morfologické značky až po povrchovou a hloubkovou syntaktickou analýzu.

Texty použité k sestavení korpusu byly vybrány z beletrie, legislativy Evropské unie, titulků k filmům, novin, technických dokumentací, paralelních webových stránek a projektu Navajo . Korpus neobsahuje sekvence delší než 15 po sobě jdoucích vět z jednoho zdroje a není tedy možné původní zdroje znovu sestavit.

Korpus byl sestaven zcela automaticky v několika krocích: rozdělení na věty, zarovnání vět pomocí nástroje Hunalign (Varga a kol., 2007), morfologické značkování a lemmatizace pomocí značkovače Morče (Hajič a kol., 2007) a pravidlového lemmatizéru pro angličtinu a zarovnání na úrovni slov pomocí GIZA++ (Och a Ney, 2000) včetně symetrizace pomocí různých technik. Závislostní analýza byla provedena pomocí Treex frameworku (Popel a Žabokrtský, 2010).

# Kapitola 4

# Anglicko-český slovník

V této kapitole představujeme náš systém sestavující minimální kontextově závislý slovník pro překlad z angličtiny do češtiny, který splňuje definici uvedenou v sekci 1.1, tedy systém, který dostane jako vstup text v anglickém jazyce a na výstupu vydá slovníková hesla a informace o nich, jednak v angličtině, jednak v češtině, dle jejich dostupnosti. Ačkoliv se v této práci omezujme jen na jeden jazykový pár, námi navrhované postupy jsou snadno přenositelné i na jiné jazyky, hlavním omezením pro adaptaci na jiné jazyky je malá dostupnost vhodných bází vědomostí.

Vytvářený slovník lze, vzhledem k obsaženým informacím, asi nejlépe zařadit na pomezí výkladového, či encyklopedického slovníku a slovníku překladového (viz sekce 2.2). O těchto typech slovníků předpokládáme, že jsou nejčastěji překladateli využívány pří vytváření překladu. Ověření tohoto předpokladu je pak součástí vyhodnocení systému (viz kapitola 6). Příklady slovníků vytvořených systémem jsou v příloze A.

Naším cílem bylo vytvoření systému, který bude co nejméně využívat metody založené na strojovém učení s učitelem. Výhoda takového přístupu je zjevná, systém je možné adaptovat na různé domény, či jazyky bez potřeby velkého množství ručně anotovaných dat. Lze však předpokládat, že metody založené na strojovém učení s učitelem by mohly dosáhnout lepších výsledků podobně jako u jiných úloh z NLP.

Dalším naším požadavkem na systém byla modularita, aby bylo možné snadno modifikovat proces vytváření slovníku, zejména nezávislost zpracování vstupního textu na použitých zdrojích. To sice přináší určité problémy (viz sekce 4.5), ale napomáhá snadné rozšiřitelnosti o nové zdroje informací, či budoucímu vývoji systému (viz kapitola 7). Z tohoto důvodu jsme proces vytváření slovníku rozdělili do několika oddělovaný částí, jsou to:

1. předzpracování vstupu,
2. výběr kandidátů na hesla slovníku,
3. vyhledání relevantních informací o kandidátech,
4. finální sestavení slovníku.

Dále jsou v této kapitole popsány postupy použité v jednotlivých krocích při sestavování kontextově závislého slovníku naším systémem a v závěru diskutujeme nevýhody námi zvoleného přístupu. Systém jsme implementovali pomocí

jednoduchého frameworku, který zajišťuje provádění těchto kroků a je popsán v kapitole 5.

## 4.1 Předzpracování

Cílem úvodní fáze je příprava vstupního textu pro pozdější zpracování. Během této fáze je provedena nejprve segmentace a tokenizace vstupního textu. Následně jsou k tokenům doplněna lemmata, morfologické značky a jsou označeny pojmenované entity (viz 2.3), které jsou využity při výběru kandidátů na hesla slovníku.

K tomuto účelu využíváme značkovače MorphoDiTa a NameTag (Straková a kol., 2014) včetně modelů pro angličtinu a češtinu, se kterými jsou tyto nástroje distribuovány. Poznamenejme, že toto je jediná našeho část systému, která využívá strojové učení s učitelem.

## 4.2 Výběr hesel slovníku

Jedním ze stěžejních kroků při sestavování minimálního kontextově závislého slovníku je výběr slovníkových hesel ze vstupního textu, ke kterým budou v následujících krocích vyhledávány informace do slovníku. Jako součást tohoto kroku v našem systému však považujeme kompletní zpracovaní vstupního textu, tedy i výběr kontextu pro každé heslo, který je využíván k určení jeho významu, a případně vyhledání koreferencí (viz sekce 7.4). V dalších krocích již se vstupním textem nijak nepracujeme.

V rámci našeho systému jsme se rozhodli rozlišovat slovníková hesla, která jsou tvořena pojmenovanými entitami, a hesla z ostatních slov, které označujeme jako koncepty. Jelikož i pojmenované entity jsou tvořeny koncepty rozlišujeme i speciální případy, tedy koncept v úseku slov tvořících pojmenovanou entitu a jednoslovné pojmenované entity, které zároveň tvoří koncept. Například z věty „*The European Medicines Agency (EMA) is a European Union agency for the evaluation of medicinal products.*“ může být vybrána pojmenovaná entita *European Medicines Agency*, ale i koncept *european*. Toto rozdělení je užitečné zejména při sestavování výsledného slovníku, jelikož jednotlivé typy hesel mohou mít pro překladatele různý význam (viz sekce 2.3) a mohou pomoci při výběru vhodných zdrojů informací pro dané heslo.

Protože je naším cílem poskytnout překladateli slovník, který obsahuje všechny informace potřebné k překladu vstupního textu, ale který nesmí být zbytečně příliš obsáhlý, aby uživatele nezahltil, je důležité nějakým způsobem odhadnout jaká hesla uživatel zná, nebo jsou jinak nedůležitá a není nutné je zahrnout do výsledného slovníku. Tuto filtraci je vhodné provést v různých fázích zpracování, jednak během výběru hesel, jednak během sestavování výsledného slovníku (viz sekce 4.4). Důvod k rozdělení je zřejmý: čím později bude filtrace provedena, tím více úkonů bude provedeno zbytečně, ale v pozdějších fázích mohou být dostupné informace užitečné pro přesnější filtrování. Příkladem může být význam přiřazený kandidátovi během zjednoznačňování, či charakter nalezených informací o daném heslu.

V této fázi jsou nejprve vybrány za kandidáty všechny pojmenované entity a koncepty tvořené jedním tokenem (včetně tokenů tvořících pojmenované entity) a jsou zařazeny do odpovídající kategorie. Při výběru předpokládáme, že víceslovné koncepty jsou buď pojmenované entity, nebo je možné je rozdělit mezi více slovníkových hesel aniž by docházelo ke změně významu. Následně je provedena filtrace vybraných kandidátů, jsou ponecháni kandidáti, kteří:

a) jsou pojmenované entity,

b) nejsou interpunkční znaménka,

c) nejsou tvořeni pouze jedním znakem,

d) nejsou tvořeni pouze číslicemi,

e) se neskládají pouze ze stopslov,

f) jsou tvořeni slovy, která byla odhadnuta jako neznámá na základě pravděpodobností výskytu jejich lemmat v jazyce získaných metodou maximální věrohodnosti z korpusu, tedy pokud je pravděpodobnost menší než konstanta. V našem případě z anglické části paralelního korpusu CzEng.

Poté je ke kandidátům, kteří prošli filtrací, doplněn kontext, který bude později použit k zjednoznačnění jejich významu. Tento kontext je tvořen ostatními tokeny z věty, ve které se kandidát nachází ve vstupním textu.

## 4.3 Vyhledávání informací

Během toho kroku v procesu vytváření slovníku jsou vyhledány relevantní informace o kandidátech na slovníková hesla v použitých zdrojích (viz kapitola 3) a jejich úprava pro použití ve slovníku. Vyhledávání je založeno na poznatcích ze zjednoznačňovaní významu slov a připojování významu pojmenovaným entitám (viz sekce 2.4). V našem systému jsme použili metodu popsanou v článku Fanta a kol. (2015) založenou na vyhledávání informací (viz sekce 2.4.3). Na rozdíl od metody uvedené v původním článku jsme však použili jinou strukturu indexu (viz sekce 4.3.2), která zohledňuje strukturu použitých zdrojů informací podobně jako rozšířený Leskův algoritmus (Banerjee a Pedersen, 2002).

Kvůli různorodosti použitých zdrojů a informací, které tyto zdroje poskytují, jsme použili pro každý zdroj samostatný podsystém. Všechny tyto podsystémy však sdílí základní strukturu použitého algoritmu. Nejprve je pro každého kandidáta zkonstruován dotaz pro vyhledávací engine (information retrievel engine) z tokenů tvořících heslo a kontext a ten je následně použit k vyhledání nejvhodnějších dokumentů[1] v indexu daného zdroje. Tato úloha se může opakovat s různými dotazy v závislosti na výsledku předchozího vyhledávání. Nejdříve jsou pří vyhledávání použity co nejpřesnější dotazy a pokud neuspějí, jsou případně použity dotazy obsahující slabší podmínky. Z výsledků vyhledávání jsou pak vybrány dokumenty, z nichž jsou extrahovány informace pro slovník.

[1]Dokumentem rozumíme článek z Wikipedie, synset z WordNetu, nebo jeden pár vět z korpusu CzEng

### 4.3.1 Příprava dat

Ačkoliv použité zdroje poskytují potřebné informace, nejsou primárně určeny pro tento typ úloh a námi použité algoritmy a bylo je nutné pro použití v našem systému upravit. Úpravy byly provedeny zejména v datech Wikipedie a v paralelním korpusu CzEng, jelikož WordNet svou strukturou vyhovuje základním požadavkům.

Články z Wikipedie jsme převedli na prostý text, který je jednak nutný pro vytváření indexů, jednak lépe zpracovatelný při sestavování slovníku. Během převodu byly z článku odstraněny veškeré části nesouvisející tématem, které článek popisuje, a informace nevhodné pro určování významu pomocí vyhledávání informací. Například byl z textu, kromě formátování, odstraněn obsah, *infoboxy*, reference, externí odkazy, či informace o neúplnosti článku a podobně.

Abychom však neztráceli užitečné informace, které byly z textů vypuštěný převodem na prostý text, extrahovali jsme pro každý článek seznam alternativních názvů, aliasů, a seznamy článků, na něž článek odkazuje a které naopak odkazují na daný článek. Za aliasy článků jsou v těchto seznamech považovány názvy stránek s přesměrováním a kotvy interních odkazů vedoucích na daný článek (viz sekce 3.1).

V případě paralelního korpusu CzEng jsme odebrali všechny páry vět, které z našeho pohledu nevyhovují algoritmu použitému pro vyhledávání, například příliš krátké věty, které neobsahují dostatečný kontext pro desambiguaci, ale vyhledávací modely jim přiřazují příliš velké ohodnocení. Odstraněny byly všechny páry vět, jejichž anglická část

a) obsahuje méně než čtyři tokeny,

b) neobsahuje žádné určité sloveso,

c) nezačíná velkým písmenem,

d) nekončí interpunkčním znaménkem pro ukončení věty (tečkou, vykřičníkem, nebo otazníkem).

Tato operace odfiltrovala přibližně 46,5% ze všech párů obsažených v korpusu CzEng.

Konkrétní detaily přípravy zdrojů pro naší implementaci jsou popsány v příloze B.1.

### 4.3.2 Indexace zdrojů

Pro každý zdroj jsme vytvořili fulltextový index pomocí knihovny Apache Lucene.[2] Jelikož Fanta a kol. (2015) ukázali, že volba modelu pro ohodnocování dokumentů (ranking) v Lucene má malý vliv na výsledky desambiguace a nelze určit, který je výhodnější, použili jsme v indexech pouze model založený na TF-IDF.[3]

[2] `http://lucene.apache.org/core/`

[3] `http://lucene.apache.org/core/5_4_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html`

V ideálním případě by měli všechny zdroje indexy se stejnou strukturou, ale velké rozdíly mezi nimi to neumožňují. Struktura jednotlivých indexů, kromě identifikátoru dokumentu, je následující:

**Wikipedie**

**Název článku** Název článku bez upřesnění významu obsahu (viz sekce 3.1). V indexu se tato položka vyskytuje ve dvou variantách, jednak normalizována analyzátorem pro anglický jazyk z Lucene[4], jednak pouze převedena na malá písmena.

**Alternativní názvy** Aliasy názvu článku získané z přesměrování a odkazů vedoucích na článek (viz sekce 4.3.1). Stejně jako v u předchozí položky indexu jsou i aliasy indexovány ve dvou variantách.

**Text** Předzpracovaný text článku zpracovaný pomocí analyzátoru pro anglický jazyk.

**Názvy okolních článků** Seznam názvů článků bez upřesnění významu obsahu, z kterých vedou odkazy zpracovávaný článek a na které vedou odkazy z tohoto článku. Názvy jsou zpracovány pomocí analyzátoru.

**WordNet**

**Slova** Seznam lemmat všech slov ze synsetu. Stejně jako v případě názvu článku v indexu Wikipedie jsou slova indexovaná ve dvou variantách.

**Glosa** Glosa popisující význam synsetu a příklady použití slov ze synsetu, pokud jsou součástí WordNetu. Tato položka je zpracována pomocí analyzátoru.

**Slovní druh** Slovní druh synsetu.

**Příbuzná slova** Slova ze synsetů spojených se zpracovávaným synsetem pomocí nějakého vztahu (viz sekce 3.2). Slova jsou zpracována analyzátorem.

**Příbuzné glosy** Glosy příbuzných synsetů. Glosy jsou zpracovány analyzátorem.

**CzEng**

**Anglická věta** Anglická věta z větného páru. Věta je indexována jednak s použitím analyzátoru pro angličtinu jako v předchozích případech, jednak analyzátorem, který provádí stejné úpravy kromě stematizace.

### 4.3.3 Výběr relevantních dokumentů

Pro vyhledávání dokumentů v indexech jsou použity různé dotazy v závislosti na zdrojích. Protože Wikipedie a WordNet mají podobnou strukturu, lze je považovat za sémantické sítě (viz kapitola 3), je podobný i způsob dotazování do jejich indexů, naopak CzEng je velmi odlišný, zejména tím, že dokumenty nepopisují význam konceptů, či pojmenovaných entit, to se týká jak dotazování, tak zpracování výsledků.

[4]Analyzátor pro angličtinu odebírá stopslova a ostatní slova převádí na malá písmena stematizuje pomocí Porterova algoritmu(Porter, 1980).

V případě Wikipedie a WordNetu dotazy typicky vyžadují přítomnost textu kandidáta v názvu článku, či mezi jeho aliasy v případe Wikipedie nebo mezi slovy synsetu v případě WordNetu, čímž jsou vybrány všechny dokumenty popisující nějaký význam textu kandidáta, a vyhledávají text kandidáta společně se slovy z kontextu v ostatních položkách indexu, což určuje ohodnocení jednotlivých významu. V případě WordNetu je navíc vyžadován stejný slovní druh, pokud ho lze pro kandidáta určit. Pro každého kandidáta jsou použity nejvýše dva dotazy. První vyžaduje přesnou shodu mezi základním tvarem textu kandidáta a názvem článku, či mezi jeho aliasem, nebo slovem ze synsetu. Pokud tento dotaz neuspěje, je použit podobný dotaz, který používá stematizaci při vyhledávání textu kandidáta v těchto položkách indexu. Z nalezených dokumentů je použit dokument, který získá pro daný dotaz nejvyšší skóre od vyhledávacího enginu.

Jelikož CzEng poskytuje oproti ostatním zdrojům výrazně méně informací vhodných pro výběr vět obsahujících text kandidáta ve stejném významu jako ve vstupním textu, je použit i jednodušší dotaz, který vyžaduje, aby dokument obsahoval text kandidáta po analýze bez použití stematizace (viz sekce 4.3.2) a slova z kontextu s jejím použitím. První část dotazu má opět zajistit výběr správných vět, druhá potom slouží k jejích ohodnocení. Z nalezených dokumentů je v následujících krocích použito $N$ nejlepších, kde $N$ závisí na počtu výsledků a je omezeno shora konstantou.

### 4.3.4 Vytváření informací

Z vybraných dokumentů jsou následně navrženy informace, které mohou být použity ve výsledném slovníku. Tyto informace jsou rozděleny do tří kategorii: a) definice, b) překlad, c) příklad. Definice jsou navíc rozlišeny i podle jazyka, tedy na české a anglické.

Každý zdroj poskytuje informace z různých kategorii:

**Wikipedie** Wikipedie poskytuje zejména definice významu kandidátů, anglické i české, a případně i překlad textu kandidáta. Jako definici konceptu nebo pojmenované entity používáme první větu nalezeného článku, což je možné díky doporučenému stylu pro psaní příspěvků do Wikipedie.[5]

Česká definice a překlad je závislý na mezijazykových odkazech (viz sekce 3.1). Pokud ve Wikipedii existuje český ekvivalent anglického článku, je z něj definice získána stejným způsobem jako v případě angličtiny a jako překlad je použit název českého článku, nicméně překlad používáme jen v případě, že původní anglický článek má přesně stejný název jako je text kandidáta, tedy byl nalezen pomocí prvního dotazu do indexu Wikipedie.

**WordNet** WordNet poskytuje především definice a příklady použití konceptů a méně často i pojmenovaných entit, které jsou extrahovány z vybraného synsetu. Tyto informace není třeba v systému nijak upravovat, pouze jsou odebrány příklady, které neobsahují text kandidáta. K definici jsou navíc přidána synonyma, která jistě mohou pomoci s porozuměním.

[5]První věta každého článku Wikipedie by měla odpovědět na otázku: Kdo nebo co je předmětem článku a proč je významný? Viz `https://en.wikipedia.org/wiki/Wikipedia:Writing_better_articles#Lead_section`.

**CzEng** CzEng se od ostatních zdrojů odlišuje tím, že je zpracováváno více dokumentů. Z vybraných vět je vytvořen překlad textu kandidáta (viz sekce 4.3.5) a každé vybrané věty je navíc vytvořen příklad jeho použití.

### 4.3.5 Vytváření překladu z paralelního korpusu

Vytváření překladu z vět nalezených v korpusu CzEng je založeno na předpokladu, že slovo použité v určitém významu bude ve většině případů překládáno do jiného jazyka stejně. Algoritmus nejprve vytvoří všechny možné překlady textu kandidáta pomocí každého z nalezených párů vět a následně vybere nejčastější překlad. Pokud má více překladů nejvyšší počet výskytů, je vybrán překlad, který vznikl z dokumentu s nejlepším ohodnocením při vyhledávání.

Tento způsob výběru je výhodný ze dvou hledisek. Jelikož je zarovnání vět a slov v korpusu CzEng prováděno zcela automaticky, může nastat situace, kdy zarovnání není správné, a výběr překladu z více vět může tento nedostatek eliminovat. Druhou výhodou je výběr lepšího překladu pro danou situaci vzhledem k víceznačnosti přirozených jazyků. Počet požitých párů vět je však nutné vhodně omezit, aby nebyl vybrán nejčastější překlad daného hesla v paralelním korpusu místo nejlepšího překladu v daném významu, jelikož vyhledávání vždy najde všechny větné páry obsahující překlad daného hesla ohodnocené podle podobnosti kontextu hesla a anglické věty. V našem systému jsme počet vybraných vět omezili konstantou již při vyhledávání (viz sekce 4.3.3).[6]

Takto získané překlady je výhodné pro použití ve slovníku nějakým způsobem normalizovat, aby byl zachován způsob organizace použitý v klasických slovnících (viz sekce 2.2). Pokud bychom chtěli zachovat slova v určitém tvaru, objeví se problém s různou bohatostí morfologie jazyků. Pro výrazy, které jsou překládány na jedno slovo, lze jako normální tvar použít jejich lemmata. V případě překladů skládajících se z více slov je situace poněkud složitější, v českém jazyce je možné použít například metodu popsanou v práci Kubát (2014).

## 4.4 Sestavování slovníku

Závěrečnou fází vytváření slovníku je jeho sestavení. Během této fáze jsou vybrány informace, které budou použity ve finálním slovníku, z informací poskytnutých vyhledávacími podsystémy a případně mohou být odstraněna i některé slovníková hesla. Odebrána jsou například hesla, ke kterým nebyly nalezeny žádné informace. Protože je naším cílem vytvoření minimálního slovníku, je ke každému slovníkovému heslu vybrána nejvýše jedna definice, příklad a překlad. Výjimku tvoří definice, které mohou být vybrány zároveň v obou jazycích.

Jelikož výstupy vyhledávacích podsystému jsou v našem případě neporovnatelné, je nutné při výběru informací použít metodu nezávislou na předchozích krocích. Definice a překlady však není možné porovnávat bez nějakého ohodnocení správnosti a to v této fázi není možné získat, proto jsme použili pravidla, která vyberou nejvhodnější informaci podle zdroje, ze kterého pochází.

[6] Při vytváření testovacích slovníků (viz kapitola 6) bylo průměrně nalezeno 550 větných párů pro pro jedno slovníkové heslo a byla použita konstanta $N = 50$. Pro 27% hesel však nebyl v indexu nalezen žádný větný pár.

Pro slovníková hesla tvořená pojmenovanou entitou má největší prioritu definice z Wikipedie, pokud ta neexistuje použije se definice z WordNetu. Pro slovníková hesla tvořená konceptem je pravidlo opačné, tedy přednost má definice z WordNetu. Pro překlady je pravidlo jednodušší, pokud existuje překlad z Wikipedie, je použit vždy, jinak se použije překlad získaný z CzEngu, případně žádný.

Výběr příkladů je založen na myšlence, že překladatelům nejvíce pomůže pokud je neznámé slovo ve větě doplněno zejména slovy známými, jinak řečeno pokud v příkladu nerozumí pouze vysvětlovanému slovu. Všechny příklady získané v předchozích krocích jsou proto ohodnoceny pomocí pravděpodobností výskytu lemmat jejich tokenů v korpusu, podobně jako je tomu při výběru kandidátů na slovníková hesla (viz sekce 4.2). Jelikož toto ohodnocení zvýhodňuje krátké příklady, které často neobsahují dostatečné množství informací, aby byly užitečné, jsou všechny příklady kratší než průměrná délka vybraných příkladů penalizovány za délku. Použitý vzorec pro ohodnocení příkladu tedy je:

$$\text{score}(e) = BR \cdot \prod_{t \in e} p(t)$$

$$\text{BR} = \begin{cases} 1 & \text{je-li } l(e) > a \\ \exp(1 - a/l(e)), & \text{je-li } l(e) \leq a \end{cases}$$

kde $e$ je množina tokenů příkladu bez interpunkce, $a$ je průměrný počet tokenů v příkladu a $l(e)$ je počet tokenů v příkladu. Ve výsledném slovníku je použit příklad s nejlepším ohodnocením.

## 4.5 Problémy systému

Hlavní nedostatky systému souvisí zejména s dekompozicí na dílčí úlohy a volbou postupu vyhledávání. Protože jsme jako způsob získávání znalostí do slovníků zvolili zjednoznačňovaní významu slov a připojování významu pojmenovaným entitám, není oddělení výběru slovníkových hesel od vyhledávání informací úplně vhodné. Znalost dostupných zdrojů již v úvodních fázích zpracování může usnadnit výběr slovníkových hesel (viz sekce 2.4.1) a zpřesnit vyhledávání informací. Také v případě nejasností při výběru správného významu lze použít nejčastější význam daného hesla v použitém zdroji informací (most frequent sense, MFS). Nicméně tento přístup vede k monolitickému podsystému pro každý zdroj informací a přináší nové problémy. Není však vždy možné tímto přístupem zpracovávat velmi objemné zdroje informací, v takovém případě je naopak výhodný námi navrhovaný postup (viz sekce 7.2).

Pokud budeme předpokládat, že všechny zdroje dokážeme zpracovat, vhodným řešením by mohlo být vytvoření jednoho zdroje, který by spojoval všechny ostatní. O tento přistup se snaží například BabelNet (Navigli a Ponzetto, 2010), který spojuje mnoho zdrojů, jako jsou WordNet, Wikipedie, Wiktionary a další, do jedné sémantické sítě. Takový zdroj by mohl zmírnit výše uvedené nedostatky a vyřešil by problém se slučováním výsledků různých podsystémů. Vytvoření takového zdroje však jistě přinese nové problémy, zejména kvůli různorodosti informací potřebných k sestavení slovníku.

Algoritmy pro zjednoznačňovaní významu slov a připojování významu pojmenovaným entitám se z pravidla snaží najít právě jeden nejvhodnější význam

pro danou situaci, což dělá i náš systém, ale při sestávání slovníku takový přístup může být považován za nesprávný kvůli víceznačnosti přirozených jazyků. Lepším řešením takové situace je předložení všech přípustných významů, ale ne vždy je možné všechny významy získat. Navíc v našem systému může být získání všech přípustných významů problematické, jelikož nedokážeme porovnávat význam definic z různých zdrojů. Řešením tohoto problému může být například již zmiňovaný univerzální zdroj.

Ze stejného důvodu je v našem systému problematické slučování slovníkových hesel, která popisují stejný význam, a jsou tedy duplicitní. Nicméně u krátkých textu je tento problém téměř zanedbatelný.

# Kapitola 5

# Implementace

Tato kapitola obsahuje popis naší implementace systému pro vytváření minimálního kontextově závislého slovníku. I když se v této práci zabýváme pouze sestavováním slovníků pro překlad z angličtiny do češtiny, aplikace je navržená tak, aby bylo co nejsnazší tento proces modifikovat, případně přidat implementaci pro další jazykový pár.

Aplikace je naprogramována v programovacím jazyce Kotlin[1], který je překládán do Java bajtkódu, a proto je možné aplikaci spustit na všech systémech, pro které existuje implementace Java Virtual Machine, lze jí tedy považovat za multiplatformní, avšak některé komponenty třetích stran použité při sestavování anglicko-českého slovníku jsou překládány do strojově závislého kódu a tedy platformově závislé a je nutné použít jejich verzi pro konkrétní systém.

Uživatelská dokumentace systému je uvedena v příloze B.

## 5.1 Architektura

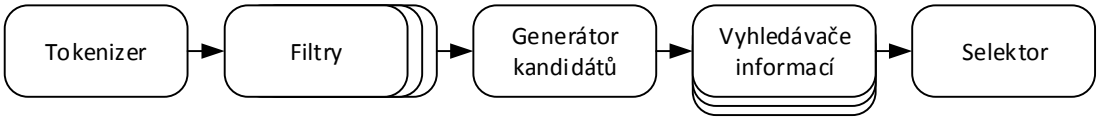
Systém je navržen jako jednoduchý framework, který zapouzdřuje proces vytváření slovníku a tokenizaci textu a zajišťuje přístup k těmto procesům pomocí webových služeb. Architektura frameworku se skládá ze tří vrstev: vrstvy webových služeb (či prezentační vrstvy), servisní vrstvy a aplikační vrstvy. První dvě vrstvy zajišťují zejména komunikaci s klienty systému a spravují komponenty z aplikační vrstvy, které zodpovídají za vytváření slovníků pro různé jazykové páry.

Vrstva webových služeb zodpovídá za zpřístupnění systému pomocí REST API. Skládá se z webového serveru (servletového kontejneru) a implementací konkrétních webových služeb, které tvoří fasády nad servisní vrstvou. Tuto vrstvu implementují třídy z balíčku *cz.cuni.mff.ufal.codetionary.internal.rest*.

Servisní vrstva je velmi jednoduchá: obsahuje pouze třídy, které se starají o výběr vhodné komponenty z aplikační vrstvy: pipeline (viz sekce 5.1.1), resp. tokenizéru v závislosti na požadované operaci. Třídy této vrstvy se nachází v balíčku *cz.cuni.mff.ufal.codetionary.internal.services*.

Hlavní třídou frameworku je třída *cz.cuni.mff.ufal.codetionary.DictionaryApplication*, která poskytuje přístup k horním vstávám frameworku. Třída umožňuje registraci jednotlivých pipeline a spuštění webového rozhraní.

[1] `https://kotlinlang.org/`




Obrázek 5.1: Pořadí komponent v pipeline

Framework je postaven nad *Spring frameworkem*[2], který používá pro vkládání závislostí (dependency injection) mezi jednotlivými komponentami systému. Vrstva webových služeb používá vložený webový server *Jetty*[3] a knihovnu *Jersey*[4] pro vytváření webových služeb.

### 5.1.1 Vytváření slovníku

Za vytváření slovníku zodpovídají tzv. roury (pipelines), které tvoří aplikační vrstvu frameworku a jsou volány pomocí tříd ze servisní vrstvy. Hlavní třídou je třída *Pipeline* z balíčku *cz.cuni.mff.ufal.codetionary.pipeline*, která zajišťuje volání jednotlivých kroků během vytváření slovníku (viz obrázek 5.1) a následné sestavení výsledného slovníku. Kromě metod na sestavování slovníku třída obsahuje také všechna důležitá metadata (název, zdrojový jazyk, cílový jazyk, podporované zdroje textů), které se využívají při výběru nejvhodnější pipeline pro zpracování požadavku.

Instance této třídy, které jsou neměnné (immutable object), je možné vytvářet pomocí třídy *PipelineBuilder* ze stejného balíčku. Aby byly vytvořené pipeline aplikaci dostupné, je nutné zaregistrovat je v instanci třídy*DictionaryApplication*, která slouží jako hlavní vstupní bod aplikace.

### 5.1.2 Programovatelné komponenty

Jednotlivé kroky sestavování slovníku zajišťují třídy implementující rozhraní z balíčku *cz.cuni.mff.ufal.codetionary.pipeline.components* a pro přenos dat mezi jednotlivými kroky používají datové třídy z balíčku *cz.cuni.mff.ufal.codetionary .pipeline.models*.

Jednotlivé kroky jsou tokenizace a segmentace (rozhraní *ITokenizer*), filtry pracující s tokeny (například lemmatizace; rozhraní *ITokenFilter*), generátor kandidátů do slovníku (rozhraní *ICandidateGenerator*), vyhledávače relevantních informací k jednotlivým kandidátům (rozhraní *IInformationSearcher*) a závěrečný výběr informací do slovníku (rozhraní *ISelector*). Tyto kroky jsou vykonávány v uvedeném pořadí, výjimku tvoří vyhledávače informací, které mohou být spuštěny paralelně. Poznamenejme, že filtry jsou spouštěny v pořadí v jakém byly přidány do instance třídy *PipelineBuilder* a mohou být na sobě závislé.

Pipeline v jednotlivých krocích používá různé reprezentace dat. Zatímco v prvních krocích, tokenizaci a filtraci, je vstupní text rozdělen pouze na věty a jednotlivé tokeny v nich (třídy *Document*, *Sentence* a *Token*), kandidáti na položky ve slovníku (třída *Candidate*) jsou pak reprezentováni jako tokeny, které je tvoří, a

[2] `http://projects.spring.io/spring-framework`
[3] `https://eclipse.org/jetty/`
[4] `https://jersey.java.net/`

kontext, tedy tokeny z jejich okolí. Kandidáti mohou být čtyř druhů a to: pojmenovaná entita, koncept, koncept v pojmenované entitě (například slovo *European* v pojmenované entitě *European Medicines Agency*) a jednoslovná pojmenovaná entita, která je zároveň konceptem. Jednotlivé druhy jsou reprezentovány výčtovým typem *EntryType*.

Podobně informace (třída *Information*) nalezené pro jednotlivé kandidáty obsahují různá metadata, například typ informace, či skóre vyjadřující důvěru v informaci, která lze využít při závěrečném výběru informací do výsledného slovníku.

Všechny metody těchto rozhraní komponent mají jako parametr, kromě potřebných modelů, také parametr typu *Context*. Během celého běhu algoritmu je předávána pouze jedna instance tohoto typu, proto je vhodný pro předávání doplňujících informací mezi nesouvisejícími kroky. Standardně tento parametr obsahuje parametry, se kterými byla pipeline volána, tedy netokenizovaný vstupní text, zdrojový jazyk, cílový jazyk, zdroj textu a název pipeline, což umožňuje modifikovat zpracování například podle zdroje textu vstupního textu. To je užitečné například při zpracování vstupů z různých sociálních sítí, které používají části textu se speciálním významem, například hashtagy.

Všechny komponenty musí být thread-safe, jelikož webové aplikace jsou téměř vždy více vláknové (záleží na nastavení servlet kontejneru).

## 5.2 Implementace anglicko-české pipeline

Implementace anglicko-českého slovníku (viz kapitola 4) je realizována pomocí konkrétních komponent pro výše popsaný framework.

Z pohledu architektury tato implementace přidává navíc datovou vrstvu, která poskytuje data potřebná k sestavování slovníků. Datová vrstva se skládá z databází obsahujících data z jednotlivých zdrojů (viz kapitola 3) a fulltextových indexů vytvořených pro tyto databáze. Naše implementace datové vrstvy používá relační databázový systém *MySQL*[5] pro uložení dat z české a anglické Wikipedie a dokumentovou databázi *MongoDB*[6] pro uložení dat z paralelního korpusu CzEng. Data WordNetu jsou již distribuována ve formě databáze. K vytvoření fulltextových indexů pro tyto databáze a následné dotazování je použita knihovna *Apache Lucene*[7]. Vytvoření těchto databází a indexu je blíže popsáno v příloze B.1.

Komponenty používané k sestavení anglicko-české pipeline se nachází v balíčcích vnořených ve jmenném prostoru *cz.cuni.mff.ufal.codetionary.components*. Úvodní fáze zpracování jsou implementovávány pomocí knihoven *MorphoDiTa*[8] a *NameTag*[9]. Pro tyto knihovny implementace obsahuje adaptéry, které zajišťují tokenizaci a segmentaci (třída *EnglishTokenizer*), lemmatizaci a morfologické značkování (třída *MorphologyFilter*) a rozpoznávání pojmenovaných entit (třídy *NERFilter* a *NERWebFilter*). Díky tomu, že *NameTag* používá interně knihovnu *MorphoDiTa*, nedochází při kombinování těchto knihoven k problémům s různým dělením vstupního textu a podobně.

[5] `https://www.mysql.com/`
[6] `https://www.mongodb.com/`
[7] `http://lucene.apache.org/core/`
[8] `https://github.com/ufal/morphodita`
[9] `https://github.com/ufal/nametag`

Výběr potenciálních slovníkových hesel na základě předchozích kroků zajišťuje třída *CandidateGenerator*. Chování této třídy je blíže popsáno v sekci 4.2.

Vyhledávání relevantních informací ve fulltextových indexech a jejich získávání z databází datové vrstvy obstarávají třídy *WikiSearcher*, *WordNetSearcher* a *CzEngSearcher*. Tyto třídy kromě výběru a získávání informací do podoby, v které jsou vkládány do výsledných slovníků. Tedy například třída *WordNetSearcher* ze získaných dat generuje překlady a podobně.

Poslední důležitou třídou a komponentou anglicko-české pipeline je třída *ProbabilitySelector*. Tato třída zajišťuje slučování výstupů z vyhledávačů a výběr informací a hesel do výsledných slovníků. Bližší popis chování této třídy je v sekci 4.4.

Podrobnější popis implementace vytváření anglicko-českého slovníku a frameworku lze nalézt v automaticky generované dokumentaci ze zdrojových kódů, která je součástí elektronické přílohy (viz příloha D).

# Kapitola 6

# Vyhodnocení

V této kapitole představujeme způsob vyhodnocení navrhovaného systému a jeho výsledky. Jelikož nám nejsou známy žádné jiné systémy řešící stejný, nebo podobný problém, které by byly podrobně analyzovány (viz sekce 2.5), navrhujeme zcela nový způsob vyhodnocení.

Způsob vyhodnocení je založený na ruční anotaci vstupních textů a výstupů systému. Cílem je zjistit správnost předkládaných informací a spokojenost uživatelů s těmito informacemi, tedy zda systém předložil informace pro všechna slova a pojmenované entity, o kterých uživatel požaduje nějaké doplňující informace a zda předložené informace uživatelé považují za užitečné. Součástí vyhodnocení je i analýza shody mezi anotátory, která ukazuje složitost problému.

## 6.1 Testovací data

K vyhodnocení systému byly využity texty ze sociální sítě Twitter, které se týkají se převážně ukrajinské krize, získané pomocí systému Twitter Crowd Translation (viz sekce 2.1). Vzhledem k této skutečnosti je sestavování slovníků náročnější než je tomu u běžných textů, jako jsou novinové články. Přispěvatelé často používají jazykové prostředky typické pro konkrétní sociální síť, celý systém byl však navrhován pro obecnou doménu. Například rozpoznávání pojmenovaných entit může dosahovat menší přesnosti pro takový druh textů. Výhodou této volby je otestování systému v podmínkách, které motivovaly jeho vznik a v kterých může být později nasazen. Příklady vstupních textů je možné nalézt v sekci 2.1 a příklady vygenerovaných slovníku v příloze A.

Tabulka 6.1 obsahuje základní informace o textech použitých při hodnocení a pro ně vytvořených slovnících.[1] Při získávání těchto statistik bylo použito automatické rozpoznávání pojmenovaných entit pomocí nástroje NameTag a jednoduchý algoritmus určující koreference mezi nimi,který je založen na porovnávání tokenů tvořících pojmenované entity. Proto jsou tyto statistiky jen přibližné.

[1] Poznamenejme, že při generování slovníků byla použita spíše malá filtrace slovníkových hesel — odebírána byla pouze hesla, jenž jsou tvořeny tokeny, které patří mezi 0,2% nejčastějších tokenů v anglické části korpusu CzEng (viz sekce 4.2).

| | |
|---|---|
| Počet dokumentů | 250 |
| Počet pojmenovaných entit | $\sim$ 454 |
| Počet dokumentů bez pojmenovaných entit | $\sim$ 4% |
| Počet unikátních pojmenovaných entit | $\sim$ 187 |
| Počet slovníkových hesel | 1832 |
| — počet pojmenovaných entit | 454 |
| — počet konceptů | 1378 |
| Počet informací | 5004 |
| — počet definic | 1729 |
| — počet překladů | 1563 |
| — počet příkladů | 1712 |

Tabulka 6.1: Základní informace o testovacích datech.

## 6.2 Anotační prostředí

Anotační prostředí (viz obrázek 6.1) se skládalo z textu, pro který je slovník sestavován, textového pole pro překlad, oblasti s vybranými úseky[2] vstupního textu, o kterých anotátor vyžaduje další informace, zaškrtávacího políčka, jenž označuje, že je vstupní text špatně tokenizován, ovládacích tlačítek a samotného slovníku.

Slovník pak obsahuje jednotlivá hesla a různé informace, které se k nim vztahují. Každá informace je zařazena do jedné ze tří kategorií: definice, překlady a příklady. Ke každé položce slovníku (heslu, či informaci) náleží dvě zaškrtávací políčka, která určují zda položka je správná nebo užitečná. Tato políčka mohou nabývat tří hodnot: neoznačeno, ano a ne, kde neoznačeno vyjadřuje nejistotu při výběru.

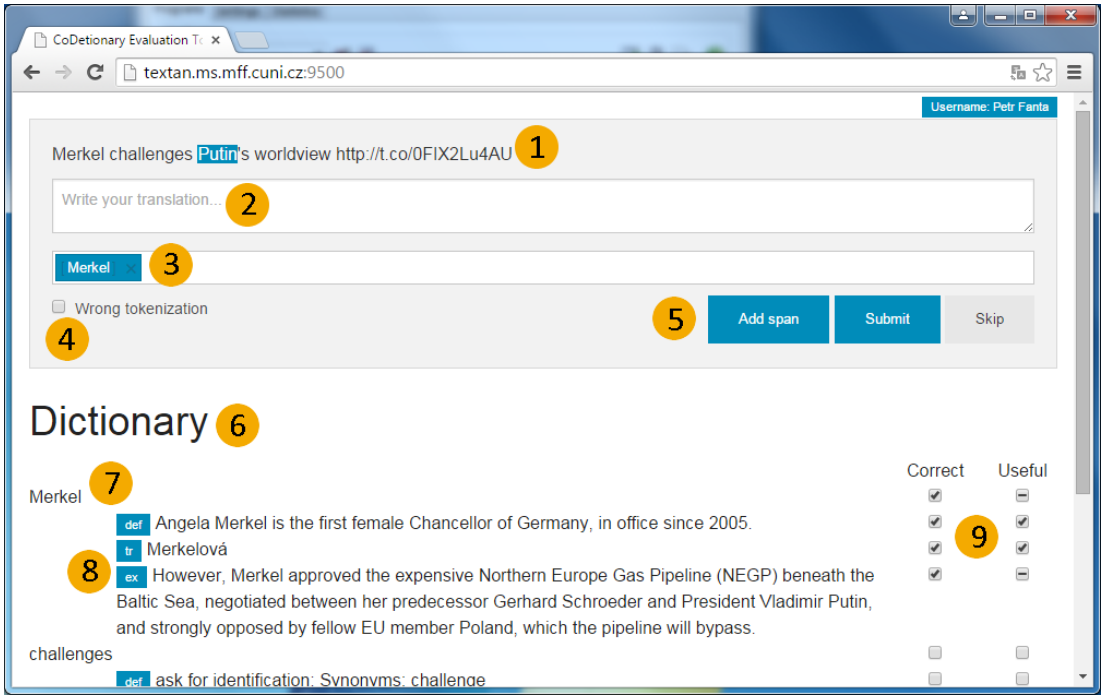
Technická dokumentace a popis nasazení anotačního prostředí je uveden v příloze C.

## 6.3 Průběh anotace

Anotování jednotlivých textů a k nim přiřazených slovníků se skládalo ze dvou částí. Během první části si měli anotátoři přečíst překládaný text a měli v něm vybrat všechny úseky, o kterých potřebují nějaké dodatečné informace, či vysvětlení, aby je dokázali správně pochopit a přeložit. V případě, že požadovaný úsek nešel v anotačním prostředí vybrat, měli označit vstupní text jako špatně tokenizovaný. Tato část měla být nezávislá na předloženém slovníku, měly tedy být vybrány i úseky, které již byly ve slovníku obsaženy.

Ve druhé části měli anotátoři v předkládaném slovníku označit správnost a užitečnost slovníkových hesel a jednotlivých informací o nich a přeložit vstupní text. Informace měla být označena jako správná, pokud popisuje stejný význam jako daný úsek ve vstupním textu. V případě překladů měl být navíc český ekvivalent v základní tvaru. Zdůrazněme, že význam měl být porovnáván výhradně se

[2]Jeden úsek se může skládat i z více nespojitých částí textu.




Obrázek 6.1: Anotační prostředí. 1) vstupní text, 2) pole pro překlad, 3) oblast s vybranými úseky, 4) označení špatné tokenizace, 5) ovládací tlačítka, 6) část se slovníkem, 7) heslo ve slovníku, 8) informace o hesle ve slovníku, 9) zaškrtávací políčka určená k ohodnocení hesel a informací.

vstupním textem, nikoliv s ostatními položkami v daném slovníkovém hesle. Užitečnost pak vyjadřuje názor anotátora, zda mu poskytnutá informace pomohla s porozuměním textu, či s překladem.

Slovníková hesla měla být považována za správná, pokud systém vybral správný úsek vstupního textu. Pokud například systém zahrnul do hesla interpunkční znamínko, mělo být heslo označené jako špatné. Heslo mělo být označené jako užitečné, pokud se anotátor domníval, že jakákoliv informace, ne nutně předložená ve slovníku, o daném heslu může vylepšit jeho překlad daného úseku. Poznamenejme, že užitečnost u hesel není shrnutí anotací informací předkládaných ve slovníku.

Důraz byl během anotace kladen zejména na označování správnosti informací, aby mohla být vyhodnocena přesnost výběru informací. Naopak v případě užitečnosti měli překladatelé velkou volnost. Například pokud se nějaké slovníkové heslo často opakovalo, mohl se jejich názor na užitečnost během zpracování změnit, jelikož si informace o daném hesle již zapamatovali a podobně. Jednotlivé dokumenty byly předkládány ve stejném pořadí všem účastníkům experimentu, čímž byly zajištěny pro všechny stejné podmínky, avšak bylo dovoleno přeskočit ohodnocování dokumentu, pokud byl vstupní text pro anotátora nesrozumitelný, například tweet: *@darlingrostov @PolskaSotnja @tombreadley @hut_oks @sws_himik в Brisbane* `http://t.co/9NdC4pkaVD`.

Lidé, kteří se účastnili experimentu, neměli žádné lingvistické vzdělání, ani se více nezabývali překladem a část z nich neměla ani vzdělání zaměřené na informatiku. Autoři anotací nebyli nijak obeznámeni s problematikou automatického sestavování minimálního kontextově závislého slovníku, kromě výše uvedených

instrukcí k vytváření anotací, a nemohli tedy být ovlivněni použitými technikami.

Reakce všech účastníků experimentu na jeho průběh byly velmi podobné. Často vyjadřovali překvapení nad složitostí úlohy, což bylo podpořeno i velkou nejistotou při jednotlivých úkonech.

## 6.4 Shoda mezi anotátory

Než představíme výsledky jednotlivých anotací, pokusíme se nejdříve vyhodnotit, jak je samotné ohodnocování slovníků složité. Toto hodnocení jsme založili na výpočtu shody mezi anotátory (inter-annotator agreement, IAA), což je míra toho, jak často se anotátoři shodnou při hodnocení dané položky. V naší práci tuto hodnotu definujeme jako procento případů, kdy se dvojice anotátorů shodne v anotaci některé položky slovníku (2-IAA). Protože autoři anotací mohli během hodnocení přeskočit některé dokumenty nebo mohli zpracovat různý počet dokumentů, výpočet byl proveden pouze se slovníky, které ohodnotili oba anotátoři. V tabulce 6.2 uvádíme makro průměr a směrodatnou odchylku hodnot naměřených pro slovníková hesla a jednotlivé druhy informací získané z anotací všech dvojic hodnotitelů.[3] Obecně platí, že shodu mezi anotátory je třeba brát s opatrností, zejména vzhledem k výše uvedeným důvodům, kvůli kterým některé položky nebyly započítány.

| **Položka** | **Vše**[a] | **Správnost** | **Užitečnost** |
|---|---|---|---|
| Slovníková hesla | 32.60±25.75 | 79.52±18.46 | 38.09±23.51 |
| Informace | 38.05±15.11 | 70.91± 9.50 | 47.95±17.61 |
| — definice | 35.97±17.46 | 67.96±12.69 | 47.28±18.95 |
| — překlady | 46.87±24.86 | 80.71± 9.00 | 54.12±28.69 |
| — příklady | 32.00±12.41 | 64.87± 7.97 | 42.96±18.46 |

*Pozn:* [a] Shoda ve správnosti i užitečnosti.

Tabulka 6.2: Shoda mezi anotátory při anotaci položek slovníku. (Všechny hodnoty jsou uváděny v procentech.)

Vhledem k nastavení experimentu lze předpokládat, že shoda v případě správnosti bude poměrně vysoká. Naopak v případě užitečnosti můžeme očekávat, že shoda bude výrazně nižší, protože různí překladatelé mají různé znalosti a pomáhají jim různé druhy informací. Tento předpoklad částečně potvrzují i naměřené hodnoty, zejména v případě užitečnosti. Poměrně nízká shoda v hodnocení přesnosti je však zajímavá. Tyto hodnoty lze připisovat tomu, jak složité je pro autory anotací porovnání významu informací obsažených ve slovníku se vstupním textem. Příčin může být mnoho, například různé vnímání významu jednotlivých slov (viz sekce 2.4.4), problémy s porozuměním vstupnímu textu a informacím ve slovníku, či nedostatečné pochopení řešené úlohy. Poslední možnost podporuje i

[3]Hodnoty naměřené pro jednotlivé dvojice a jiné podrobné statistiky lze získat pomocí programu anotačního prostředí, viz příloha C.

nízká shoda v hodnocení správnosti slovníkových hesel, jelikož tato podúloha je výrazně jednodušší v porovnání s hodnocením informací.

Druhou metodou použitou k vyhodnocení shody mezi anotátory je Cohenova kappa (Cohen, 1960). Na rozdíl od předchozí metody Cohenova kappa bere v úvahu, že shoda mezi hodnotiteli může vzniknout náhodně. Podobně jako v předchozím případě jsme kappu vypočetli pro všechny dvojice anotátorů s tím, že byly započítány pouze slovníky, které hodnotili oba anotátoři. V tabulce 6.3 jsou uvedeny průměrné hodnoty a směrodatné odchylky pro jednotlivé druhy položek ve slovníku.

| **Položka** | **Vše** | **Správnost** | **Užitečnost** |
|---|---|---|---|
| Slovníková hesla | 0,11±0,17 | 0,01±0,07 | 0,12±0,17 |
| Informace | 0,12±0,09 | 0,25±0,10 | 0,12±0,16 |
| — definice | 0,15±0,13 | 0,31±0,11 | 0,15±0,18 |
| — překlady | 0,15±0,09 | 0,36±0,12 | 0,18±0,17 |
| — příklady | 0,08±0,08 | 0,13±0,13 | 0,07±0,13 |

Tabulka 6.3: Průměrné hodnoty Cohenovy kappy vypočtené pro všechny dvojice hodnotitelů se směrodatnými odchylkami.

Kappa byla vypočtena jako:

$$\kappa = \frac{P_a - P_e}{1 - P_e}$$

kde $P_a$ je relativní pozorovaná shoda mezi hodnotiteli, tedy hodnota odpovídající naší první metodě pro vyhodnocení shody mezi anotátory, a $P_e$ je hypotetická pravděpodobnost náhodné shody. Pokud je mezi hodnotiteli absolutní shoda, pak $\kappa = 1$, pokud existuje pouze náhodná shoda, je $\kappa \leq 0$.

Celkově nejlepší shody podle obou metod bylo dosaženo při hodnocení nabízených překladů. Z hlediska dalších hodnocení systému založených na anotacích je však důležitá zejména shoda v označování správnosti položek ve slovních, protože ovlivňuje spolehlivost vyhodnocení přesnosti systému.

## 6.5 Vyhodnocení přesnosti

Jelikož není možné dostatečně přesně porovnávat význam různých tvrzení strojově, není možné založit hodnocení přesnosti na předkládaných informací na automatickém porovnávání s nějakým referenčním řešením. Proto bylo nutné vyhodnocení přesnosti systému založit na ručních anotacích, nebo jinak definovat danou úlohu, aby hodnocení dokázalo zachytit všechna možná řešení. Jak naznačuje vyhodnocení shody mezi anotátory, ani hodnocení přesnosti pomocí ruční anotace není jednoduché a tedy ani spolehlivé.

Přesnost systému je vypočtena jako podíl kladných ohodnocení správnosti od všech anotátorů v celkovém množství hodnocených položek. Při vyhodnocení nijak nerozlišujeme odpovědi od různých hodnotitelů. Tabulka 6.4 obsahuje

naměřené hodnoty pro všechny položky slovníků. V tabulce jsou navíc uvedeny hodnoty pro slovníková hesla tvořená pojmenovanými entitami, protože mají z pohledu překladatele jiný význam a systém s nimi pracuje jiným způsobem.

| **Položka** | **Správně** | **Špatně** | **Neoznačeno** |
|---|---|---|---|
| Slovníková hesla | 91,35 | 4,86 | 3,78 |
| Informace | 80,25 | 12,10 | 7,64 |
| — definice | 75,50 | 15,54 | 8,96 |
| — překlady | 85,25 | 9,22 | 5,53 |
| — příklady | 80,56 | 11,21 | 8,22 |

(a) Všechna hesla.

| **Položka** | **Správně** | **Špatně** | **Neoznačeno** |
|---|---|---|---|
| Slovníková hesla | 93,28 | 4,48 | 2,24 |
| Informace | 80,27 | 10,68 | 9,04 |
| — definice | 69,40 | 18,66 | 11,94 |
| — překlady | 90,00 | 4,55 | 5,45 |
| — příklady | 83,47 | 7,44 | 9,09 |

(b) Hesla tvořená pojmenovanými entitami.

Tabulka 6.4: Výsledky měření přesnosti systému. (Všechny hodnoty jsou uváděny v procentech.)

Jelikož správnost hesel, vzhledem k jejich výběru a instrukcím k provádění anotace, hodnotí spíše správnost segmentace a tokenizace, není její vysoká hodnota překvapivá. Hodnoty, které byly označovány jako špatné, tvoří většinou špatně rozpoznané pojmenované entity (například *Merkel &*), náhodné shluky znaků obsažené v tweetech (například *Th*). Spornou položkou, která byla označována jako špatná, jsou jména osob. Systém zařazuje do slovníku jak celá jména, tak i jednotlivé části zvlášť. Z našeho pohledu je to správné chování systému. Například pokud se v textu objeví jméno *Ignác Novák*, nemusí o něm systém najít žádné informace, ale informace, že *Novák* je příjmení používané v České republice, na Slovensku a v Maďarsku, pro cizince užitečná být může. Nicméně by bylo lepší, aby systém vybíral slovníková hesla volněji jen v případě, že nenajde žádné informace o heslech přesnějších.

Naopak hodnoty naměřené pro předkládané informace překvapivé jsou. Lze očekávat, že výsledky měření pro naší úlohu budou podobné hodnocení přesnosti metody pro desambiguaci, kterou je naše řešení inspirováno (Fanta a kol., 2015). A to zejména v případě výběru definic, kdy jsou používány téměř totožné zdroje informací. Tato metoda však dosáhla v rámci oficiálního hodnocení v rámci konference SemEval přesnosti pouze 41,2% (Moro a Navigli, 2015). Příčin může být několik: například může být algoritmus pro tento typ úloh vhodnější, naše úpravy tohoto algoritmu přinesly zlepšení, nebo je to důsledek nepřesnosti ručního hodnocení.

Vyzdvihněme však, podobně jako v případě shody mezi anotátory, vysokou přesnost překladů, jelikož výběr dat z paralelních korpusů pomocí použitého algo-

ritmu se zdál v prvotních experimentech velmi nepřesný. Příčinou tohoto úspěchu však muže být například malá víceznačnost slovníkových hesel v testovacím vzorku a podobně.

## 6.6 Vyhodnocení užitečnosti

V této sekci se pokusíme vyhodnotit spokojenost hodnotitelů se slovníky, které systém vytváří. Pokusíme se určit, zda překladatelé v těchto slovnících nějaký druh informací preferují a zda je tato preference jednoznačná.

Aby bylo možné porovnávat hodnoty naměřené pro jednotlivé anotátory, bylo nejprve nutné počty jednotlivých odpovědí převést na procentuální reprezentaci, protože každý anotátor ohodnotil různý počet slovníků, které jsou různě obsáhlé. Následně jsme hodnoty získané pro jednotlivé anotátory zprůměrovali. Vypočtené hodnoty makro průměrů a směrodatné odchylky jsou uvedené v tabulce 6.5. Díky tomuto postupu je možné pozorovat velké rozdíly mezi požadavky jednotlivých hodnotitelů, které ukazují vypočtené směrodatné odchylky.

Pouze u 39% slovníkových hesel si hodnotitelé myslí, že existuje nějaká informace, která by jim pomohla s jejich překladem, či s porozuměním jejich významu. Lze tedy předpokládat, že ostatní hesla jsou pro ně dostatečně známá. I přesto hodnotitelé považují průměrně 47% informací o slovníkových heslech za užitečné, zejména pak uvedené překlady, kterých je průměrně 58% užitečných.

Pokud sestavíme pořadí jednotlivých druhů informací na základě hodnot z tabulky 6.5, nejužitečnější se zdají překlady následovávané definicemi a příklady. Avšak pokud sestavíme stejné pořadí pro jednotlivé hodnotitele, někteří považovali za nejužitečnější druh informací definice. Žádný z hodnotitelů nepovažoval za nejužitečnější druh informace příklady, ale také nebyly vždy považovány za nejhorší zdroj informací. Pouze jeden hodnotitel považoval za nejméně užitečný druh informací o slovníkových heslech překlady. Proto nelze určit jednoznačně nejužitečnější druh informací.

## 6.7 Vyhodnocení požadovaných úseků

Další otázka, na kterou experiment měl odpovědět, je zda vytvořené slovníky skutečně obsahují hesla, která překladatelé vyžadují, a zda tato hesla obsahují správné informace.

Získávání požadovaných hodnot je založeno na porovnávání úseků vstupního textu, které uživatelé vybírali během první fáze anotace (viz sekce 6.3), a úseků tvořících slovníková hesla. Protože uživatelé mohli vybírat libovolné úseky tvořené tokeny, dokonce úseky skládající se z několika částí vstupního textu, rozhodli jsme se použít dvě metody porovnávání:

1) přesnou shodu, kdy heslo přesně odpovídá vybranému úseku, tedy je ve vstupním textu na stejné pozici a má stejnou délku,

2) částečnou shodu, kdy heslo a vybraný úsek má neprázdný průnik.

V tabulce 6.6 uvádíme poměr shodných úseků vůči všem vyznačeným úsekům, tedy množství skutečně vysvětlených úseků ve slovníku, a poměr shodných

| Položka | Užitečné | Neužitečné | Neozn. |
|---|---|---|---|
| Slovníková hesla | 38,92±14,59 | 44,45± 30,82 | 16,63±21,34 |
| Informace | 47,19±24,49 | 44,62± 31,38 | 8,19± 8,02 |
| — definice | 43,03±19,94 | 45,83± 31,41 | 11,14±12,02 |
| — překlady | 58,38±31,97 | 36,93± 33,66 | 4,69± 4,27 |
| — příklady | 41,20±26,56 | 50,42± 33,68 | 8,38± 8,20 |

(a) Všechna hesla

| Položka | Užitečné | Neužitečné | Neozn. |
|---|---|---|---|
| Slovníková hesla | 37,63±27,99 | 48,19± 29,88 | 14,18±17,96 |
| Informace | 40,79±24,62 | 50,91± 31,93 | 8,30± 9,63 |
| — definice | 37,52±10,65 | 49,82± 29,94 | 12,66±21,31 |
| — překlady | 50,66±37,89 | 46,73± 35,55 | 2,60± 4,21 |
| — příklady | 35,31±33,37 | 56,23± 37,53 | 8,46± 6,84 |

(b) Hesla tvořená pojmenovanými entitami

Tabulka 6.5: Výsledky měření užitečnosti. (Všechny hodnoty jsou uváděny v procentech.)

úseků vůči slovníkovým heslům, tedy počet hesel z předkládaných slovníků, které hodnotitelé explicitně vyžadovali.

Naměřené hodnoty ukazují, že ve většině případů slovníky opravdu obsahovaly nějaké vysvětlení pro úseky vstupního textu, které hodnotitelé vyžadovali. Zajímavý je v tomto případě rozdíl mezi hodnotami pro přesnou a částečnou shodu, protože většina slovníkových hesel dle přesnosti byla správně vybrána (viz sekce 6.5). To je způsobeno tím, že anotátoři vybírali například víceslovné úseky, nebo úseky obsahující speciální znaky, které systém produkuje pouze v případě pojmenovaných entit, popřípadě při špatně rozpoznaném úseku textu, který tvoří pojmenovanou entitu. Nižší hodnota při částečné shodě pak vypovídá o špatné filtraci hesel, nebo neúspěšném vyhledání informací o požadovaném úseku.

Velmi nízký počet explicitně vyžadovaných hesel jasně ukazuje, že použitý způsob filtrace je nedostatečný, spíše nevhodný. Příčinou může být složité hledání vhodné hranice pro zařazení slov, které není možné nijak automatizovat. Jinými možnostmi filtrace hesel se zabýváme v kapitole 7.

| | |
|---|---|
| Počet vysvětlených úseků | |
| — přesná shoda | 57,86 |
| — částečná shoda | 84,29 |
| Počet vyžadovaných hesel | |
| — přesná shoda | 14,59 |
| — částečná shoda | 21,26 |

Tabulka 6.6: Výsledky porovnávání požadovaných úseků a slovníkových hesel. (Všechny hodnoty jsou uváděny v procentech.)

V tabulce 6.7 uvádíme statistiky přesnosti a užitečnosti pro jednotlivé druhy

informací o slovníkových heslech provedené stejnými metodami jako v sekcích 6.5 a 6.6 s tím rozdílem, že uvažujeme pouze potvrzená slovníková hesla.

Naneštěstí se v tomto případě hodnoty přesnosti ani užitečnosti nezlepšily. Avšak hodnotitelé v tomto případě považují v průměru za užitečnější definice na rozdíl od původních statistik, kdy výrazně převládaly překlady.

| **Informace** | **Přesná shoda** | | | **Částečná shoda** | | |
|---|---|---|---|---|---|---|
| | Správně | Špatně | Neozn. | Správně | Špatně | Neozn. |
| Vše | 71,23 | 13,68 | 15,09 | 68,37 | 16,29 | 15,34 |
| Definice | 66,67 | 17,28 | 16,05 | 58,47 | 22,88 | 18,64 |
| Překlady | 81,36 | 6,78 | 11,86 | 80,68 | 9,09 | 10,23 |
| Příklady | 68,06 | 15,28 | 16,67 | 69,16 | 14,95 | 15,89 |

(a) Přenosnost.

| **Inform.** | **Přesná shoda** | | | **Částečná shoda** | | |
|---|---|---|---|---|---|---|
| | Užitečné | Neužit. | Neozn. | Užitečné | Neužit. | Neozn. |
| Vše | 62,4±18,4 | 23,6±23,3 | 13,9±14,1 | 51,1±27,4 | 26,6±24,4 | 22,1±31,5 |
| Definice | 66,5±25,0 | 22,3±22,4 | 11,1±15,7 | 47,7±24,7 | 24,0±25,1 | 28,2±34,3 |
| Překlady | 62,7±40,1 | 13,9±15,9 | 23,3±43,4 | 64,1±30,5 | 20,4±12,7 | 15,3±25,9 |
| Příklady | 60,6±33,2 | 32,8±34,3 | 6,5± 9,6 | 43,9±32,7 | 34,2±36,7 | 21,8±35,0 |

(b) Užitečnost.

| **Inform.** | **Přesná shoda** | | | **Částečná shoda** | | |
|---|---|---|---|---|---|---|
| | Užitečné | Neužit. | Neozn. | Užitečné | Neužit. | Neozn. |
| Vše | 88,7±12,0 | 10,5±12,7 | 0,7±1,4 | 72,0±23,6 | 12,8±17,0 | 15,0±28,0 |
| Definice | 89,5± 9,3 | 8,1±10,7 | 2,2±4,5 | 72,1±27,9 | 9,7±13,9 | 18,0±32,5 |
| Překlady | 92,3±13,3 | 7,6±13,3 | 0,0±0,0 | 91,9±10,9 | 8,0±10,9 | 0,0± 0,0 |
| Příklady | 81,0±26,9 | 18,9±26,9 | 0,0±0,0 | 60,6±33,1 | 21,6±32,5 | 17,8±32,6 |

(c) Užitečnost pouze správných informací.

Tabulka 6.7: Přenosnost systému při výběru informací na požadovaných úsecích a užitečnost těchto informací z pohledu překladatelů. (Všechny hodnoty jsou uváděny v procentech.)

Navíc v tabulce 6.7 uvádíme i statistiku užitečnosti informací pouze pro vyžadovaná hesla, které byla navíc vybrány jako správné. Tato statistika jasně říká, že pokud by systém dokázal vybírat hesla, o kterých uživatelé vyžadují nějaké doplňující informace, a pokud budou tyto informace správné, budou uživatelé systém považovat za užitečný. Tato skutečnost ukazuje, že systémy zaměňující se na sestavování minimálního kontextově závislého slovníku, mají šanci uspět.

## 6.8 Doba výpočtu

Doba sestavování slovníku závisí na mnoha faktorech, zejména na podobě vstupního textu. Vstupní text ovlivní dobu výpočtu jednak počtem slov, jednak

počtem pojmenovaných entit, které tvoří kandidáty na hesla slovníku. Díky tomu dobu výpočtu ovlivňují různá nastavení výběru a filtrování kandidátů. Dobu výpočtu samozřejmě ovlivňuje i velikost použitých zdrojů a nastavení jednotlivých komponent třetích stran.

Pro bližší představu uveďme, že průměrná doba sestavení slovníku pro jeden text z testovacích dat trvala průměrně méně než jednu vteřinu v testovacím prostředí, kde každá softwarová komponenta byla nasazena na samostatném fyzickém uzlu. Jelikož sestavování slovníku obsahuje několik sítových volání a jiných časově náročných operací a jednotlivé komponenty nebyly optimalizované pro dané prostředí, je tento údaj čistě orientační.

# Kapitola 7

# Další vývoj

Jak již bylo naznačeno v úvodu, neexistuje mnoho prací, které se zabývají podobným problémem jako tato, proto bychom rádi naznačili další možnosti, či výzvy, které jistě stojí za prozkoumání.

## 7.1 Jiné metody vyhodnocování

Stěžejní pro další vývoj podobného systému je lepší způsob vyhodnocování než námi navrhovaný (viz kapitola 6). Ten je vhodný jen pro vyhodnocení jednoho systému a jeho výsledky nelze porovnávat mezi různými systémy, bylo by vhodné navrhnout jinou metodu odstraňující tento nedostatek.

Tato metoda musí uvažovat mnoho hledisek daného problému, například složitost porovnávání předkládaných informací, různý výběr hesel slovníku různými systémy, různé automatické adaptace systému (viz sekce 7.3) a podobně. Díky těmto problémům, které pravděpodobně není možné řešit automaticky, musí nový způsob vyhodnocování opět alespoň částečně využívat lidskou anotaci a musí uvažovat slovník jako celek.

Pokud bude cílem vyhodnocení pouze porovnávání různých systémů, může být toto porovnávání založeno například na seřazení výstupů těchto systémů podle nějakého kriteria. Tímto způsobem je vyhodnocován například strojový překlad (Bojar a kol., 2011).

## 7.2 Získávání informací z internetu

Slovníky obecně mají problém s aktuálností a rozsahem (viz sekce 2.2). Stejným problémem je zatíženo i námi navrhované řešení, jelikož využívá podobné uzavřené zdroje informací (viz kapitola 3). Nešel by tento nedostatek nějak odstranit?

Díky tomu, že se internet neustále vyvíjí, nabízí se možnost nějakým způsobem získat informace do sestavovaného slovníku z obsahu internetu. Pokud by něco takového bylo možné, jistě by to pomohlo zmírnit výše uvedené nedostatky, protože lepší zdroj informací v současné době neexistuje. Naneštěstí tato myšlenka přináší mnoho nových problémů, či úloh, které je nutné vyřešit.

Prvním krokem je jistě vyhledání vhodných zdrojů pro nějakého kandidáta na heslo ve slovníku, například webovou stránku, která obsahuje informace o

kandidátovi ve stejném smyslu, který vyjadřuje v překládaném textu. K řešení tohoto podproblému mohou být využity například velké internetové vyhledávače v kombinaci se zjednoznačněním významu slov a pojmenovaných entit založeném na vyhledávání informací (viz sekce 2.4.3). Bohužel tento přístup k problému WSD a EL nedosahuje dobrých výsledků (Moro a Navigli, 2015), a proto je nutné najít jiné řešení, nebo ho nějakým způsobem vylepšit.

Dalším krokem je výběr nejlepších relevantních informací z nalezených zdrojů, které budou užitečné pro překladatele, a jejich zhuštění (zejména v případě definic), aby vytvářené slovníky vyhovovaly naší definici minimálního kontextově závislého slovníku. K tomuto zhuštění mohou být využity například algoritmy pro automatickou sumarizaci textu (Das a Martins, 2007), pokud zdroje nebude lépe strukturované pro strojové zpracovaní, například mohou být použita různá schemata přidávající sémantiku HTML.[1]

## 7.3 Adaptace na konkrétního překladatele

Pokud chceme překladatelům předkládat pouze informace, které potřebují k překladu, je jistě výhodné adaptovat systém pro jednotlivé uživatele. Adaptace systému pro uživatele se může skládat například z pouhého nastavení parametrů systému. Například možnost výběru druhů informací, které chce uživatel dostávat, či statické nastavení filtrování kandidátů na hesla slovníku (viz sekce 4.2) pro konkrétního uživatele jistě zlepší jeho spokojenost při práci se systémem, avšak automatická adaptace nabízí mnohem zajímavější možnosti.

Takový druh adaptace by měl například určovat, která slova nebo pojmenované entity asi překladatel již zná, na základě dříve přeložených textů. Dále systém může podobným způsobem vybírat druh informací předkládaných uživateli, ukáže-li se, že například definice si uživatel zapamatuje okamžitě, zatímco překlady vyžaduje neustále. Důležitým faktorem tohoto druhu adaptace je i zapomínání, jelikož nelze předpokládat, že si uživatelé budou pamatovat dříve předložené informace navždy.

Velkým problémem při návrhu takové adaptace je interakce s uživatelem. Pokud budeme předpokládat začlenění vytváření slovníku do prostředí, které nabízí systém Twitter Crowd Translation(viz sekce 2.1), bude velmi složité systém optimalizovat pro jednotlivé překladatele, protože neexistuje žádný přímočarý způsob potvrzování užitečnosti slovníku přes e-mailové rozhraní.

## 7.4 Analýza koreferencí

Výrazným vylepšením našeho postupu může být analýza koreferencí (coreference resolution) (Elango, 2005; Sapena Masip a kol., 2008) ve vstupním textu. Analýza koreferencí může pomoci například k zlepšení připojování významu pojmenovaným entitám, jestliže systém dokáže určit, která zájmena se k nim vztahují, je možné využít při zjednoznačňovaní jejich významu více různých kontextů. V takovém případě předpokládáme, že zájmeno je možné nahradit řetězcem tvořícím pojmenovanou entitu a desambiguace by měla ve všech případech vybrat stejný význam, který bude nejlépe vyhovovat dané situaci. Avšak nejlepší systémy

[1] Například schema.org

analyzující koreference v anglických textech sice dosahují na určitých testovacích datech více než 75% F-míry (například Raghunathan a kol., 2010), ale v průměru mezi různými daty nedosahují ani 60% F-míry (Pradhan a kol., 2011). Z toho důvodu není zatím možné analýzu koreferencí plně využít.

Další možností využití koreferencí je vyhledání koreferencí mezi pojmenovanými entitami. To je vhodné zejména pří sestavování slovníků pro delší texty, kdy může jeden význam vázat na více pojmenovaných entit v textu a je tedy vhodné tyto pojmenované entity spojit pod jedno slovníkové heslo. Sice by tuto situaci mělo řešit samotné připojování významu pojmenovaným entitám, nicméně použití i jiných metod může zlepšit výsledky a zmírnit problémy popsané v sekci 4.5.

## 7.5 Vliv slovníku na kvalitu překladu

Dalším zajímavým pohledem na používání minimálního kontextově závislého slovníku je jeho vliv na překlady, které s jeho použitím vznikají. Protože podobné systémy v podstatě neexistují (viz sekce 2.5), není zřejmé jakým způsobem budou překladatelé slovník využívat. Na rozdíl od klasický slovníků, jenž nabízí vždy více alternativních významů či překladů, mezi nimiž mohou překladatelé vybírat, předkládá náš systém nejvýše jednu možnost, o které tvrdí, že je správná. Pokud budou překladatelé vždy používat překlady poskytované systémem, muže docházet k degradaci překladu, nebo jazyka jako takového. Podobným způsobem mohou kvalitu překladu ovlivňovat i současné CAT nástroje, avšak ty z pravidla používají informace, které zadávají sami překladatelé, proto vliv na překlad je menší. Pokud ale bude slovník používán zejména k porozumění textu, tento problém se nemusí vůbec projevit.

Z toho důvodu by bylo vhodné tuto oblast lépe prozkoumat. Zajímavými údaji můžou být například vyhodnocení kvality překladu, či shody mezi překladateli (inter-translator agreement).

# Kapitola 8

# Závěr

V této práci jsme se zabývali automatickým sestavováním minimálních kontextově závislých slovníků pro překladatele. Nejprve jsme v práci definovali, co je minimální kontextově závislý slovník a pokusili jsme se ukázat přínos takových slovníků při řešení problémů, se kterými se překladatelé setkávají při překladu krátkých zpráv ze sociální sítě Twitter. Následně jsme úlohu zařadili do kontextu již existujících prací a analyzovali jsme ji z pohledu souvisejících úloh používaných při počítačovém zpracování přirozeného jazyka.

Na základě této analýzy jsme navrhli a implementovali systém, který takový typ slovníku pro překlad z angličtiny do češtiny zcela automaticky sestavuje. Informace pro sestavované slovníky jsou získávány z dostupných otevřených zdrojů pomocí metod pro zjednoznačňovaní významu slov a připojování významu pojmenovaným entitám. Těžiště našeho řešení spočívá ve výběru vhodných slovníkových hesel a ve výběru relevantních informací popisujících význam těchto hesel a v získávání správných překladových ekvivalentů.

Poté jsme navrhli experiment, který měl za cíl ověřit správnost našeho řešení a ověřit užitečnost vytvářených slovníků pro překladatele. Experiment byl založen na ručním hodnocení minimálních kontextově závislých slovníků sestavených pro krátké zprávy ze sociální sítě Twitter zaměřené na konkrétní doménu a vyznačování částí těchto textů, ke kterým hodnotitelé vyžadovali další informace.

V závěru práce jsme ukázali různé možnosti dalšího vývoje v oblasti sestavování minimálních kontextově závislých slovníků, jako je adaptace slovníku na konkrétního překladatele, řešení problému s omezenou velikostí použitých zdrojů, či další možnosti vylepšení našeho řešení dané úlohy.

Na základě získaných dat v rámci provedeného experimentu jsme se pokusili vyhodnotit přesnost našeho řešení, dále jsme se pokusili určit jaký druh informací překladatelům pomáhá a jakou část vytvářených slovníků překladatelé skutečně potřebují.

Jako první jsme vyhodnotili shodu mezi jednotlivými hodnotiteli. K tomu jsme použili jednak jednoduchou procentuální metodu, jednak Cohenovu kappu. Na základě tohoto hodnocení můžeme konstatovat, že i samotná úloha, kterou hodnotitelé řešili během hodnocení je velmi složitá. Tento fakt je založen na vyhodnocení shody mezi jednotlivými hodnotiteli v položkách, pro které byla nastavena velmi striktní pravidla hodnocení.

Anotátoři se v hodnocení správnosti jednotlivých položek slovníku shodli průměrně v 80% v případě hodnocení slovníkových hesel a v 71% v případě všech

druhů informací, přesněji v 68% hodnocení definic, 81% hodnocení překladů a 65% hodnocení příkladů. Přesnost systému na základě anotací je poměrně vysoká. Pro jednotlivé druhy informací jsou výsledky následující: 76% v případě definic, 85% v případě překladů a 80% u příkladů. Souhrnně pro všechny druhy informací je přesnost systému 80%. Přesnost výběru slovníkových hesel, která se řídila jinými pravidly, je 91%.

Hodnocení užitečnosti pouze dokázalo, že požadavky jednotných překladatelů jsou velmi odlišné. Toto tvrzení potvrdila jak malá shoda v anotacích užitečnosti, tak samotné naměřené hodnoty. Shoda hodnotitelů pro užitečnost slovníkových hesel je pouze 38% a 48% pro všechny druhy informací. Největší průměrná shoda v hodnocení užitečnosti byla v případě překladů, a to 54%. Můžeme říci, že překlady byly průměrně hodnoceny nejlépe, nicméně rozdíly mezi jednotlivými druhy jsou malé a někteří hodnotitelé považovali za nejužitečnější definice. Nemůžeme tedy říci, který druh informací byl pro překladatele nejužitečnější, tomu nasvědčují i vysoké odchylky hodnocení jednotlivých účastníků experimentu.

Posledním cílem experimentu bylo vyhodnocení, zda systém správně vybírá slovníková hesla. Ukázalo se, že uživatelé požadovali pouze 15%, popřípadě 21%, ze všech hesel vedených v hodnocených slovnících v závislosti za způsobu porovnávání vybraných slov překladateli s hesly ve slovníku. Avšak slovníky obsahovali nějaké vysvětlení k 58% (či 84%, podle metody porovnání) úseků požadovaných uživateli. To jasně ukazuje, že námi použitá metoda filtrace je nedostatečná, nicméně není alespoň úplně špatná.

Pokud se však zaměříme pouze na užitečnost informací, které jsou obsaženy v heslech shodných s požadovanými úseky a které jsou zároveň označeny jako správné, výsledky jsou velmi slibné. V závislosti na metodě porovnávání je $89\pm12\%$, popřípadě $73\pm24\%$ informací užitečných, navíc v obou případech překlady mají užitečnost přes 90% s poměrně malou odchylkou. To ukazuje, že přesný systém s dobrým způsobem filtrování slovníkových hesel bude překladatelům opravdu užitečný.

Na závěr musíme říci, že daný problém je v mnoha ohledech velmi složitý a i jednotné dílčí úlohy, které jsou aktivně zkoumány, zatím nedosahují vynikajících výsledků a přináší mnoho nevyřešených problémů, proto nelze takové výsledky očekávat ani v případě automatického sestavování kontextově závislých slovníků.

# Seznam použité literatury


Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. a Ives, Z. (2007). *Dbpedia: A nucleus for a web of open data*. Springer.

Banerjee, S. a Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational linguistics and intelligent text processing*, pages 136–145. Springer.

Bojar, O., Ercegovčević, M., Popel, M. a Zaidan, O. F. (2011). A grain of salt for the wmt manual evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11. Association for Computational Linguistics.

Bojar, O., Žabokrtský, Z., Dušek, O., Galuščáková, P., Majliš, M., Mareček, D., Maršík, J., Novák, M., Popel, M. a Tamchyna, A. (2012). The Joy of Parallelism with CzEng 1.0. In *Proceedings of LREC2012*, Istanbul, Turkey, May 2012. ELRA, European Language Resources Association. In print.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20**(1), 37–46.

Das, D. a Martins, A. F. (2007). A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, **4**, 192–195.

Elango, P. (2005). Coreference resolution: A survey. *University of Wisconsin, Madison, WI*.

Fanta, P., Sudarikov, R. a Bojar, O. (2015). TeamUFAL: WSD+EL as document retrieval. In Nakov, P., Zesch, T., Cer, D. a Jurgens, D., editors, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 350–354, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics. ISBN 978-1-941643-40-2.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Guo, S., Chang, M.-W. a Kiciman, E. (2013). To link or not to link? a study on end-to-end tweet entity linking. In *HLT-NAACL*, pages 1020–1030.

Hajič, J., Votrubec, J., Krbec, P., Květoň, P. a kol. (2007). The best of two worlds: Cooperation of statistical and rule-based taggers for czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, pages 67–74. Association for Computational Linguistics.

Kilgarriff, A. (1997). I don't believe in word senses. *Computers and the Humanities*, **31**(2), 91–113.

Kilgarriff, A. a Rosenzweig, J. (2000). Framework and results for english senseval. *Computers and the Humanities*, **34**(1-2), 15–48.

Kubát, P. (2014). Normalizace pojmenovaných entit v českých textech. Bachelor's thesis, Matematicko-fyzikální fakulta, Univerzita Karlova.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM.

Mihalcea, R. a Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, **38**(11), 39–41.

Moro, A. a Navigli, R. (2015). SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *Proc. of SemEval-2015*.

Moro, A., Cecconi, F. a Navigli, R. (2014a). Multilingual Word Sense Disambiguation and Entity Linking for Everybody. In *Proceedings of the 13th International Semantic Web Conference, Posters and Demonstrations (ISWC 2014)*, pages 25–28, Riva del Garda, Italy, 2014a.

Moro, A., Raganato, A. a Navigli, R. (2014b). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, **2**, 231–244.

Nadeau, D. a Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, **30**(1), 3–26.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, **41**(2), 10.

Navigli, R. a Ponzetto, S. P. (2010). Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics.

Nebhi, K., Nerima, L. a Wehrli, E. (2013). Nerits-a machine translation mashup system using wikimeta and dbpedia. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 312–318. Springer.

Och, F. J. a Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 1086–1090. Association for Computational Linguistics.

Ponzetto, S. P. a Navigli, R. (2010). Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1522–1531. Association for Computational Linguistics.

Popel, M. a Žabokrtský, Z. (2010). Tectomt: modular nlp framework. In *Advances in natural language processing*, pages 293–304. Springer.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, **14**(3), 130–137.

Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R. a Xue, N. (2011). Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27. Association for Computational Linguistics.

Procházka, A. E. J. H. J. a Smrž, O. Entity recognition based on the co-occurrence graph and entity probability.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D. a Manning, C. (2010). A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501. Association for Computational Linguistics.

Sapena Masip, E., Padró, L. a Turmo Borras, J. (2008). Coreference resolution survey.

Ševčíková, M., Žabokrtský, Z. a Krůza, O. (2007). *Zpracování pojmenovaných entit v českých textech*. Universitas Carolina Pragensis.

Shen, H., Bunescu, R. a Mihalcea, R. (2013). Coarse to fine grained sense disambiguation in wikipedia.

Snyder, B. a Palmer, M. (2004). The english all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43.

Straková, J., Straka, M. a Hajič, J. (2013). A new state-of-the-art czech named entity recognizer. In *Text, Speech, and Dialogue*, pages 68–75. Springer.

Straková, J., Straka, M. a Hajič, J. (2014). Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P14/P14-5003.pdf`.

Šubert, E. a Bojar, O. (2014). Twitter Crowd Translation – Design and Objectives.

Sudarikov, R. a Bojar, O. (2015). Giving a sense: A pilot study in concept annotation from multiple resources. In Žabokrtský, Z., editor, *UFAL WDS 2015 (Conference of PhD Students in Mathematical Linguistics)*, pages 14–21, Praha, Czechia, 2015. Institute of Formal and Applied Linguistics, Charles University in Prague, Institute of Formal and Applied Linguistics, Charles University in Prague.

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L. a Trón, V. (2007). Parallel corpora for medium density languages. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, **292**, 247.

Vasilescu, F., Langlais, P. a Lapalme, G. (2004). Evaluating variants of the lesk approach for disambiguating words. In *LREC*.

Zhong, Z. a Ng, H. T. (2010). It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83. Association for Computational Linguistics.

# Dodatek A

# Příklady slovníků

Tato kapitola obsahuje ukázky slovníků vygenerovaných naším systémem (viz kapitola 4) pro dvě zprávy ze sociální sítě Twitter, které byly náhodně vybrány z dat, jenž sloužila k vyhodnocení systému (viz sekce 6.1).

| **Vstup** | *Today's IMF decision proves the civilized world believes in, and supports Ukraine. #UnitedForUkraine #MBΦ #IMF http://t.co/t4SE3XUX3K* |
|---|---|
| Today | **Def** the present time or age; Synonyms: today<br>**Tr** dnes<br>**Ex** Today proved that. |
| IMF | **Def** a United Nations agency to promote trade by increasing the exchange stability of the major currencies; Synonyms: International Monetary Fund, IMF<br>**Tr** MFF<br>**Ex** So what should world leaders do with the IMF? |
| proves | **Def** obtain probate of; Synonyms: prove<br>**Tr** dokázat<br>**Ex** Today proves that it doesn't work. |
| civilized | **Def** having a high state of culture and development both social and technological; Synonyms: civilized, civilised<br>**Tr** civilizovaný<br>**Ex** You are the best soldiers in the civilized world. |

| | |
|---|---|
| Ukaine | **Def** Ukraine is a country in Eastern Europe. It has an area of 603,628 km2 (233,062 sq mi), making it the largest country entirely within Europe. Ukraine borders ussia to the east and northeast, Belarus to the northwest, Poland, Slovakia and Hungary to the west, Romania and Moldova to the southwest, and the Black Sea and Sea of Azov to the south and southeast, respectively.<br>**Tr** ukrajina<br>**Ex** In Ukraine today, these are not abstract questions. |

| | |
|---|---|
| **Vstup** | *In my telephone conversation with Merkel, Hollande and Putin I stressed that the situation around Debaltseve was in breach of Minsk accords.* |
| telephone | **Def** electronic equipment that converts sound into electrical signals that can be transmitted over distances and then converts received signals back into sounds; Synonyms: telephone, phone, telephone set<br>**Tr** telefonní<br>**Ex** I will telephone my family. |
| conversation | **Def** the use of speech for informal exchange of views or ideas or information etc.; Synonyms: conversation<br>**Tr** rozhovor<br>**Ex** I recounted my conversation with her. |
| Merkel | **Def** Angela Merkel is the first female Chancellor of Germany, in office since 2005.<br>**Tr** Merkelová"<br>**Ex** But, as I said, back then Merkel did not dare to act. |
| Hollande | **Def** François Gérard Georges Nicolas Hollande (born 12 August 1954) is a French politician who has been the President of France since 2012. |
| Putin | **Def** Russian statesman chosen as president of the Russian Federation in 2000; formerly director of the Federal Security Bureau (born in 1952); Synonyms: Putin, Vladimir Putin, Vladimir Vladimirovich Putin<br>**Tr** Putin<br>**Ex** Putin acts according to his convictions. |

| | |
|---|---|
| stressed | **Def** test the limits of; Synonyms: try, strain, stress<br>**Tr** zdůraznit<br>**Ex** I am not stressed. |
| breach | **Def** an opening (especially a gap in a dike or fortification); Synonyms: breach<br>**Tr** porušení<br>**Ex** My loss counts as breach. |
| Minsk | **Def** Minsk is a doom metal/post-metal band from Peoria, Illinois founded in 2002. Self described as "psychedelic metal", their songs tend to start out as slow and simple, and become heavy and complex towards the end. Their sound draws equally from sludge metal, doom metal, hardcore punk, ambient and noise music, with a highly psychedelic attitude achieved through tribal drum patterns, thick layers of synthesizers and keyboards, and echoing vocals. They are named after the capital of Belarus.<br>**Tr** Minsk<br>**Ex** Did you know I can buy nuclear warheads in Minsk for 40 million? |

# Dodatek B

# Uživatelská dokumentace hlavního programu

Systém pro sestavování minimálního anglicko-českého kontextově závislého slovníku, *CoDetionary*, je webová aplikace poskytující REST rozhraní. Hlavní část programu je naprogramovaná v jazyce *Kotlin 1.0*, který je překládán do Java bajtkódu a potřebuje ke spuštění *Java Virtual Machine (JVM)* ve verzi 6 nebo novější. Jelikož JVM je dostupná na mnoha zařízeních a operačních systémech, lze i tuto část systému považovat za multiplatformní.

Avšak systém využívá nativní knihovny třetích stran naprogramované v programovacím jazyce *C++*, které je nutné samostatně přeložit pro jednotlivé architektury. Jedná se o nástroje *MorphoDiTa* a *NameTag*. Oba nástroje jsou dostupné pro operační systémy Linux a Windows v 32-bitové i 64-bitové. Nicméně součástí této práce jsou pouze verze pro 64-bitové Linuxové systémy (viz příloha D).

Jako externí datová úložiště systém využívá relační databázi *MySQL* a dokumentovou databázi, či NoSQL databázi *MongoDB*.

## B.1 Příprava zdrojů

Jelikož množství použitých dat přesahuje možnosti příloh k této práci, věnujeme následující sekci detailnímu popisu přípravy dat pro systém. Cílem tohoto procesu je vložení upravených dat do databází a vytvoření fulltextových indexů nad těmito databázemi a jinými zdroji dat.

K tomuto účelu slouží skripty a nástroje přiložené v balíčku se systémem (adresář *codetionary-bin/tools* v elektronické příloze k této práci, viz příloha D). Vstupem pro tyto nástroje byly následující zdroje:

- paralelní korpus CzEng 1.0,[1]
- databázové soubory WordNeru ve verzi 3.1,[2]
- výpis článků české a anglické Wikipedie v rámci projektu Kiwix z května 2015,[3]

[1] http://ufal.mff.cuni.cz/czeng/czeng10/

[2] `https://wordnet.princeton.edu/wordnet/download/current-version/`

[3] `http://wiki.kiwix.org/wiki/Content_in_all_languages`

- výpis mezijazykových odkazu z databáze anglické Wikipedie z května 2015,[4].

Zpracování WordNetu je snadné, protože je distribuován ve formátu vhodném pro účely systému a stačí pro něj vytvořit fulltextový index. K tomuto účelu slouží skript *wordnet_indexer.sh*, který očekává jako parametr cestu k adresáři s databází WordNetu a který v aktuálním adresáři vytvoří adresář s indexem, *wordnet_index*. Jelikož je WordNet nejmenší použitý zdroj a je s ním prováděno málo operací, jeho zpracování proběhne na běžném počítači během několika minut.

V případě paralelního korpusu CzEng je situace složitější. Nejprve je nutné vložit data do databáze a následně pro ně vytvořit fulltextový index. K vložení dat do databáze slouží skript *czeng_inserter.sh*, který očekává tři poziční parametry:

1. cestu k balíčku dat CzEngu v prostém formátu,
2. cestu k odpovídajícímu balíčku dat CzEngu v export formátu,
3. přístupový řetězec k MongoDB databázi.

Tento skript musí být spuštěn pro všechny balíčky dat CzEngu, skript může být spouštěn paralelně. Kromě vložení dat do databáze skript provede všechny potřebné úpravy dat (viz sekce 4.3.1).

K vytvoření indexu pro databázi pak slouží skript *czeng_inserter.sh*, který jako parametr očekává přístupový řetězec k MongoDB databázi obsahující CzEng a který v aktuálním adresáři vytvoří adresář s indexem, *czeng_index*. Celková doba zpracování CzEngu je v řádu hodin až dnů v závisti na paralelizaci vkládání dat do databáze a použitém hardwaru.

Nejobsáhlejším zdrojem dat je Wikipedie. Její zpracování je otázkou dní až týdnů při použití dobrého hardwaru.[5] Při zpracování Wikipedie je potřeba provést mnoho kroků při extrakci dat vhodných pro účely systému. Jelikož zpracování oficiálních výpisů databáze Wikipedie je kvůli použitému značkovacímu jazyku, Wikitextu, velmi náročné, rozhodli jsme se využít alternativní zdroj těchto dat, projekt Kiwix, který obsahuje články Wikipedie již převedené do HTML.

Naneštěstí projekt Kiwix používá speciální formát souborů ZIM, určený pro offline používání Wikipedie. Proto prvním krokem při zpracování Wikipedie je převod ZIM souborů do jednoduchého textového formátu, se kterým umí pracovat ostatní skripty a nástroje. K tomuto účelu slouží skript *zim.sh*. Tento skript očekává jediný parametr obsahující cestu k *.zim* souboru a vypíše na výstup data v novém formátu. Skript používá jednoduchý nativní program (naprogramovaný v C++) *zimconverter*, který ke svému spuštění potřebuje sdílenou knihovnu *zimlib*.[6]

Dalším krokem je vložení článků převedených na prostý text do databází pro českou a anglickou Wikipedii. To lze provést pomocí skriptu *wiki_pages.sh*, který čte ze standardního vstupu převedený ZIM soubor a jako parametry očekává přístupové údaje k MySQL databázi učené pro danou jazykovou verzi, tedy přístupový řetězec, uživatelské jméno a heslo. Dále uvedené úkony se týkají pouze databáze obsahující anglickou Wikipedii.

[4]Soubory enwiki-20150403-langlinks.sql.gz a enwiki-20150403-page.sql.gz dostupné na adrese `http://dumps.wikimedia.org/enwiki/20150403`

[5]Vytváření indexu pro Wikipedii vyžaduje až 16 GB RAM.

[6]`http://www.openzim.org/wiki/Zimlib`

Pro ostatní kroky je nutné vytvořit dva pomocné mapovací soubory: první obsahuje mapování databázových identifikátorů na názvy stránek, druhý mapování stránek s přesměrováním, které mohou být tranzitivně závislé, na koncové stránky (články). Tyto soubory lze získat pomocí skriptů *wiki_db_id.sh* a *wiki_redirections.sh*, kde první očekává jako parametry přístupové údaje k databázi a vypíše na standardní výstup požadované mapování. Druhý pak čte ze standardního vstupu převedený ZIM soubor a vypisuje na standardní výstup mapování přesměrování na koncové stránky.

Posledním krokem nutným k vytvoření databází je vložení anglicko-českých mezijazyčných odkazů do databáze určené pro anglickou Wikipedii. Naneštěstí Kiwix projekt mezijazyčné odkazy neobsahuje, proto je nutné použít přímo výpisy databáze Wikipedie. Vložení lze provést pomocí skriptu *wiki_langlinks.sh*, který očekává následující parametry:

1. výpis mezijazyčných odkazů z databáze Wikipedie (například soubor *enwiki-20150403-langlinks.sql.gz*),
2. výpis stránek z databáze Wikipedie (například soubor *enwiki-20150403-page.sql.gz*),
3. kód cílového jazyka odkazů, pro získání anglicko-českých odkazů tedy *cs*,
4. mapování identifikátorů z databáze na názvy stránek,
5. mapování názvů stránek s přesměrováním na koncové stránky,
6. přístupové údaje k databázi.

Dále je nutné extrahovat údaje, které jsou ztraceny během převodu článků Wikipedie na prostý text, tedy sousední stránky a alternativní názvy jednotlivých stránek. K tomuto účelu slouží skripty *wiki_aliases.sh* a *wiki_neighbours.sh*. Oba tyto skripty čtou převedený ZIM soubor ze standardního vstupu a jako parametry očekávají výše zmíněné mapovací soubory. Na standardní výstup vypisují požadované seznamy, tedy pro každou koncovou stránku seznam stránek, které na ní odkazují a na které odkazuje daná stránka, a seznam aliasů získaných z kotev odkazů a názvů stránek s přesměrováním.

Posledním krokem před vytvořením fulltextového indexu pro anglickou Wikipedii, je vypsání všech koncových článků převedených na prostý text z databáze. To se ukázalo jako komfortnější řešení než dotazování do databáze během indexace. K tomu slouží skript *wiki_db_articles.sh*, jenž jako parametry očekává přístupové údaje k databázi a na výstup vypíše seznam článků převedených na prostý text.

K vytvoření indexu pro Wikipedii slouží skript *wiki_indexer.sh*, která jako parametry očekává výstupy předchozích kroků, tedy:

1. seznam článků Wikipedie bez formátování,
2. seznam sousedních článků pro každý článek,
3. seznam alternativních názvů pro každý článek.

Tímto je předpřipravení Wikipedie pro účely sytému hotové. Ačkoliv by jednotlivé skripty šly spojit do jednoho, přišlo nám toto řešení vzhledem k době výpočtu jednotlivých kroků nepraktické.

## B.2 Konfigurace

Systém je nutné před spuštěním správně nakonfigurovat. Jedná se zejména o nastavení přístupu k jednotlivým databázím a indexům. Bez tohoto nastavení není možné aplikaci spustit.

Pro konfiguraci systému slouží konfigurační soubor *CoDetionary.properties* umístěný v kořenovém adresáři aplikace. Dostupná nastavení jsou:

**server.host** Nastavení konkrétního sítového rozhraní, na kterém bude webový server poslouchat. Pokud klíč není nastaven, nebo hodnota je *0.0.0.0*, webový server poslouchá na daném portu na všech síťových rozhraních.

**server.port** Nastavení portu webového serveru. Výchozí hodnota: *9821*

**en.morphology.model** Nastavení cesty k jazykovému modelu pro nástroj MorphoDiTa. Výchozí hodnota: *models/english-morphium-wsj-140407.tagger*

**en.ner.model** Nastavení názvu modelu pro nástroj NameTag.[7] Výchozí hodnota: *english-conll-140408*

**en.wordnet.index** Nastavení cesty k indexu pro WordNet. Výchozí hodnota: *indexes/wordnet_index*

**en.wordnet.dictionary** Nastavení cesty k adresáři se soubory WordNetu. Výchozí hodnota: *dict*

**en.wiki.index** Nastavení cesty k indexu pro anglickou Wikipedii. Výchozí hodnota: *indexes/wiki_en_index*

**en.wiki.jdbc.url** Nastavení přístupového řetězce k databázi obsahující informace z anglické Wikipedie. Výchozí hodnota: *jdbc:mysql://localhost:3306/wiki_en?useUnicode=true*

**en.wiki.jdbc.username** Nastavení uživatelského jména pro přístup k databázi obsahující anglickou Wikipedii. Výchozí hodnota: *root*

**en.wiki.jdbc.password** Nastavení hesla pro přístup k databázi obsahující anglickou Wikipedii. Výchozí hodnota není nastavena.

**cs.wiki.jdbc.url** Nastavení přístupového řetězce k databázi obsahující informace z české Wikipedie. Výchozí hodnota: *jdbc:mysql://localhost:3306/wiki_cs?useUnicode=true*

**cs.wiki.jdbc.username** Nastavení uživatelského jména pro přístup k databázi obsahující českou Wikipedii. Výchozí hodnota: *root*

**cs.wiki.jdbc.password** Nastavení hesla pro přístup k databázi obsahující anglickou Wikipedii. Výchozí hodnota není nastavena.

[7] V současné verzi aplikace je používána webová služba umožňující značkování anglických textů, jelikož model pro angličtinu zatím není volně dostupný. Viz `http://ufal.mff.cuni.cz/nametag`.

**en_cs.czeng.index** Nastavení cesty k indexu pro paralelní korpus CzEng. Výchozí hodnota: *indexes/czeng_index*

**en_cs.czeng.mongo.url** Nastavení přístupového řetězce k databázi obsahující CzEng. Výchozí hodnota: *mongodb://localhost*

**en.probabilities** Nastavení cesty k modelu obsahujícímu pravděpodobnosti výskytu lemmat v anglickém jazyce. Výchozí hodnota: *models/en-probabilities_lemmas.gz*

**en.probabilities.threshold** Nastavení hranice pravděpodobnosti určující obecně známá lemmata. Výchozí hodnota: *0,999*

## B.3 Spuštění aplikace

Aplikace je distribuována jako balíček skládající se ze spustitelného JAR souboru, nativních knihoven a jazykových modelů. Pro spuštění aplikace je nutné vytvořit databáze obsahující potřebná data a fulltextové indexy pro všechny zdroje dat (viz sekce B.1) a aplikaci správně nakonfigurovat (viz sekce B.2).

Aplikaci je možné spustit z kořenového adresáře balíčku pomocí příkazu:

```
java -jar CoDeTionary-all.jar
```

Součástí balíčku je i skript, který umožňuje na Linuxových systémech spustit aplikaci na pozadí jako démon. Výhodou požití skriptu je nezávislost na aktuálním pracovním adresáři. Skript lze volat následujícím způsobem:

```
run.sh {start|stop|restart|status}
```

V obou případech je spuštěná aplikace v základním nastavení dostupná na portu *9821*, tedy například na URI: `http://localhost:9821`.

## B.4 Komunikace

Ke komunikaci s aplikací slouží jednoduché REST API. Toto API obsahuje dvě služby: *dictionary*, která vytváří kontextově závislý slovník podle vstupních parametrů, a *tokenize*, která vstupní text rozdělí na tokeny a je určena zejména pro snazší manipulaci s výsledným slovníkem. Služba *dictionary* je v základním nastavení dostupná na adrese *http://[adresa serveru]:9821/dictionary*. Obdobně je dostupná služba *tokenize* na *http://[adresa serveru]:9821/tokenize*.

Obě tyto služby lze volat pomocí dotazovacích metod GET i POST protokolu HTTP a přijímají tyto parametry:

**text** Vstupní text, pro který je vytvářen minimální kontextově závislý slovník. Parametr je povinný.

**source_lang** Kód jazyka vstupního textu podle normy ISO 639 alpha-2, případně alpha-3, pokud alpha-2 pro daný jazyk neexistuje. Parametr je povinný. V současné verzi programu je dostupný vstupní jazyk pouze angličtina (hodnota *en*).

**target_lang** Kód jazyka, do kterého bude překládáno. Pro tento parametr stejná omezení jako pro parametr *source_lang*. V současné verzi programu je dostupný cílový jazyk pouze čeština (hodnota *cs*).

**source** Nepovinný parametr, který umožňuje specifikovat původ textu. Může, ale nemusí, k němu být přihlédnuto při sestavování slovníku.

**pipeline** Jméno pipeline. Pomocí tohoto parametru lze vynutit zpracování požadavku konkrétní pipeline. Parametr je nepovinný.

Odpověď služeb je vždy ve formátu JSON. Pokud volání služby bylo úspěšné, odpověď vrací stavový kód HTTP protokolu 200 OK. V takovém případě tělo odpovědi obsahuje pole ve formátu JSON obsahující jednotlivá slovníková hesla a informace, které k nim náleží, popřípadě pole ve formátu JSON obsahující vstupní text rozdělený na věty a tokeny. Výpis B.1 obsahuje příklad úspěšného volání služby */dictionary* a výpis B.2 obsahuje příklad úspěšného volání služby */tokenize*.

V případě neúspěchu služby vrací odpovídající kód HTTP protokolu, tedy buď kód ze skupiny chyb klienta (4xx Client error), nebo kód ze skupiny chyb serveru (5xx Server error). Tělo odpovědi v takovém případě obsahuje chybový kód a popis chyby ve formátu JSON (viz výpis B.3).

```
POST /dictionary HTTP/1.1
Host: localhost:9821
Content-Type: application/x-www-form-urlencoded

source_lang=en&target_lang=cs&text=Obama is the 44th President of
↪  the United States.

HTTP/1.1 200 OK
Content-Type: application/json

{
  "entries" : [ {
    "text" : "Obama",
    "start" : 0,
    "length" : 5,
    "type" : "CANE",
    "fields" : [ {
      "value" : "Barack Hussein Obama II (born August 4, 1961) is
↪  an American politician and the 44th and current President of
↪  the United States.",
      "source" : "Wiki#Barack Obama",
      "type" : "D"
    }, {
      "value" : "Obama",
      "source" : "CzEng",
      "type" : "T"
    }, {
      "value" : "The last chapter is about the present, and
↪  president Obama.",
      "source" : "CzEng",
      "type" : "E"
    } ]
  },
  // ...
  ]
}
```

Výpis B.1: Ukázka volání služby */dictionary*. (Z dotazu i odpovědi byly vypouštěny některé HTTP hlavičky.)

```
POST /dictionary HTTP/1.1
Host: localhost:9821
Content-Type: application/x-www-form-urlencoded

source_lang=en&target_lang=cs&text=Obama is the 44th President of
 ↪  the United States.

HTTP/1.1 200 OK
Content-Type: application/json

[ [ {
  "form" : "Obama",
  "start" : 0,
  "length" : 5
},
// ...
] ]
```

Výpis B.2: Ukázka volání služby */tokenize*. (Z dotazu i odpovědi byly vypouštěny některé HTTP hlavičky.)

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Content-Length: 68

{
  "status" : 400,
  "message" : "Null or empty argument 'text'."
}
```

Výpis B.3: Ukázka odpovědi v případě chyby. (Z příkladu byly vypouštěny některé HTTP hlavičky.)

# Dodatek C

# Uživatelská dokumentace anotačního prostředí

Anotační prostředí je jednoduchá webová aplikace naprogramovaná v programovacích jazycích *Java 8* a *Kotlin 1.0* s pomocí *Spring* frameworku. Oba tyto jazyky jsou překládány do Java bajtkódu, a proto je možné aplikaci spustit v systémech, pro které existuje implementace *Java Virtual Machine* ve verzi 8 a novější. Lze tedy aplikaci považovat na multiplatformní.

Jako datové úložiště program používá relační databázi *MySQL*. Databázový systém lze však snadno pomocí vhodné konfigurace změnit.

## C.1 Spuštění aplikace a přístup k aplikaci

Aplikace je distribuována jako jeden spustitelný JAR soubor, který je možné spustit příkazem:

```
java -jar CoDetionaryEvaluationTool.jar
```

Aby spuštění aplikace mohlo být úspěšné, je nutné aplikaci správně nakonfigurovat. Možnosti konfigurace jsou blíže popsány v sekci C.2.

V základním nastavení je webové rozhraní aplikace dostupné na portu 8080, tedy na příklad na adrese `http://localhost:8080`. Ve webovém rozhraní jsou dostupné následující cesty:

| | |
|---|---|
| **/** | Anotační prostředí, viz sekce 6.2. |
| **/statistics** | Webová stránka obsahující podrobné statistiky získané z anotací provedených v anotačním prostředí. Na rozdíl od statistik uvedených v kapitole 6 stránka obsahuje i hodnoty naměřené pro jednotlivé hodnotitele a podobně. |
| **/documents-info** | Webová stránka obsahující základní informace o vstupních textech, ke kterým již byly v systému vytvořeny slovníky. |
| **/documents-info/NE** | Webová stránka obsahující výstup základní analýzy koreferencí mezi pojmenovanými entitami v již zpracovaných textech (viz sekce 6.1). |

## C.2 Konfigurace

Konfiguraci aplikace lze provést pomocí konfiguračního souboru *application.properties* umístěného v pracovním adresáři. V souboru je možné nastavit následující klíče:

**server.port** Nastavení portu webového serveru. Výchozí hodnota: *8080*

**spring.datasource.url** Nastavení přístupového řetěze pro databázi. Výchozí hodnota: *jdbc:mysql://localhost:3306/codetionary_evaluation?useUnicode=true &characterEncoding=UTF-8&useSSL=false*

**spring.datasource.username** Nastavení uživatelského jména pro přístup k databázi. Výchozí hodnota: *root*

**spring.datasource.password** Nastavení hesla pro přístup k databázi. Výchozí hodnota není nastavena.

**service.dictionary.uri** Nastavení URI služby pro vytváření slovníků. Služba je používána pouze v případě, že v aplikaci existuje vstupní text, který zatím nebyl zpracován. Výchozí hodnota: *http://localhost:9821/dictionary*

**http://localhost:9821/tokenize** Nastavení URI služby, která poskytuje tokenizaci vstupního textu. Služba je používána pouze v případě, že v aplikaci existuje vstupní text, který zatím nebyl zpracován. Výchozí hodnota: *http://localhost:9821/tokenize*

## C.3 Přednastavené anotační prostředí

Nasazení anotačního prostření nastaveného tak, jak bylo použito během experimentu prezentovanému v kapitole 6, je možné pomocí konfiguračních souborů pro virtualizační nástroj *Docker*.[1] Součástí vytvářených obrazů jsou i data získaná během experimentu, což umožňuje procházení jednotlivých statistik pomocí anotační aplikace.

K vytvoření virtuálních kontejnerů je vhodné použít rozšíření Dockeru *docker-compose*, které umožňuje vytvoření kompletní infrastruktury složené z Docker kontejnerů. Konfigurační soubor pro *docker-compose* se nachází v elektronické příloze (viz příloha D) v adresáři *codetionaryevaluationtool-data*. Pro vytvoření a spuštění kontejnerů obsahujících aplikaci a databázi lze z uvedeného adresáře použít příkaz:[2]

```
docker-compose up
```

Pokud vytvoření a spuštění obou kontejnerů proběhne v pořádku, je anotační prostředí dostupné na adrese: `http://localhost:8080`.

[1] `https://www.docker.com/`

[2] Protože inicializace databáze je při vytváření kontejneru pomalá a Docker neumožňuje pozdržení spuštění závislých kontejnerů dokud inicializace nedoběhne, je nutné nejdříve spustit kontejner s databází samostatně. K tomu slouží příkaz: *docker-compose up db*. Při dalším startu vytvořeného kontejneru již oddělené spouštění není potřebné.

# Dodatek D

# Obsah elektronické přílohy

Elektronická příloha k této práci obsahuje:

- zdrojové kódy implementace anglicko-českého slovníku včetně zdrojových kódů frameworku a nástrojů potřebných pro úpravu zdrojů — adresář *codetionary*,
- zkompilovaný program pro 64-bitové linuxové systémy včetně a nástrojů potřebných pro úpravu zdrojů — adresář *codetionary-bin*,
- automaticky generovanou dokumentaci ke zdrojovým kódům — adresář *codetionary-doc*,
- zdrojové kódy anotačního prostředí — adresář *codetionaryevaluationtool*,
- zkompilovaný program anotačního prostředí — adresář *codetionaryevaluationtool-bin*,
- data použitá a získaná během vyhodnocování systému — adresář *codetionaryevaluationtool-data*,
- elektronickou verzi tohoto textu.

Součástí přílohy nejsou všechna data potřebná ke spouštění hlavního programu kvůli jejich velikosti, viz příloha B.