
**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**
**BRNO UNIVERSITY OF TECHNOLOGY**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**
**FACULTY OF INFORMATION TECHNOLOGY**

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**
**DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA**


# POROVNÁVÁNÍ ANOTAČNÍCH NÁSTROJŮ

**COMPARISON OF ANNOTATION TOOLS**

**BAKALÁŘSKÁ PRÁCE**
**BACHELOR'S THESIS**


**AUTOR PRÁCE** **DÁVID PREXTA**
**AUTHOR**

**VEDOUCÍ PRÁCE** **Ing. JAROSLAV DYTRYCH**
**SUPERVISOR**



**BRNO 2017**

*Zadání bakalářské práce/20061/2016/xprext00*

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií Akademický rok 2016/2017

# Zadání bakalářské práce

Řešitel: **Prexta Dávid**
Obor: Informační technologie
Téma: **Porovnávání anotačních nástrojů**
**Comparison of Annotation Tools**
Kategorie: Zpracování řeči a přirozeného jazyka

Pokyny:

1. Seznamte se s nástroji pro automatické sémantické anotování textu a s formáty jejich výstupů.
2. Prostudujte existující řešení pro srovnávání anotačních nástrojů a vyhodnocení jejich úspěšnosti.
3. Navrhněte nový nástroj pro porovnávání anotačních nástrojů, který umožní práci s velkými datovými sadami. Kromě statistik musí poskytnout i možnost detailního vyhodnocení výsledků porovnávání a identifikaci chyb, kterých se jednotlivé nástroje dopouštějí.
4. Implementujte navržené řešení.
5. Zhodnoťte dosažené výsledky a navrhněte možná rozšíření do budoucna.

Literatura:

- Dle doporučení vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Dytrych Jaroslav, Ing.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2
L.S.

doc. Dr. Ing. Jan Černocký
*vedoucí ústavu*

# Abstrakt

Táto práca sa zaoberá problematikou porovnávania anotačných nástrojov pri práci s rozličnými dátovými sadami a získaním výsledkov porovnávania použiteľných pre vylepšenie znalostnej bázy anotátorov. V práci sú analyzované existujúce riešenia a ich nedostatky, z ktorých sú vyvodené požiadavky na nové riešenie. Ďalšie časti sa zaoberajú návrhom, implementáciou a testovaním výsledného nástroja, ktorý je v závere zhodnotený a sú navrhnuté možné rozšírenia do budúcna.



# Abstract

This work deals with the comparison of annotation tools when working with various data sets, and obtaining the results of comparisons useful for improving the knowledge base of the annotators. The thesis analyzes the existing solutions and their drawbacks, from which the proposals of the new solution are deduced. The other sections deals with the design, implementation and testing of the resulting tool, which is evaluated at the conclusion, and possible future extensions are suggested.



# Kľúčové slová

Anotačné nástroje, rozpoznávanie pomenovaných entít, porovnanie výsledkov, identifikácia chýb, spracovanie prirodzeného jazyka, SEC, NER, veľké dáta

# Keywords

Annotation tools, named entity recognition, result comparison, error identification, natural language processing, SEC, NER, big data


# Citácia

PREXTA, Dávid. *Porovnávání anotačních nástrojů*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Dytrych

# Porovnávání anotačních nástrojů

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Jaroslava Dytrycha. Ďalšie informácie mi poskytli Doc. RNDr. Pavel Smrž, Ph.D. a Ing. Lubomír Otrusina. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Dávid Prexta
16. mája 2017


## Poďakovanie

Rád by som sa poďakoval členom Výskumnej skupiny znalostných technológií za všetky poskytnuté informácie a predovšetkým vedúcemu práce pánovi Ing. Jaroslavovi Dytrychovi za cenné rady, odbornú pomoc a priateľský prístup pri vypracovávaní tejto bakalárskej práce.

# Obsah

# Zoznam obrázkov

# Zoznam tabuliek

# Kapitola 1

# Úvod

Táto práca sa zaoberá návrhom a implementáciou nástroja pre porovnávanie anotačných nástrojov, ktorý dokáže pracovať s veľkými dátovými sadami, poskytuje detailné informácie k výsledkom porovnávania a umožňuje identifikáciu chýb, ktorých sa anotačné nástroje dopustili, spolu s možným spôsobom opravy.

Kapitola 2 uvedie čitateľa do problematiky, oboznámi ho s nástrojmi na automatické anotovanie textu a s dostupnými riešeniami pre ich porovnávanie, u ktorých sa zameriava hlavne na ich problémy. Na základe týchto nedostatkov sa vytvoria požiadavky na nové riešenie.

V 3. kapitole sú priblížené dostupné dátové sady a rozoberá sa problematika spracovania týchto sád pomocou nástrojov na analýzu prirodzeného jazyka. Na základe analýzy spracovania dátových sád sú vyčlenené výstupy z vybraných krokov spracovania, ktoré sa použijú ako vstup nástroja.

V kapitole 4 sú stručne popísané technológie použité pri implementácii. V kapitole 5 je čitateľ oboznámený s návrhom celého nástroja, popisom hlavných komponentov a ich vzájomným prepojením spolu s návrhom prípravy dátových sád. Implementácia nástroja je obsiahnutá v kapitole 6. Obsahuje podrobnejší popis funkcionality netriviálnych komponentov a knižníc pre prípravu dátovej sady.

7. kapitola oboznamuje čitateľa s uskutočnenými testami a ich výsledkami. Nástroj je porovnaný z hľadiska využitia operačnej pamäte a rýchlosti pri spracovaní vstupu oproti existujúcemu riešeniu. Ďalej je analyzovaná doba celého porovnávania pri rôznych spôsoboch spustenia nástroja a taktiež sú analyzované výsledky porovnania dátovej sady Wikipédie s anotátorom NER pomocou novo vytvoreného nástroja.

Posledná kapitola zhrňuje výsledky práce a udáva možné rozšírenia a vylepšenia nástroja v budúcnosti.

# Kapitola 2

# Porovnávanie anotačných nástrojov

Behom posledných rokov sa počet nástrojov pre sémantické anotovanie textu značne zvýšil. Pre väčšinu týchto nástrojov existujú publikované výsledky, ktoré by bolo možné využiť na ich porovnanie. Problém je, že pre každý z nich sú tieto výsledky získavané z rôznych dátových sád a sú počítané iným spôsobom. Takéto porovnanie by teda bolo veľmi náročné a neobjektívne. To vedie k vzniku nástrojov umožňujúcich porovnanie anotátorov s využitím rôznych dátových sád a vyhodnocovanie výsledkov jednotným spôsobom.

Táto kapitola sa ďalej zaoberá vysvetlením, čo sú to nástroje na automatické sémantické anotovanie textu, momentálne dostupnými nástrojmi na porovnávanie, ich výsledkami a nedostatkami, z ktorých sú odvodené požiadavky na nové riešenie.

## 2.1 Nástroje na automatické anotovanie textu

Pre porovnávanie nástrojov na automatické sémantické anotovanie textu je najprv potrebné pochopiť, čo tieto nástroje umožňujú. Sémantické anotovanie je proces spracovania daného textu alebo iného obsahu, pri ktorom sú priradené dodatočné informácie k určitým konceptom daného obsahu. Po sémantickej anotácii sa dokument stáva zdrojom informácie, ktorú je možné interpretovať a ďalej použiť počítačom.

Anotátor analyzuje vstupný text a identifikuje v ňom dôležité koncepty a entity ako napr. dátumy, ľudí, miesta, udalosti apod. Po tomto je potrebné nájdené entity jednoznačne identifikovať podľa dostupnej znalostnej bázy (angl. knowledge base skrátene KB), ktorá je zdrojom komplexných informácií o rôznych entitách a je spracovateľná počítačom. Pre identifikáciu konkrétnej entity v znalostnej báze sa používa URI. Všetky entity nájdene v texte sú potom jednoznačne definované podľa dostupnej znalostnej bázy, napr. New York je identifikovaný ako mesto a jednoznačne určený ako New York, USA.

Existuje veľa anotátorov komerčných a aj bezplatných, medzi najznámejšie patrí napr. Alchemy API[1], TAGME[2], Cogito API[3] apod. Vo výskumnej skupine znalostných technológií (KNOT) sa pracuje s vlastným nástrojom pre rozpoznávanie pomenovaných entít (NER) dostupným pomocou nástroja SEC (Semantic Enrichment Component), ktorý je popísaný nižšie.

[1] Alchemy API: https://www.ibm.com/watson/developercloud/doc/alchemylanguage/index.html
[2] TAGME: http://pages.di.unipi.it/ferragina/cikm2010.pdf
[3] Cogito API: https://developer.cogitoapi.com/docs

## 2.2 Semantic Enrichment Component

Výskumná skupina znalostných technológií (skrátene KNOT) pracuje na vývoji nástroja Semantic Enrichement Component (skrátene SEC), ktorý sprístupňuje služby rôznych nástrojov pre sémantické obohacovanie textu. Podstatná je funkcia umožňujúca anotáciu vstupného dokumentu pomocou anotátora zvoleného užívateľom, a možnosť vyhľadávania entít podľa zadaných URI, ktorá poskytuje informácie z KB k nájdenej entite vrátane jej typu [6]. Táto funkcia sa bude dať využiť pri príprave dátovej sady, kedy bude potrebné jednotlivým entitám priradiť typ z KB, ktorú využíva NER.

### 2.2.1 Práca so SEC

SEC pre svoju činnosť využíva Unix domain sockety a komunikácia s ním je založená na modele klient – server. Pre prácu so SEC je potrebné mu predať konfiguráciu vo formáte JSON, ktorá obsahuje informácie o aktuálnej požiadavke. V konfigurácii musí byť vždy udaná požadovaná služba a formát výstupu.

Existujú klientské skripty, pomocou ktorých sú sprístupnené služby poskytované serverom. Keďže môj nástroj bude pre svoju činnosť od nástroja SEC vyžadovať len službu pre anotáciu textu a získanie entít podľa odkazov a konfigurácia bude vždy rovnaká, bude vhodné vytvoriť modul, ktorý bude priamo komunikovať s nástrojom SEC a dožadovať sa na túto službu bez využitia klientských skriptov a ušetrí tak čas.

| | |
|---|---|
| mg4j | Výstup obsahuje vstupný text vo vertikálnom formáte mg4j. Na každom riadku sa môžu nachádzať dodatočné informácie ku konkrétnemu slovu vo formáte TSV. Význam jednotlivých stĺpcov vstupného vertikálu závisí od konfigurácie, ktorú je možné zadať alebo sa použije predvolená konfigurácia podľa prvých 13 riadkov mg4j[4]. |
| Manatee | Výstup je vo vertikálnom formáte Manatee. Informácie o anotáciách sa nenachádzajú na riadku ako u mg4j, ale sú doplnené značkami XML podľa typu nájdenej entity, napr. `<location>`. Každá značka obsahuje atribúty jednoznačne identifikujúce nájdenú entitu (odkazy do znalostnej báze, na obrázky apod.). |
| Manatee2 | Výstup vo formáte Manatee2 sa líši od formátu Manatee tým, že nájdené anotácie nie sú doplnené pomocou značiek XML, ale sú pridané do dodatočných stĺpcov podobne ako u mg4j. |
| Elasticsearch | Výstup pre každý vstupný vertikál obsahuje kolekciu štruktúr `annotation`. Tieto štruktúry obsahujú položky `article` a `title` s devertikalizovaným vstupným textom a názvom a položku `uri` s URI vstupného dokumentu. Za každým slovom v texte sa v hranatých zátvorkách nachádzajú informácie z pôvodného dokumentu a prípadne ďalšie informácie, ak sa jedná o nájdenú anotáciu. |

Tabuľka 2.1: Výstupné formáty nástroja SEC pri službe anotovania vertikálu

[4] Význam jednotlivých stĺpcov formátu mg4j https://docs.google.com/spreadsheets/d/1S4sJ00akQqFTEKyGaVaC3XsCYDHh1xhaLtk58Di68Kk/edit

Výstupný formát nástroja SEC je možné zvoliť z niekoľkých možných variant. Výstupné formáty pri využití služby anotovania vertikálneho textu sú dostupné v tabuľke 2.1. Dostupné formáty pre službu anotovania obyčajného textu sú popísané v tabuľke 2.2.

Nástroj SEC je dostupný aj ako verejná služba na adrese *http://sec.fit.vutbr.cz/* bežiaca na porte 8082.

| | |
|---|---|
| HTML | Výstupom je dokument vo formáte HTML obsahujúci pôvodný text rozšírený o nájdené anotácie, umiestnené v HTML prvku `<span>` s triedou `annotation`. Každá anotácia obsahuje jej prislúchajúce vlastnosti, ako sú napr. typ, odkazy do znalostnej bázy apod., umiestnené v prvkoch HTML `<span>` a navyše obsahujú vizuálne informácie ako obrázky, mapy apod. |
| XML | Výsledný dokument je vo formáte XML. Výstup obsahuje pôvodný text rozšírený o nájdené anotácie, ktoré sú umiestnené v značkách `<annotation>`. Jednotlivé anotácie sa potom nachádzajú v značkách XML s názvom podľa typu (napr. `<person>` `<location>` apod.), ktoré obsahujú k nim príslušné atribúty ako napr. odkazy do znalostnej bázy. |
| SXML | Výstup je taktiež vo formáte XML, ktorý ale obsahuje len nájdené anotácie. Hlavný rozdiel oproti XML je ten, že všetky anotácie sa nachádzajú v značkách `<suggestion>`, anotovaný text je umiestnený v značkách `<text>` a atribúty anotácie sú umiestnené v samostatných značkách `<attribute>`. Je určený pre ďalšie spracovanie. |
| Text | Výstupom je textový dokument čitateľný človekom, obsahujúci len nájdené anotácie a ich vlastnosti. Tento formát je vhodný ako výstup pre užívateľa, nie je vhodný na ďalšie spracovanie. |
| NIF | Výstupom je dokument vo formáte NIF, ktorý obsahuje pôvodný text a k nemu nájdené anotácie. Formát NIF je bližšie popísany v použitých technológiách 4.2 |
| RDF | Výstup je vo formáte RDF[5] vypracovanom organizáciou W3C a obsahuje len anotácie nájdené vo vstupnom texte. |
| Index | Obsahuje pôvodný dokument spolu s nájdenými anotáciami. Je určený pre indexáciu pomocou ElasticSearch. |

Tabuľka 2.2: Výstupné formáty nástroja SEC pri službe anotovania

### 2.2.2 Podpora viacerých NERov

Aby môj nástroj mohol porovnávať rôzne anotátory, je potrebné mať dostupné rozhranie pre komunikáciu s týmito nástrojmi. SEC poskytuje možnosť vytvoriť rozhranie pre ľubovoľný anotačný nástroj, ktorý je potom zvolený pomocou parametra. Komunikácia s anotátormi sa teda vždy bude uskutočňovať cez SEC a vždy rovnakým spôsobom. V nástroji SEC budú podporované nasledujúce anotátory:

[5]Resource Description Framework https://www.w3.org/RDF/

1. **Cogito API** vyvinuté spoločnosťou Expert Systems poskytuje online dostupnú službu pre sémantické anotovanie textu. Táto služba je platená ale je možné využiť aj neplatenú variantu, ktorá je obmedzená na maximálny počet volaní 500 za deň. Vstup, s ktorým pracuje Cogito API, musí byť vo formáte prostého textu a výstup je možné zvoliť z formátov JSON, XML alebo RDF.

2. **Alchemy API** vyvinuté spoločnosťou Alchemy API patriacou pod spoločnosť IBM poskytuje podobne ako Cogito API online dostupnú službu pre sémantické anotovanie. Podobne poskytuje platenú aj neplatenú verziu, kde neplatená je obmedzená na 1000 volaní služby za deň. Požadovaný formát vstupu je prostý text a výstup je poskytovaný vo formáte JSON alebo XML.

3. **Tagme** je nástroj vyvinutý laboratóriami A3 lab, University of Pisa. Taktiež sa jedná o online dostupnú službu, v tomto prípade už ale o neplatenú so žiadnymi obmedzeniami na volania. Vstupný formát je znova prostý text a výstup je dostupný len vo formáte JSON.

4. **Aida** je bezplatný offline nástroj na sémantické anotovanie textu, vyvinutý v Max Planck Institute for Informatics in Saarbücken, Nemecko. Pre jeho funkčnosť je potrebná Java 8 a databázový systém PostgreSQL pre uloženie znalostnej bázy. Po spustení je AIDA dostupná ako webová služba, ktorá je dostupná na lokálnej sieti. Podporuje vstup vo formáte prostého textu a JSON a výstup je možný len vo formáte JSON.

Vďaka tejto podpore zo strany nástroja SEC nebude v mojom nástroji potrebné vytvárať rozhrania pre každý anotátor, ale pri volaní služby sa predajú informácie o tom, ktorý z nich sa má využiť. Samotné rozhrania pre komunikáciu budú integrované v nástroji SEC a bude ich teda možné využiť aj mimo môj nástroj.

## 2.3 BAT framework

Jedným z nástrojov umožňujúcich porovnanie výsledkov anotátorov je BAT framework, vyvinutý laboratóriami A3, Department of Computer Sience, University of Pisa [7]. Dokáže pracovať len s Wikipedia knowledge base, čo je príliš obmedzujúce. Každá URI entity v dátovej sade alebo vo výsledku anotátora je prevádzaná na URI Wikipédie a po tom na príslušné Wikipédia ID [8]. Entity, u ktorých to nie je možné, sú zahodené a dochádza tak k strate.

Súčastou frameworku je päť dátových sád, ktoré je možné použiť ako vstup. Je možné pridať aj ďalšie vlastné dátové sady, problém je ale to, že dátové sady nemajú žiadny špecifikovaný formát, a teda je pre pridanie potrebné vytvoriť novú triedu, ktorá umožní načítať a spracovať novú sadu. Vytvorenie samotnej sady je taktiež na užívateľovi, pretože framework neposkytuje funkcie pre jej vytvorenie.

Ďalej poskytuje rozhrania pre päť anotátorov, ktoré sa dajú využiť pre uskutočnenie porovnávaní. Opäť je možné pridať ďalšie anotátory, a to vytvorením rozhrania pre nový anotátor.

Nástroj vyberá jednotlivé časti textu dátovej sady a tie predáva anotátoru, s ktorým pracuje ako s čiernou skrinkou. Po spracovaní preberá výsledok, ktorý ďalej analyzuje, porovnáva s entitami v dátovej sade a uskutočňuje výpočty pre zistenie, do akej miery je výsledok správny [8].

Nevýhodou BAT frameworku je hlavne obmedzenie na jednu KB, výsledky len vo forme štatistík a zložité pridávanie nových dátových sád.

## 2.4 General Entity Annotation Benchmark

Je ďalší nástroj pre porovnávanie anotačných nástrojov, vyvinutý výskumnou skupinou Agile Knowledge Engineering and Semantic Web (AKSW) na univerzite University of Leipzig, založený na BAT frameworku. Narozdiel od BAT nie je obmedzený len na Wikipedia knowleage base, ale využíva rozpoznávanie URI, a tak môže pracovať s rôznymi KB [13]. Ďalej obsahuje nové spôsoby porovnávania a metriky výpočtu, ktoré sú popísane nižšie.

Aktuálna verzia nástroja GERBIL obsahuje jedenásť základných dátových sád a deväť anotátorov. Dátové sady je možné pridávať dvoma spôsobmi, a to buď vytvorením rozhrania pre novú sadu, alebo prevodom dátovej sady do formátu NIF. Pridávanie anotátorov je podobné ako v BAT frameworku, a to vytvorením rozhrania pre daný anotátor.

### 2.4.1 Možnosti porovnávaní v GERBIL

GERBIL obsahuje niekoľko možností testovania anotátorov. Pre kontrolu pozície entít využíva dva sposôby, a to silná kontrola a slabá kontrola. Pri silnej kontrole sa entita považuje za nájdenú, ak sa jej indexy presne zhodujú s indexmi nejakej entity z dátovej sady. Pri slabej kontrole sa entita považuje za nájdenú aj v prípade, že len prekrýva časť nejakej entity z dátovej sady, napr. v dátovej sade existuje entita New York City, anotátor vráti entitu New York. Pri silnej kontrole sa nejedná o zhodu, slabá kontrola bude považovať entity za zhodné z hľadiska pozície. Všetky porovnávania poskytované gerbilom sú popísané v tabuľke 2.3.

Nástroj na automatické anotovanie textu, ktorý sa využíva vo výskumnej skupine znalostných technológií (KNOT), nedokáže spracovávať text s už vopred vyznačenými entitami a vracať k nim potrebné informácie. Porovnávania D2KB a Etyping sa teda použiť nedajú, ich princíp bude ale možné využiť pri iných úkonoch.

### 2.4.2 Vstup a výstup

GERBIL pre svoju prácu využíva formát Natural Language Programming Interchange Format (NIF), ktorý je popísaný v použitých technológiách. Dátové sady a výstupy z anotátorov očakáva v tomto formáte. Formát vstupu je teda jasne daný, čo je oveľa efektívnejšie než situácia u BAT frameworku, kde je potrebné pre každú dátovú sadu implementovať nové rozhranie. Dátovu sadu je ale stále nutné vytvoriť vo formáte NIF a taktiež je potrebné vytvoriť rozhranie pre anotátor, ktoré prevedie výstup do tohto formátu. GERBIL ale poskytuje knižnice, ktoré umožňujú vytvorenie dátových sád v NIF a zjednodušujú vytváranie rozhrania pre anotátor [13]. Stále je ešte dostupná možnosť implementovať vlastné rozhrania ako u BAT frameworku.

Aby mohol GERBIL s novým anotátorom komunikovať, je potrebné, aby bol anotátor dostupný cez REST API, a nástroju GERBIL sa pridá konfigurácia k tomuto API. Všetky dáta, ktoré sú zasielané, sú vo formáte NIF a po ich prijatí sú využité vyššie zmienené rozhrania pre ich spracovanie a predanie anotátoru (prípadne pre prácu s dátovou sadou). Obrázok 2.1 ukazuje tok dát v nástroju GERBIL.

Rozhrania medzi dátovou sadou a anotátorom sú potrebné len v prípade že dátová sada nie je vo formáte NIF a anotátor nedokáže pracovať s formátom NIF priamo.

| | |
|---|---|
| A2KB | Pri tomto porovnávaní sa z dátovej sady vyberie čistý text, ktorý sa predá anotátoru na spracovanie. Vo výsledku očakáva pomenované entity, ktoré boli nájdené v texte spolu s ich odkazmi do KB. Výsledky sa porovnávajú s entitami danými v dátovej sade. |
| D2KB | Porovnávanie, pri ktorom sa anotátoru predáva na vstup text s už vopred označenými entitami. Antotátor by mal k týmto entitám priradiť odkaz do určitej KB. Pri tomto type sa neberú do úvahy pozície entít. |
| C2KB | Anotátoru sa predá čistý text z dátovej sady pre spracovanie. Vo výsledku sa očakáva zoznam entít nájdených v dokumente, bez udania ich pozície v texte a ich odkazov do KB. |
| ERec | Funguje podobne ako A2KB, s tým rozdielom, že anotátor vracia len zoznam entít nájdených v texte a ich pozície. Odkazy do KB nie sú pri tomto porovnávaní potrebné ani zohľadňované. |
| Etyping | Podobne ako pri D2KB je anotátoru predaný text s už vopred vyznačenými entitami. K týmto entitám musí anotátor určiť typ entity z KB. |

Tabuľka 2.3: Možnosti porovnávani v GERBIL

### 2.4.3 Nevýhody gerbilu

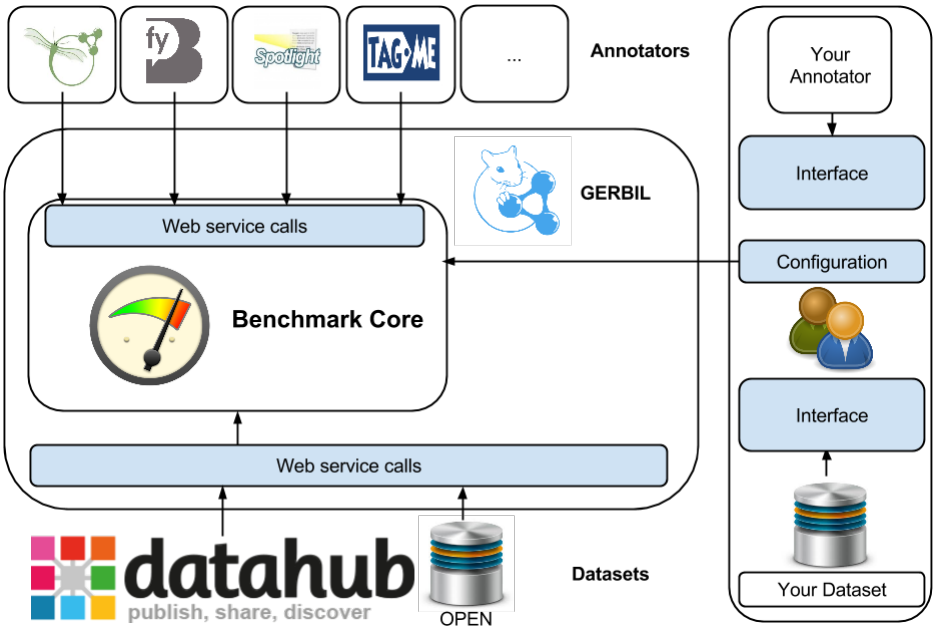
Súčasne dostupné nástroje na porovnávanie majú niekoľko problémov. Tieto nástroje sú navrhnuté pre prácu s dátovými sadami, ktoré sú krátke a ich veľkosť sa pohybuje radovo v stovkách kilobajtov. Dátové sady, s ktorými sa pracuje vo výskumnej skupine KNOT, sú oveľa väčšie, napr. veľkosti dátových sád z projektu Wikilinks sa pohybujú radovo v jednotkách gigabajtov. Pri takýchto dátových sadách sú vysoké nároky na systémove prostriedky a to hlavne na operačnú pamäť, keďže tieto nástroje načítavajú celé dátové sady do operačnej pamäte. Z tohto dôvodu je taktiež obmedzený maximálny počet paralelne bežiacich procesov nástroja GERBIL. Pri spracovaní vstupov vo formáte NIF sa využíva sekvenčné vyhľadávanie, ktoré je pri veľkom počte dokumentov a entít v dátových sadách pomalé a teda značne predlžuje celý proces.

Ďalší problém sú spôsoby porovnávania a ich nekompatibilita s nástrojom SEC. Ako už bolo vyššie zmienené, tak porovnávania D2KB a Etyping sa v SEC uskutočňovať nedajú. Ostatné porovnávania by bolo možné použiť, ale chýbajú v nich porovnávania typov, ktoré sú priradené entitám v SEC.

Ďalším problémom sú výsledky poskytované týmito nástrojmi. Výstupy nástroja GERBIL sú len štatistické a pomocou nich môžme len zistiť, ktorý anotátor dopadol lepšie a aké skóre dosiahol.

## 2.5 Požiadavky na moje riešenie

Nový nástroj musí byť schopný spracovávať vstupy vo formáte NIF. Momentálne dostupné datové sady sú vo formátoch mg4j a vertikálny text, preto nástroj musí obsahovať podporu




Obr. 2.1: Tok dát v nástroji GERBIL

pre prevod týchto formátov do NIF. SEC dokáže vytvárať výsledok vo formáte NIF a obsahuje podporu viacerých nástrojov NER, preto je potrebné zabezpečiť komunikáciu so SEC, cez ktorú sa bude pristupovať k jednotlivým anotátorom.

Výstup z nástroja by mal byť obšírnejší a okrem štatistických výsledkov musí uchovávať detailné informácie o danom porovnávaní. Výsledkom by mali byť informácie o entitách dokumentu rozdelených podľa toho či anotátor uspel pri ich hľadaní, alebo uspel len čiastočne a splnil len niektoré kritériá, prípadne neuspel alebo sa jedná o entity nadbytočné. Ku každej entite musia byť uvedené všetky potrebné informácie a to jej indexy, odkazy do KB, odkaz na dokument, z ktorého pochádza a typ entity.

Je potrebné navrhnúť nový spôsob spracovávania vstupov a porovnávania tak, aby systémove požiadavky boli čo najnižšie a znížilo sa tak využitie operačnej pamäte na minimum. Taktiež je potrebné, aby sa výrazne znížil čas potrebný pre spracovanie vstupu.

Nástroj bude musieť uskutočňovať porovnávania podobné A2KB a ERec. Bude však musieť umožňovať porovnávanie entít podľa ich typov. Pri každom porovnávaní si bude možné zvoliť, čo všetko sa má kontrolovať – zhoda pozície, odkazy entít do KB, typy entít. Musí taktiež umožňovať filtrovanie entít podľa typu a podľa toho pracovať pri danom experimente len s entitami určitého typu. Pre porovnávanie typov je však potrebné tieto typy určiť a preto bude potrebná schopnosť priradiť typy entitám v dátovej sade pred porovnávaním.

Okrem nových spôsobov porovnania bude potrebné, aby nástroj vedel identifikovať chyby, ktorých sa jednotlivé nástroje dopúšťali, podľa ktorých bude možné nástroj ďalej upravovať a vylepšovať.

# Kapitola 3

# Dátové sady

Najväčšia dátová sada, ktorej spracovaním sa zaoberá skupina KNOT, je sada z projektu CommonCrawl[1]. Jedná sa o súhrn veľkého počtu internetových stránok, ktoré sú zbierané a agregované neziskovou organizáciou Common Crawl a voľne dostupné každému [12]. Dáta sú uložené vo formáte web archive, ktorý pozostáva z množiny WARC záznamov. Každý záznam obsahuje hlavičku metadát a samotný obsah. Veľkosť jednej takejto sa pohybuje okolo dvadsaťpäť terabajtov (napr. sada CC-2015-27 má 26,28 TB, CC-2015-40 má 21,13 TB a CC-2016-36 má 26,78 TB).

Ďalšie sady sú sada ClueWeb09 a ClueWeb12, ktoré podobne ako Commoncrawl obsahujú súhrn rôznych internetových stránok, ktorých počet je však výrazne nižší. Jedná sa o platené dátové sady projektu Lemur, University of Massachusetts. Dáta sú taktiež uložené vo formáte web archive a veľkosť týchto sád je 5,54 TB pre ClueWeb12 a 5 TB pre ClueWeb09.

Z projektu Wikilinks, ktorého cieľom bolo z dátových sád ClueWeb a CommonCrawl vyextrahovať odkazy na Wikipédiu vznikla ďalšia dátová sada. Jedná sa o podmnožinu stránok z vyššie zmienených dátových sád, ktoré v sebe obsahujú odkazy na Wikipédiu. Táto sada je taktiež uložená vo formáte web archive a jej veľkosť približne 115 GB (napr. sada Wikilinks CC-2015-18 – 115,47 GB, CC-2015-22 – 115,18 GB).

Posledná je sada obsahujúca stránky Wikipédie, získane z Wikipedia dumpu a je uložená v súboroch vo formáte predspracovanej Wikipédie. Tieto súbory pozostávajú z prvkov `<doc>`, obsahujúcich atribúty určujúce unikátne ID, titulok stránky a URI dokumentu. Vo svojom vnútri obsahujú text samotnej stránky Wikipédie, využívajúci základné značky HTML. Veľkosť jednej tejto sady je približne 14 GB.

## 3.1 Spracovanie dátových sád

Všetky vyššie zmienené dátové sady sú spracovávané nástrojmi na analýzu prirodzeného jazyka. Prvý krok je prevod vstupnej sady pomocou nástroja zvaného vertikalizátor, do vertikálneho formátu, ktorý na každom riadku obsahuje jednu pozíciu (prípadne štruktúrnu značku) a u pozícií sú tabulátorom oddelené jednotlivé atribúty [2]. Pri vertikalizácii taktiež dochádza k odstráneniu nedôležitých časti stránok a vo výsledku ostáva len užitočný text na spracovanie. Každý dokument vo vertikálnom formáte je obalený v značkách `<doc>` s povinnými parametrami `title` – obsahujúci titulok dokumentu a `url` – URL adresa do-

[1] http://commoncrawl.org/

kumentu. Pri vertikalizácii taktiež dochádza k rozdeleniu obsahu dokumentu na odstavce nachádzajúce sa v značkách `<p>`.

Pri tvorení rozsiahlych dátových sád často dochádza k vzniku duplicít uložením rovnakých dokumentov niekoľkokrát alebo výskytom rovnakých časti obsahu u viacerých dokumentov. O odstránenie týchto duplicít sa stará druhý krok spracovania – deduplikácia. Odstraňované sú celé dokumenty v prípade viacnásobného uloženia, alebo sa odstraňujú odstavce, na ktoré bol dokument rozdelený pri vertikalizácií, ktoré už existujú v iných dokumentoch. Vstup aj výstup je v rovnakom vertikálnom formáte s tým, že výstup je ukrátený o duplicitné časti.

V ďalšom kroku sú jednotlivým pozíciám vo vete pridelené im odpovedajúce slovné druhy a lemma. Výstupom je vertikálny súbor, ktorý obsahuje na každom riadku jednu pozíciu, jej slovný druh a lemmu, ale štruktúrne značky sa už neumiestňujú na osobitnom riadku ako pri vertikalizácii, ale ku každej pozícií sú udané značky nachádzajúce sa pred a za ňou.

Po tomto kroku nasleduje závislostná analýza, ktorá každej pozícií vo vete priradí závislé slovo, jeho funkciu vo vzťahu, pozíciu vo vete a lemmu, pozíciu vo vete a vzdialenosť párového slova. Každe slovo je vo vzťahu s práve jedným slovom vo vete. Štruktúrne značky sa vo výstupe opäť nachádzajú na vlastnom riadku a význam jednotlivých stĺpcov odpovedá prvým dvanástim stĺpcom formátu mg4j.

Výsledok z predchádzajúceho kroku sa ďalej sémanticky anotuje pomocou anotátora s využitím funkcie nástroja SEC pre anotovanie vertikálu. Pre výstup sa používa formát mg4j, pričom prvých dvanásť stĺpcov ostáva rovnakých ako vo vstupnom súbore a zvyšné stĺpce závisia od výsledku anotátora. V prípade nájdenej entity sa vo zvyšných stĺpcoch nachádzajú informácie o nájdenej anotácii ako napr. typ, URI do znalostnej bázy, obrázky apod. Na týchto výsledkoch sa následne uskutočňuje indexácia, ktorá slúži k vytvoreniu indexov pre vyhľadávanie.

## 3.2 Zhodnotenie spracovania

Celý vyššie zmienený proces spracovania dátových sád je pri veľkom objeme dát časovo veľmi náročný, preto sa spracovanie takýchto sád uskutočňuje externe na superpočítači Salomon, kde je ale obmedzený počet jádrohodín pre výpočet. Spracovanie jedné dátové sady CommonCrawl vyžaduje až približne 210 000 jádrohodín a teda získanie celého výsledku je veľmi drahé.

Pre vstup nástroja bude teda potrebné využiť niektorý z výstupov spracovania, alebo ak tieto výstupy nebudú vhodné, je potrebné zabezpečiť nový formát vstupu a to tak, aby sa nijak neovplyvnil aktuálny proces spracovania dátových sád vzhľadom k jeho náročnosti.

# Kapitola 4

# Použité technológie

## 4.1 Python

Python je interpretovaný, vysokoúrovňový, dynamicky typovaný, multi-paradigmový programovací jazyk, ktorý vytvoril Guido van Rossum v roku 1991. Medzi silné stránky jazyka patrí dobrá čitateľnosť zdrojového kódu, nezávislosť na platforme, rozsiahla štandardná knižnica a knižnice tretích strán [10]. Dôležitá je taktiež kompatibilita s nástrojom SEC, ktorý je vytvorený práve v tomto jazyku. Z týchto dôvodov bol Python zvolený ako hlavný implementačný jazyk nástroja. Použitá bola verzia 2.7, prednastavená na serveroch KNOT a použitá v nástroji SEC, ktorá je v súčastnosti stále udržiavaná. Najnovšia verzia je v dobe písania práce verzia 3.6.

## 4.2 NLP Interchange Format – NIF

NIF je formát založený na formáte RDF, ktorého cieľom je zaistenie interoperability medzi nástrojmi na spracovanie prirodzeného jazyka, anotáciami a jazykovými zdrojmi (angl. language resources). Primárne je určený na uloženie textu určitého dokumentu a anotácii, ktoré sú uložené v jednotlivých záznamoch [9]. Text je v zázname uložený ako prostý text a ako identifikátor slúži jeho URI. Všetky entity vzťahujúce sa k tomuto dokumentu sa naňho odkazujú práve touto URI, pričom každá entita obsahuje atribúty reprezentujúce jej pozíciu v danom texte. Okrem toho môžu byť uvedené ďalšie atribúty určujúce ďalšie vlastnosti ako napríklad odkazy do znalostnej bázy, typ a podobne.

Tento formát bol zvolený ako vstupný formát dátových sád, a to hlavne vďaka možnosti uloženia textu vo formáte prostého textu a anotácii so všetkými potrebnými informáciami, dobrej čitateľnosti pre človeka a súčasne vďaka kvalitnej dokumentácii a dobrej špecifikácii formátu umožňujúcej dobré spracovanie počítačom.

## 4.3 MediaWiki API

MediaWiki API je webová služba umožňujúca prístup k dátám z databázy MediaWiki. V nástroji je použitá len akcia `query`, slúžiaca pre získanie informácii o konkrétnej stránke z Wikipédie [11]. Použitá je pre aktualizáciu odkazov v príprave dátovej sady a pre kontrolu presmerovania pri identifikácii chýb.

## 4.4 Google Knowledge Graph Search API

Jedná sa o API vytvorené spoločnosťou Google v Decembri 2015, umožňujúce prístup k entitám zo znalostnej bázy Google Knowledge Graph [14]. V nástroji je použitá služba vyhľadania entity podľa Freebase id, u ktorých je vo výsledku uvedená odpovedajúca URI z Wikipédie. To umožnuje mapovať odkazy z Freebase na Wikipédiu alebo DBpédiu.

## 4.5 Requests

Requests je knižnica jazyka Python, umožňujúca jednoduchú a bezpečnú komunikáciu pomocou protokolu HTTP/1.1. Zjednodušuje a sprehľadňuje tvorbu požiadaviek, pre ktoré sú parametre uložené vo forme slovníku textových reťazcov. Taktiež umožňuje získať odpoveď vo formáte JSON, vďaka vstavanému JSON dekodéru [4]. V nástroji je táto knižnica použitá pre komunikáciu s MediaWiki API a Google Knowledge Graph Search API a je použitá aj pri komunikácii s online dostupnými anotátormi v nástroji SEC. Použitou verziou v čase tvorby práce je verzia 2.13.0.

## 4.6 JSON

JavaScript Object Notation je formát slúžiaci pre prenos dát, jednoduchý pre spracovanie počítačom a súčasne čitateľný a jednoducho upraviteľný človekom, nezávislý na programovacom jazyku [5]. Znalosť tohto formátu je potrebná pre implementáciu spracovania odpovede z anotátorov, MediaWiki a Google Knowledge Graph Search API, ktoré poskytujú odpoveď práve v tomto formáte, a taktiež na tvorbu vstupnej konfigurácie nástroja SEC.

## 4.7 HTTP

Hypertext Transfer Protocol je bezstavový aplikačný protokol fungujúci na princípe požiadaviek – odpoveď, pôvodne vytvorený pre prenos hypertextových dokumentov vo formáte HTML, dnes sa ale využíva aj na prenos iných informácii [3]. Najnovšia verzia protokolu je HTTP/2, ktorá vznikla v roku 2015, definovaná normou RFC 7540 [1].

Znalosť tohto protokolu je potrebná na implementáciu modulov pre komunikáciu s anotátormi, MediaWiki a Google Knowledge Graph Search API.

# Kapitola 5

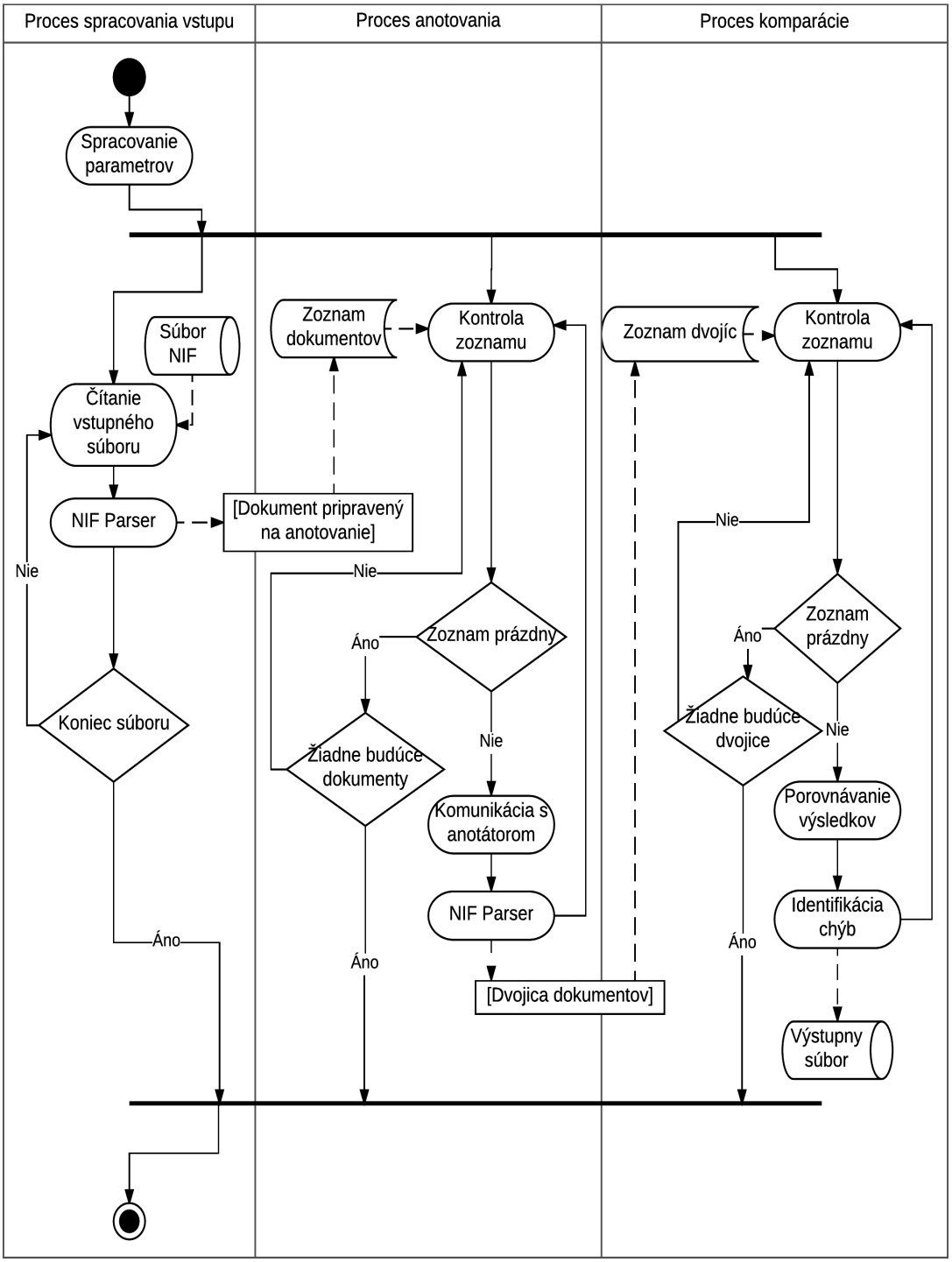
# Návrh

Nový nástroj pracuje so vstupnou dátovou sadou vo formáte NIF, z ktorej postupne získava jednotlivé dokumenty. Tie spracuje a predá príslušnemu anotátoru na analýzu, z ktorej získa výsledok vo formáte NIF. Ten sa znovu spracuje a následne sa uskutoční porovnanie výsledku s dokumentom z dátovej sady.

Nástroj je navrhnutý ako skupina paralelne bežiacich procesov, ktoré si medzi sebou predávajú dáta a vykonávajú určitú časť celej činnosti nástroja. Na obrázku 5.1 je znázornená architektúra nástroja predstavujúca jednotlivé procesy a predávanie dát medzi nimi.

Proces spracovania vstupu číta vstupný súbor vo formáte NIF a jednotlivé dokumenty spracováva knižnicou NIF Parser. Keď je spracovaný celý vstupný súbor, tento proces uvedomí ostatné o tom, že ďalšie dokumenty už nebudú. Spracované dokumenty z dátovej sady sú predané do zoznamu dokumentov určených na anotovanie. Z tohto zoznamu si ich vyberá proces anotovania, ktorý pre nich získa výstup z anotátora a ten spracuje pomocou knižnice NIF Parser. Následne je k dispozícii dvojica dokumentov pripravená pre porovnávanie, ktorá sa vloží do zoznamu dvojíc pre porovnávanie. Posledný proces komparácie si z neho vyberá všetky dvojice, ktoré porovnáva, uskutočňuje identifikáciu chýb a výsledky zapisuje do výstupných súborov. Proces anotovania môže mať viacero inštancií pre urýchlenie anotovania dokumentov.

Kapitola ďalej popisuje návrh hlavných komponentov nástroja z obrázku 5.1 a taktiež prípravu dátovej sady pred porovnávaním.




Obr. 5.1: Diagram aktivít navrhnutého nástroja

## 5.1 Príprava dátovej sady

Keďže každý anotátor pracuje so vstupným textom vo formáte prostého textu, je potrebné zabezpečiť vstup v tomto formáte. Žiadny z výstupov jednotlivých krokov spracovania dostupných sád takýto formát neposkytujú. Pre vstup by bolo možné použiť vertikálny formát, ktorý je podporovaný aj ako vstup do SEC. Pri jeho použití by však bolo potrebné pri každom spustení uskutočniť devertikalizáciu a potom pracovať so SEC. Predanie vstupu do SEC vo vertikálnom formáte by bolo nevýhodné keďže by sa znovu uskutočnila devertikalizácia, ktorá by sa musela uskutočniť už v nástroji na porovnanie.

Výhodné bude použiť formát NIF, ktorý obsahuje dokument vo formáte prostého textu a k nemu potrebné informácie o entitách. Po spustení sa teda načíta konkrétny text dokumentu a predá sa nástroju SEC, ktorý ho môže priamo spracovávať anotátorom bez ďalších úprav. SEC taktiež poskytuje výstup v tomto formáte. Jeho ďalšou výhodu je aj to, že je pre človeka čitateľnejší narozdiel od vertikálneho formátu, takže do výsledkoch porovnania je jednoduchšie nahliadnuť na konkrétny dokument.

Pri veľmi rozsiahlych dátových sadách, ako je napr. CommonCrawl, je počet dokumentov v tejto sade na jednom serveri okolo 9 000 000 . Spracovanie jedného dokumentu nástrojom SEC trvá približne 1 až 3 sekundy v závislosti od dĺžky textu, a teda porovnávanie takéhoto počtu dokumentov by z toho dôvodu zabralo príliš veľa času. Takéto dátové sady sa ale spracovávajú externe na superpočítači Salomon a výstup zo SEC je uložený vo formáte mg4j, ktorý obsahuje informácie o entitách nájdených nástrojom SEC v tejto sade. Vďaka tomu bude možné previesť tieto výstupy do NIF a uskutočniť v takýchto prípadoch porovnávania bez aktívnej spolupráce so SEC.

## 5.2 Prevod vertikálného súboru a mg4j do formátu NIF

Čítanie vertikálneho súboru sa bude uskutočňovať po riadkoch, ktoré sa budú ukladať do jedného zoznamu. Po načítaní koncovej značky dokumentu `</doc>` sa zoznam načítaných tokenov prevedie do formátu NIF a až potom bude nasledovať čítanie ďalšieho dokumentu. Tým sa minimalizuje pamäťová náročnosť pri prevode, keďže v operačnej pamäti bude vždy len jeden dokument z vertikálneho súboru.

Zoznam načítaných tokenov sa rozdelí na časti nachádzajúce sa medzi značkami `<s>` a `</s>`, reprezentujúcimi jednu vetu dokumentu. Všetky vety budu následne prevedené do formátu prostého textu, pričom sa zároveň uskutoční identifikácia entít v týchto vetách, ku ktorým budú dopočítané presné pozície vo výslednom texte. Okrem toho je potrebné jednotlivým entitám určiť typ podľa znalostnej bázy, na čo nástroj bude využívať službu nástroja SEC `get_entitity_by_uri`, pričom na vyhľadávanie sa použije URI entity z vertikálneho súboru.

Taktiež je potrebné skontrolovať platnosť odkazov entity do znalostnej bázy, DBpédie a Wikipédie v dátovej sade. Niektoré URI môžu byť zastaralé z dôvodu presmerovania, prípadne zle uložené (napr. URI Wikipédie ukončené príponou .html). Pri takýchto odkazoch nástroj aktualizuje URI na platné verzie. Po dokončení do výstupného súboru nástroj uloží príslušne záznamy NIF.

Keďže sa jedná taktiež o vertikálny súbor, čítanie súboru mg4j sa uskutoční podobne ako čítanie obyčajného vertikálneho súboru. Rozdiel je v značke `%%#DOC`, reprezentujúcej začiatok dokumentu. Koncová značka neexistuje a ďalší dokument začína novou značkou `%%#DOC`. Po načítaní jedného dokumentu nástroj znovu rozdelí text na vety a tie spracuje rovnako ako v predchádzajúcom prípade. Vety začínajú značkou `%%#SEN` a taktiež nemajú

koncovú značku. Okrem entít z dátovej sady sa však vo vetách identifikujú aj entity nájdené anotátorom a výsledkom sú teda dva súbory NIF, kde jeden reprezentuje pôvodnú dátovú sadu a jej entity a druhý výsledky z anotátora.

## 5.3 Spracovanie formátu NIF

Zo súčasne dostupných riešení pre spracovanie formátu NIF je k dispozícií knižnica Gerbil Nif transfer, ktorá je súčasťou nástroja GERBIL. Je napísaná v jazyku Java a pre každý dokument vo vstupnom NIF súbore vracia inštanciu triedy Document reprezentujúcu jeden spracovaný dokument. Pre testovanie tejto knižnice bol vytvorený jednoduchý program v jazyku Java, ktorý pomocou zmienenej knižnice musel pre každý dokument vo vstupnom súbore postupne získať inštanciu triedy Dokument. V tabuľke 5.1 sú uvedené výsledky týchto testov.

| **Súbor** | **Počet dokumentov** | **Veľkosť v MB** | **Operačná pamäť v MB** | **Čas** |
|---|---|---|---|---|
| test1.ttl | 15 500 | 42 | 1304 | 10,48 s |
| test2.ttl | 31 000 | 81,4 | 1558 | 18,86 s |
| test3.ttl | 42 500 | 122 | 1780 | 27,71 s |
| test4.ttl | 57 500 | 163 | 2030 | 35,59 s |
| test5.ttl | 77 500 | 209 | 2300 | 44,62 s |

Tabuľka 5.1: Rýchlosť spracovania a využitie operačnej pamäte pri spracovaní dokumentov vo formáte NIF nástrojom Gerbil Nif transfer

Z tabuľky je vidieť, že knižnica značne vyťažuje operačnú pamäť počítača, preto je potrebné vytvoriť vlastné riešenie, ktoré bude aj zároveň rýchlejšie ako testovaná knižnica. Toto riešenie bude implementované ako trieda `NIFParser`, ktorej inštancie budú mať sprístupnenú metódu pre spracovanie dokumentu vo formáte NIF predaného ako parameter tejto metódy. Okrem toho bude obsahovať dve spôsoby spracovania v závislosti od toho, či vstupom bude usporiadaný alebo neusporiadaný NIF.

Spolu s ním bude vytvorená trieda `Dokument`, reprezentujúca jeden spracovaný dokument aj s jeho entitami. Po spracovaní bude trieda `NIFParser` vracať inštanciu práve tejto triedy.

### Čítanie usporiadaného súboru NIF

Súbor NIF pozostáva z jednotlivých záznamov, ktoré sú od seba oddelené jedným prázdnym riadkom. Dokumenty v usporiadanom NIF súbore začínajú záznamom reprezentujúcim nový dokument, za ktorým nasledujú ďalšie záznamy patriace tomuto dokumentu. Po prechode na záznam o novom dokumente už nebudú nasledovať žiadne položky patriace predchádzajúcim dokumentom. Súbory vznikajúce prípravou dátovej sady popísanej vyššie budú vždy usporiadané.

Pre čítanie tohto vstupu bude vytvorená trieda `NIFReader`, ktorej inštancie budú môcť volať metódu pre získanie ďalšieho dokumentu dátovej sady na spracovanie. Dokumenty bude teda možné čítať postupne a tým sa značne ušetrí operačná pamäť, keďže nie je potrebné načítať celý vstupný súbor.

### Čítanie neusporiadaného súboru NIF

Narozdiel od usporiadaného NIF sú v neusporiadanom jednotlivé záznamy chaoticky uložené v súbore a nie je možné uskutočniť čítanie po častiach reprezentujúcich dokument. Celý obsah súboru bude teda potrebné uložiť do operačnej pamäte, rozdeliť na záznamy a tie medzi sebou priradiť na základe odkazov. To by v prípade veľkých dátových sád znamenalo vysoké využitie operačnej pamäte. Časová zložitosť je v tomto prípade lineárna pre priraďovanie entít k dokumentom a celý proces by bol teda veľmi zdĺhavý. Je však potrebné, aby nástroj disponoval možnosťou spracovať tento typ vstupu, keďže niektoré menšie dátové sady sú uložené týmto spôsobom (napr. dátové sady poskytované nástrojom GERBIL).

## 5.4 Anotácia dokumentu

Po načítaní a spracovaní vstupu je potrebné jednotlivé dokumenty anotovať požadovaným anotátorom. Všetky inštancie triedy `Dokument` z predchádzajúceho kroku budu vložené do zoznamu, v ktorom budu čakať na anotovanie. Tento zoznam bude zdieľaný s procesom, ktorý si z neho bude postupne vyberať jednotlivé dokumenty a anotovať ich využitím služby `anotate` nástroja SEC.

Pre komunikáciu s nástrojom SEC bude vytvorená trieda `Anotator`, ktorej bude pri inicializácii parametrom predaný konkrétny anotátor, ktorý sa má použiť pre anotovanie. Všetky inštancie tejto triedy budu taktiež disponovať viacerými konfiguráciami, pre využitie rôznych služieb nástroja SEC. Objekty triedy budú mať k dispozícií metódu `process`, ktorej parametrami budú vstupný text pre anotátor a informácia o tom, ktorá konfigurácia sa pri práci s nástrojom SEC použije. Ako výsledok bude vrátený výstup z nástroja SEC vo formáte NIF. Ten sa spracuje objektom triedy `NIFParser`, rovnako ako vstup z dátovej sady. Po anotovaní je k dispozícii dvojica objektov triedy `Dokument` pripravená na porovnávanie.

## 5.5 Podpora viacerých nástrojov NER

Na pridanie nového anotátora existuje v nástroji SEC trieda `NERTemplate`, predstavujúca šablónu pre anotátor. Podpora viacerých nástrojov NER je teda realizovaná modulom pre každý nový anotátor, obsahujúci triedu, ktorá je potomkom triedy `NERTemplate`. Každá z nich musí poskytovať metódu `_process`, ktorej parametrom bude text určený k anotovaniu.

Jednotlivé moduly komunikujú pomocou protokolu HTTP s konkrétnym anotátorom a po prijatí výsledku ho prevádzajú na výstup v Backusova-Naurovej forme, ktorý predávajú nástroju SEC. Vstupný text je pre každý anotátor vo formáte prostého textu, kedže s týmto formátom dokáže pracovať každý z nich. Všetky atribúty, ktoré nie sú anotátorom vrátené ale sú požadované nástrojom SEC, sa ešte dopočítajú (napr. indexy), a pre každú entitu sa zistia čísla riadkov v KB využívané nástrojom SEC. To sa uskutoční využitím nástroja figa, ktorý pomocou konečného automatu prejde URI entity a v prípade úspechu vráti informácie o entite, vrátane čísel riadkov v KB.

## 5.6 Porovnanie výsledkov

Pre porovnávanie výsledkov je potrebné vytvoriť vlastné riešenie, ktoré bude umožňovať silnú alebo slabú kontrolu pozície entít v texte rovnako ako v nástroji GERBIL. To bude realizované triedou `Comparator`, disponujúcou metódou `compare`, ktorej sa pomocou parametru predá dvojica objektov triedy `Dokument` určená k porovnaniu a informácia o zvolenej kontrole pozícií. Okrem toho bude schopná porovnávať zhodu odkazov do znalostnej bázy, kontrolu typov a prípadne filtrovať porovnávanie len na určitý typ entít. Všetky tieto kritéria budú taktiež predané pomocou parametrov metódy. Ako výsledok bude vracať inštanciu triedy `EvaluationResult`, obsahujúcu informácie o porovnaní.

Každej entite v dátovej sade sa pokúsi podľa kontroly pozície nájsť odpovedajúcu entitu vo výsledku vrátenom anotátorom. Všetky entity, ku ktorým nebude nájdená odpovedajúca entita z anotátoru, budú vyhodnotené ako nenájdené a uložené v zozname nenájdených entít v triede `EvaluationResult`.

Pokiaľ nebude zvolené ďalšie kritérium porovnávania, entity, ku ktorým bola nájdená dvojica, sa uložia do zoznamu nájdených entít. V opačnom prípade budú ale musieť vyhovieť všetkým ďalším kritériám, inak budu uložené do zoznamu chybných entít.

Všetky zvyšné entity nájdené anotátorom budú uložené do zoznamu nadbytočných entít.

## 5.7 Identifikácia chýb

Súčasťou nástroja bude taktiež možnosť identifikovať chyby, ktorých sa nástroj dopustil, a určiť prípadné riešenie tejto chyby a tým vylepšovať znalostnú bázu. Tento krok nástroja bude voliteľný, keďže je ho možné uskutočniť len pri práci s anotátorom výskumnej skupiny KNOT. Dôvodom je potreba prístupu k znalostnej báze anotátora, čo pri externých nástrojoch možné nie je. Entity uložené v zoznamoch chybných a nenájdených entít budú reprezentované triedou `Marking`, ktorej súčasťou budú premenné identifikujúce chybu a riešenie.

### Chyby v nenájdených entitách

Pri týchto entitách došlo k tomu, že anotátor na ich mieste nerozpoznal vôbec žiadnu entitu. V tomto prípade sa nástroj pokúsi zistiť, prečo k tomuto stavu došlo. Zo všetkých entít, ktoré sú v tomto zozname, sa bude zaoberať len tými, ktoré sú obsiahnuté v znalostnej báze nástroja SEC a mali byť teda rozpoznané.

1. **Neznáma kotva** – jedná sa o chybu, kedy SEC danú entitu nemá uloženú v znalostnej báze pod pojmom, pod ktorým je uvedená v dátovej sade. Nástroj musí vyhľadať entitu v znalostnej báze a zistiť, či je v nej takto uvedená. Pre vyhľadanie entity sa použije URI a služba nástroja SEC `get_entities_by_uri`. Konfigurácia k tejto službe je uložená v inštancii triedy `Anotator` a volí sa parametrom metódy `process`. Pri týchto chybách je možné rozšíriť znalostnú bázu pridaním alternatívnych názvov daným entitám.

2. **Rozpoznanie časti entity** – k tejto chybe môže dôjsť v prípade, že SEC pozná entitu pod pojmom, pod ktorým je uvedená v texte, ale rozpozná ju len čiastočne (napr. z *Birmingham Blitz* rozpozná len časť *Birmingham*), alebo rozpozná väčšiu časť textu (napr. namiesto *Barack Obama* rozpozná *President Barack Obama*).

3. **Chyba kontextu** – je chyba, kedy SEC pozná entitu pod daným pojmom, na jej mieste však nerozpozná vôbec nič. To je väčšinou spôsobené chybným kontextom, v ktorom sa entita nachádza. Nerozpoznanie entít v tomto prípade nie je teda úplne chybou anotátora.

## Chybne rozpoznané entity

Jedná sa o rozpoznanie chybnej entity na mieste entity v dátovej sade. Entita vrátená anotátorom sa nezhoduje v odkazoch do znalostných báz alebo typoch entít a na prvý pohľad sa javí ako chybná.

1. **Chyba presmerovania** – V prípade, že obe entity majú k dispozícii URI do DBpédie alebo Wikipédie, sa nástroj pokúsi určiť, či URI uložená v znalostnej báze nie je zastaralá. To uskutoční využitím MediaWiki API, s ktorým bude komunikovať pomocou protokolu HTTP. To vráti výsledok vo formáte JSON, z ktorého nástroj zistí, či sa nejedná o presmerovanie. Ak áno, znamená to, že odkazy reprezentujú tu istú entitu. V opačnom prípade je isté, že entity sú rôzne.

   Pri tejto chybe je potom možným riešením aktualizácia zastaralých URI v znalostnej báze novými URI z dátovej sady. Môže však nastať situácia, že v znalostnej báze nástroja SEC sa nachádzajú obe entity – tzn. entity sú rovnaké, majú ale iné URI, z ktorých jedna je zastaralá. V tomto stave sa jedná o duplicitné entity v KB a riešením je spojenie týchto entít do jednej, pričom sa zachovajú najnovšie odkazy.

   MediaWiki API nemá žiaden limit na počet volaní, je však potrebné dbať na to, aby sa volania neuskutočňovali paralelne.

2. **URI z rôznych domén** – znamená to, že entity neobsahujú žiadne URI z rovnakej domény, a je teda možné, že tieto entity sú zhodné. Nástroj bude schopný medzi sebou mapovať odkazy do znalostnej bázy DBpédie, Wikipédie a Freebase. Pre mapovanie Wikipédie a DBpédie sa taktiež použije MediaWiki API podobne ako pri predchádzajúcej chybe. Mapovanie Freebase na Wikipédiu alebo DBpédiu je možné za využitia Google Knowledge Graph Search API, ktoré vracia na základe poskytnutého Freebase id odpovedajúcu entitu s daným id a k nemu URI z Wikipédie. Komunikácia sa uskutoční pomocou protokolu HTTP a ako odpoveď očakáva výsledok vo formáte JSON.

   Vďaka tejto funkcii bude možné entitám v znalostnej báze pridať ďalšie odkazy, ktoré doteraz nemali. Opäť môže nastať situácia duplicitných entít v KB. V tomto prípade je riešením spojenie týchto entít dokopy, pričom sa zachovajú URI z oboch, keďže sa jedná o odkazy z rôznych domén.

3. **Chyba kontextu** – nástroju SEC je známa entita v dátovej sade, na jej mieste však rozpozná úplne inú. Možné riešenie je potom pridať alternatívny názov k entite z dátovej sady, a to v prípade, že ju SEC nemá uloženú v znalostnej báze pod pojmom z textu.

4. **Entita neznáma nástroju SEC** – po dokázaní, že entity z porovnávanej dvojice sú rozdielne a SEC vo svojej znalostnej báze nemá entitu z dátovej sady, to znamená, že anotátor sa nedopustil veľkej chyby, keďže entitu, ktorú mal nájsť, nemohol poznať. Znalostnú bázu je potom možné rozšíriť o danú entitu.

5. **Možná zhoda** – v tejto situácii nie je možné dokázať, že entity sú rozdielne – tzn. nástroj nedokázal namapovať odkazy do znalostných báz. Môže sa teda jednať o zhodné entity, nie je to ale isté. Znova sa ponúka možnosť rozšíriť znalostnú bázu o odkazy entity z dátovej sady.

# Kapitola 6

# Implementácia

Táto kapitola popisuje implementáciu nástroja podľa návrhu z kapitoly 5. Sú v nej priblížené implementačné detaily hlavných časti nástroja, ktorých implementácia nebola triviálna, a konkrétne použite niektorých technológii z kapitoly 4.

## 6.1 Príprava dátovej sady

Prevod vertikálu do formátu NIF je implementovaný ako jednoduchý automat, ktorý číta vstupný súbor po riadkoch. Je implementovaný v module `vert_to_nif`, ktorý obsahuje funkciu `transfer` s parametrami určujúcimi cestu k vstupnému a výstupnému súboru. Automat sa môže nachádzať v jednom z týchto stavov:

1. `Počiatočný stav` – čítajú sa jednotlivé riadky súboru až pokým nenarazí na riadok so značkou `<doc>` označujúcou začiatok dokumentu. Z tohto riadku sa pomocou regulárneho výrazu získa URI dokumentu a prejde sa do stavu `in_document`.

2. `in_document` – v tomto stave sa pokračuje v čítaní dokumentu, až do prečítania riadku so značkou `<s>` reprezentujúou začiatok vety, po ktorej sa prechádza do nasledujúceho stavu `in_sentence`. Postupným prechodom viet dokumentu vzniká v tomto stave zoznam s vetami dokumentu vo formáte prostého textu s entitami, ktoré im prislúchajú. Po načítaní koncovej značky dokumentu `</doc>` sa do výstupného súboru zapíšu potrebné záznamy vo formátu NIF. Najprv sa zapíše záznam reprezentujúci celý dokument a potom sa pre každú vetu dokumentu vytvorí záznam označujúci časť dokumentu, jednu vetu. Za každým týmto záznamom sa vložia záznamy o entitách v danej vete. Po tejto činnosti sa znovu prejde do stavu `Počiatočný stav`.

3. `in_sentence` – stav, v ktorom sa nástroj nachádza v jednej vete dokumentu. Postupne sa čítajú ďalšie riadky, až do načítania znaku konca vety `</s>`, po ktorom sa uskutoční návrat do stavu `in_document`. Všetky riadky medzi tým sa postupne upravia odstránením entít HTML alebo prípadných značiek HTML. Upravený text sa vkladá do zoznamu, ktorý obsahuje slová danej vety. Za každé slovo je pridaný znak medzery, okrem prípadu, kedy je na následujúcom riadku značka `<g/>` označujúca neprítomnosť medzery. V prípade riadku obsahujúceho entitu sa z neho pomocou regulárnych výrazov získa URI do znalostnej bázy a dĺžka entity určujúca počet slov z danej vety, ktoré reprezentujú entitu, vrátane aktuálneho riadku. Podľa dĺžky sa z aktuálneho zoznamu slov vety uloží text entity a dopočítajú sa jej indexy. Začiatok entity vo vete je dĺžka všetkých slov zoznamu od začiatku až po prvé slovo, ktoré je súčasťou entity.

V prípade, že sa jedná o URI s odkazom do Wikipédie alebo DBpédie, je volaná funkcia `uri_repair`, ktorá odstráni prípadné chyby v odkazoch. Odstraňujú sa prípony *.html* a uskutočňuje sa volanie Mediawiki API pomocou knižnice `requests` a jej funkcie `post`, ktoré má za úlohu zistiť, či odkazy v dátovej sade nie sú zastaralé. Ako parameter je predaný názov danej entity získaný z odkazu a očakáva sa výsledok vo formáte JSON, z ktorého je potrebná položka `revisions`. Ak je obsah tejto položky text označujúci presmerovanie, získa sa nový odkaz a URI v dátovej sade sa aktualizuje. Tým sa zaručí, že odkazy v dátovej sade budú vždy aktuálne.

Každej entite sa ešte pokúsi určiť typ v znalostnej báze nástroja SEC, a to volaním funkcie `process` inštancie triedy `Anotator`. Ako parameter sa predá URI entity, očakáva sa výsledok vo formáte JSON. Ak nie je výsledok prázdny, vyberie sa z neho odpovedajúca entita a uloží sa jej typ.

Pri návrate do stavu `in_document` sa slová vety spoja do jedného súvislého textu vo formáte prostého textu, a táto veta sa vracia spolu s jej entitami.

K prevodu formátu mg4j do NIF je vytvorený modul `mg4j_to_nif` s funkciou `transfer`, ktorý je taktiež implementovaný ako jednoduchý automat. Funguje podobne ako prevod vertikálu do NIF s tým rozdielom, že pracuje s inými značkami zmienenými v návrhu. Entity dátovej sady sú upravené rovnako ako pri prevode vertikálu. Entity nájdené anotátorom sa ale nijak neupravujú, keďže by po tom nebola odhalená chyba zastaralej URI v znalostnej báze. Vo výsledku sú dva súbory, jeden je pôvodná dátová sada a druhý výsledok z anotátora.

## 6.2 Vstup a jeho spracovanie

Čítanie vstupného súboru v usporiadanom formáte NIF sa uskutočňuje po riadkoch a je implementované ako jednoduchý automat, ktorý môže byť v jednom z nasledujúcich stavov:

1. `Počiatočný stav` – nástroj postupne číta riadky zo vstupného súboru, až pokým nenarazí na prázdny riadok oddeľujúci jednotlivé záznamy NIF. Po načítaní tohto riadku prejde do stavu `new_record`.

2. `new_record` – overí sa, či načítaný záznam predstavuje začiatok ďalšieho dokumentu z dátovej sady. Záznam reprezentujúci dokument musí obsahovať text `nif:Context`. Jeden dokument môže byť ale rozdelený na časti, ktoré reprezentujú jednotlivé vety dokumentu. Každý takýto záznam taktiež obsahuje text `nif:Context`, takže je potrebné overiť, že aktuálny záznam skutočne predstavuje začiatok nového dokumentu. K tomu je v knižnici NIFParser vytvorená funkcia `is_part`, ktorá vracia hodnotu *True* v prípade záznamu reprezentujúceho len časť dokumentu. Ak bol načítaný záznam nového dokumentu, prejde sa do stavu `process_document`, v opačnom prípade sa záznam uloží a pokračuje sa stavom `Počiatočný stav`.

3. `process_document` – načítané záznamy celého dokumentu sú predané inštancii triedy `NIFParser`, ktorá spracuje vstup a ako výsledok vráti inštanciu triedy `Dokument`. Po získaní výsledku sa proces pokúsi o zabratie semaforu zdieľaného s procesmi anotovania a po zabratí vloží dokument do zoznamu `documents_to_annotate`. Pokiaľ je v tomto zozname viac než 15 objektov súčasne, prejde sa do stavu `wait_for_space`, inak sa pokračuje stavom `Počiatočný stav`

4. `wait_for_space` – v tomto stave sa proces uspí na jednu sekundu a následne kontroluje veľkosť zoznamu `documents_to_annotate`. Pokiaľ je položiek menej ako 15, proces pokračuje stavom `Počiatočný stav`, inak sa znova uspí a čaká na uvoľnenie zoznamu. Toto obmedzenie existuje z dôvodu šetrenia operačnej pamäte, tak, aby v pamäti neboli zbytočne uložené dokumenty. Čas jedna sekunda bol zistený experimentálne, jedná sa o približný čas pre spracovanie dokumentu procesom anotovania, teda jednej položky zoznamu.

Trieda `NIFParser` poskytuje funkciu `process_records`, ktorá prijíma načítané záznamy a tie postupne spracuje. Prvý záznam reprezentuje vždy samotný dokument, pre ktorý sa vytvorí inštancia triedy `Dokument` a uloží sa jeho URI, prípadne aj text, ak sa v zázname nachádza. Všetky potrebné dáta sa zo záznamov získajú pomocou regulárnych výrazov. Pre každú entitu je vytvorená inštancia triedy `Marking`, ktorá obsahuje jej pozíciu, text, odkazy do KB a prípadný typ. Všetky tieto entity sa uložia do zoznamu `markings`, ktorý je súčasťou triedy `Dokument`.

Pokiaľ je vstupom neusporiadaný NIF, súbor sa načíta celý do operačnej pamäte a tento obsah sa predá triede `NIFParser` na spracovanie. Tá v tomto prípade rozdelí záznamy na záznamy predstavujúce dokument a entity. Pre každý dokument sa potom vytvorí inštancia triedy `Dokument` a prejdú sa všetky dostupné entity. V prípade, že entita prislúcha dokumentu, je k nemu priradená a vložená do jeho zoznamu entít. Po ukončení je vrátený zoznam objektov triedy `Dokument`.

## 6.3 Práca s anotátorom

Proces anotácie priebežne kontroluje zoznam dokumentov pre spracovanie. V prípade prázdneho zoznamu kontroluje premennú `no_more_documents`, ktorú nastavuje proces spracovania vstupu. Ak je jej hodnota *True*, tak sa tento proces končí. V prípade, že má k dispozícii dokument, zaberie semafor a daný dokument zo zoznamu vyberie a spracuje využitím vytvorenej triedy `Anotator`.

Tá disponuje funkciou `process`, ktorej sa ako parameter predajú dáta na spracovanie a informácia o požadovanej službe nástroja SEC. Pre prácu s nástrojom SEC sa využíva už existujúca knižnica `daemon_lib`, ktorá je súčasťou nástroja SEC, a jej trieda `Client`, ktorej sa ako parameter predá cesta k Unix Domain Socket, pomocou ktorého sa s nástrojom SEC komunikuje. Volaním funkcie `letProcessData` inštancie triedy `Client` sa získa výsledok pre danú konfiguráciu.

Pri službe anotovania je výsledok anotátora vo formáte NIF, a teda sa volá funkcia `process_records` triedy `NIFParser`, ktorá výstup z anotátora prevedie na objekt triedy `Dokument`. Dvojica pre porovnávanie sa vloží do zoznamu `documents_to_compare` zdieľaného s procesom komparácie.

## 6.4 Porovnávanie výsledkov

Proces komparácie porovnáva dvojice vyberané zo zoznamu `documents_to_compare`. Taktiež obsahuje premennú `no_more_documents` informujúcu o ukončení procesu po vyprázdnení zoznamu. Pre porovnávanie sa využíva inštancia triedy `Comparator` a jej funkcia `compare_results`, ktorej sú pomocou parametrov predané všetky kritéria zmienené v kapitole 5.6.

Pre každú entitu z dátovej sady sa pokúsi nájsť odpovedajúcu entitu v zozname entít získaných anotátorom. Keďže sú entity usporiadané v zozname za sebou v takom poradí, v akom sa nachádzajú v texte, je pre urýchlenie porovnávania použitá premenná `ofset` určujúca od akej pozície v zozname entít z anotátora sa má pri ďalšej entite z dátovej sady začať vyhľadávať jej dvojica. Porovnáva sa pozícia dvoch entít a v prípade, že začiatočná pozícia entity z dátovej sady je už vyššia než koncová pozícia aktuálne porovnávanej, znamená to, že ďalej v zozname už jej dvojica nebude, entita sa považuje za nenájdenú a hodnota premennej `offset` sa nastaví na pozíciu aktuálnej entity.

Pri zhode je ešte prípadne uskutočnené porovnávanie typu volaním funkcie `match_type` a porovnávanie odkazov do znalostných báz, k čomu slúži funkcia `match_kb`, ktorá ako parametre prijíma dve entity. Následne sa entita vloží do príslušného zoznamu inštancie triedy `EvaluationResult`.

## 6.5 Identifikácia chýb

Pre túto činnosť nástroja je vytvorená funkcia `identify_mistakes`, ktorá je súčasťou triedy `Comparator`. Táto časť nástroja je voliteľná a je ju možné vypnúť.

Pri spracovaní nenájdených entít sa volaním funkcie `get_type` triedy `Marking` vyberú len entity známe nástroju SEC. Entita z dátovej sady, ktorá nemá typ, je pre anotátor neznáma a ďalej s ňou pracovať nemá zmysel. Následne sa použije funkcia `process` triedy `Anotator`, tentokrát s konfiguráciou pre službu nástroja SEC `get_entities_by_uri`, a ako parameter sa predávajú jednotlivé URI entity. Ako výsledok je získaná entita zo znalostnej bázy nástroja SEC vo formáte JSON. Podľa nej sa následne určí, či je táto entita známa pod pojmom uvedeným v dátovej sade. Ak je nástroju SEC známa, nástroj sa pokúsi v entitách vrátených anotátorom zistiť, či nebola rozpoznaná len časť tejto entity.

Identifikácia chýb u chybne rozpoznaných entít začína určením zhody domén odkazov do znalostných báz u oboch entít. V prípade zhody sa použije funkcia `wiki_api_match`, ktorej sú ako parametre predané odkazy do Wikipédie alebo DBpédie. V nej je využitá knižnica `requests` a jej funkcia `post` pre komunikáciu s MediaWiki API. Od API sa požaduje akcia `query`, ktorá slúži pre získanie informácii o stránke s názvom získaným z URI entity z anotátoru. K tomu je ešte predaný parameter `prop=revisions`, ktorý informuje o tom, že chceme informácie o poslednej revízii stránky. Výsledok je vo formáte JSON, z ktorého sa v položke `pages` nachádza informácia o hľadanej stránke. Z nej sa vyberie položka `revisions`, ktorá obsahuje text stránky alebo v opačnom prípade informáciu o presmerovaní. V prípade presmerovania sa pomocou regulárneho výrazu získa odkaz a porovná sa s odkazom entity zo znalostnej bázy. V prípade zhody funkcia vracia hodnotu *True*, a znamená to chybu presmerovania.

V prípade nezhody domén sa uskutoční mapovanie u odkazov do znalostných báz, u ktorých je to možné. V prípade, že je k dispozícii URI Wikipédie z dátovej sady a URI z DBpédie u entity vrátenej anotátorom, alebo naopak, je možné tieto dva odkazy na seba jednoducho namapovať a to porovnaním zhody názvov. V prípade nezhody je ešte použitá znovu funkcia `wiki_api_match`, keďže môže ísť aj v tomto prípade o presmerovanie.

Okrem toho je možné mapovať odkazy znalostnej bázy Freebase na URI z Wikipédie alebo DBpédie. Na to je vytvorená funkcia `freebase_map`, ktorej parameter je Freebase id a odkaz do Wikipédie alebo DBpédie. Aj v nej je využitá knižnica `requests` a jej metóda `get`, a to pre prácu s Google Knowledge Graph Search API. Ako parameter pre API je predaný kľúč v parametri `key` a Freebase id v parametri `ids`. Vo vrátenom výsledku sa musí nachádzať položka `detailedDescription` a v nej položka `url`, s URI Wikipédie,

ktorá sa porovná so vstupným odkazom. V prípade zhody vracia funkcia hodnotu *True*, čo znamená zhodu entít.

V prípade, že sú dostupné len odkazy do iných znalostných báz, je výsledok označený ako možná zhoda, keďže nie je možné tieto odkazy namapovať.

# Kapitola 7

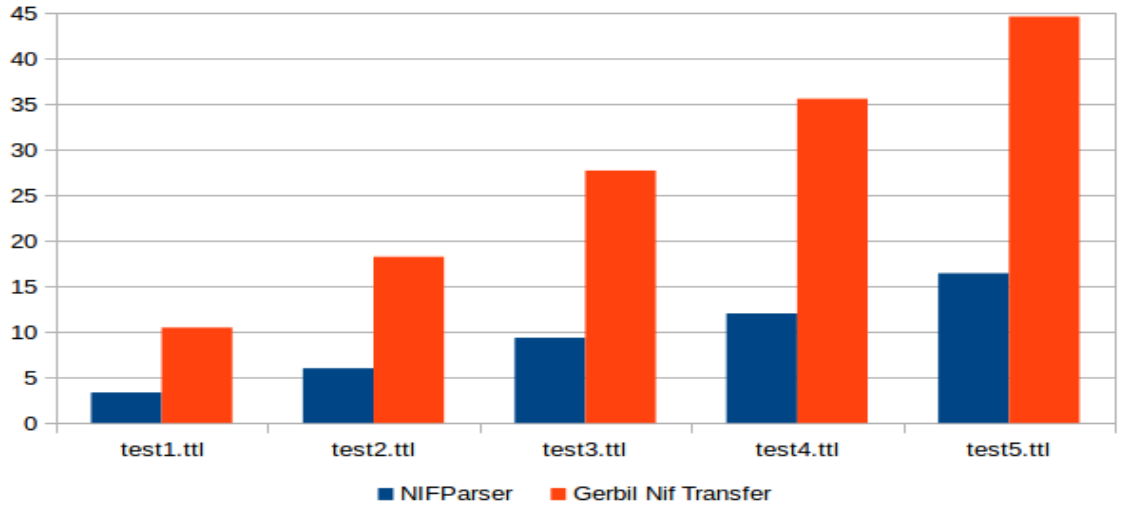
# Testovanie

## 7.1 Rýchlosť spracovania vstupu

V týchto testoch bola porovnaná rýchlosť spracovania vstupných súborov novo vytvorenou knižnicou NIFParser, oproti existujúcemu nástroju Gerbil Nif Transfer. Testovanie sa uskutočnilo na piatich súboroch s rozličným počtom dokumentov v súbore. Počet dokumentov a veľkosť každého z nich sa nachádza v návrhu v tabuľke 2.3. Pre meranie času bol použitý nástroj *time*, z ktorého bol použitý čas reálny.

Pri týchto testoch som na čítanie vstupu použil už mnou vytvorený nifReader na čítanie usporiadaného NIF. To znamená, že sa obom parserom predávali postupne načítavané dokumenty. Gerbil nif transfer bol teda týmto mierne zvýhodnený. Povodne som mal aj verziu, kde sa načítal cely obsah a predal parseru, čo na moju verziu nemalo skoro žiadny vplyv, pri gerbile som ale nemohol skoro nikdy dôjsť do konca. Vždy to časom spadlo kvôli veľkému využitiu pamäte.

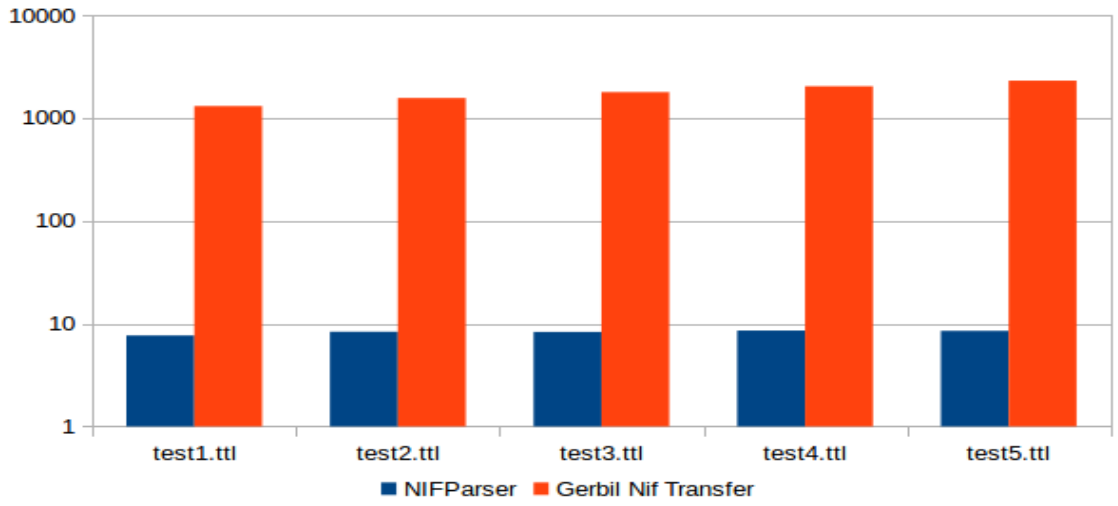
Výsledky testu sú zobrazené v grafe na obrázku 7.1. Na vertikálnej osi je vyznačený čas spracovania v sekundách a na horizontálnej osi sú označené jednotlivé súbory. Stĺpce modrej farby predstavujú nástroj Gerbil Nif Transfer a stĺpce zelenej farby nové riešenie. Z grafu je vidieť, že nová knižnica je v každom prípade približne trikrát rýchlejšia než existujúce riešenie.




Obr. 7.1: Čas spracovania testovacej dátovej sady novým a existujúcim nástrojom

## 7.2 Využitie operačnej pamäte pri spracovaní vstupu

Pre test využitia operačnej pamäte bola použitá rovnaká sada súborov ako pri teste rýchlosti. Pre odčítanie využitej pamäte bol použitý nástroj *top*, z ktorého bola použitá hodnota predstavujúca využitú fyzickú pamäť. Odčítanie sa uskutočnilo pre každý súbor v približne dvoch tretinách doby spracovania daného súboru.




Obr. 7.2: Využitie pamäte pri spracovaní testovacej dátovej sady novým a existujúcim nástrojom

Výsledky testu sú zobrazené v grafe na obrázku 7.2, kde v tomto prípade vertikálna os predstavuje využitie operačnej pamäte v Megabajtoch. Z grafu je jasne vidieť, že nové riešenie kladie značne nižšie požiadavky na pamäť než Gerbil Nif Transfer. Narozdiel od exis-

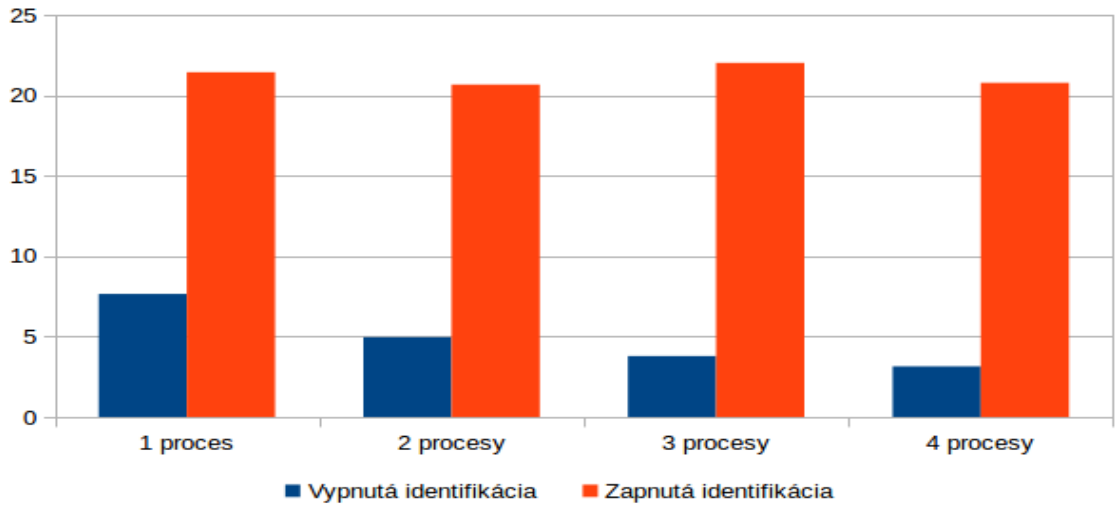
tujúceho riešenia, ktoré používa viac než jeden gigabajt pamäte pre každý súbor, knižnica NIFParser používa pri spracovaní len jednotky megabajtov.

## 7.3 Vypnutie identifikácie chýb

V týchto testoch sa porovnáva rýchlosť spracovania vstupnej dátovej sady so spustenou a následne s vypnutou identifikáciou chýb. Test sa uskutočnil na pripravenej testovacej dátovej sade, ktorá obsahovala päťsto dokumentov, a bol spustený vo variantách s jedným, dvoma, troma a nakoniec štyroma procesmi anotovania.

Výsledok testovania je zobrazený v grafe na obrázku 7.3. Na vertikálnej osi je vyznačený čas spracovania v minutách a na horizontálnej osi výsledky pre jednotlivé počty spustených procesov. Stĺpce modrej farby označujú výsledok bez identifikácie chýb a červené s identifikáciou. Z výsledkov je jasne vidieť, že pri vypnutej identifikácii počet procesov anotovania značne zrýchľuje celé spracovanie. Pri štyroch spustených procesoch sa dosiahlo viac než 100 % zrýchlenie oproti jednému procesu.

Rýchlosť pri zapnutej identifikácii chýb je ale vo všetkých prípadoch približne 21 minút, takže počet procesov anotovania v tomto prípade nehrá rolu. Tento časový rozdiel vzniká kvôli tomu, že identifikácia beží v procese komparácie, z ktorého je vytvorená práve jedna inštancia, a to z dôvodu práce s MediaWiki API, pri ktorej sa nesmie uskutočňovať paralelné volania. Využitím knižnice *time* bol počas testovania meraný čas identifikácie chýb u každého dokumentu a čas kritických časti identifikácie. Po dokončení testov bolo zistené, že priemerne 80 % času identifikácie chýb zaberá práve komunikácia s MediaWiki API. Rýchlosť teda veľmi závisí od počtu entít, u ktorých je potrebné kontrolovať presmerovania URI.




Obr. 7.3: Rýchlosť spracovania s vypnutou a zapnutou identifikáciou chýb

Z výsledkov testov som usúdil, že je vhodné poskytnúť možnosť vypnúť identifikáciu chýb pri porovnávaní. Vo výsledku budú stále dostupné štatistiky a informácie ku všetkým entitám, okrem toho, že u entít nebude uvedená chyba, ktorej sa anotátor dopustil a jej možnosť opravy. Takáto možnosť sa hodí v prípadoch, kedy po porovnaní užívateľ nepožaduje informácie potrebné pre úpravu znalostnej bázy.

## 7.4 Porovnanie s dátovou sadou Wikipédie

Pomocou nástroja bolo uskutočnené porovnávanie anotátora NER s dátovou sadou Wikipédie. Jedná sa o dátovú sadu s celkovou veľkosťou 6,73 GB, obsahujúcou 5 139 138 dokumentov. Táto sada je rovnomerne rozdistribuovaná na 50 serverov výskumnej skupiny KNOT, na ktorých sa porovnávanie uskutočnilo. Bolo spustené porovnávanie so silnou kontrolou pozícii entít, kontrolou zhody odkazov a typov spolu s identifikáciou chýb. V tabuľke 7.1 sú uvedené výsledky spracovania.

Tabuľka 7.1: Výsledky spracovania dátovej sady Wikipédie

| | |
|---|---|
| **Počet dokumentov** | 5 139 138 |
| **Počet entít** | 8 963 532 |
| **Nájdené entity** | 1 043 031 |
| **Nenájdené entity** | 7 634 001 |
| Z toho známych nástroju SEC | 359 733 |
| **Chybne rozpoznané entity** | 286 500 |

Zo všetkých entít v dátovej sade anotátor NER správne rozpoznal 11,64 %. Najväčšiu časť tvoria nenájdené entity, až 85,16 %. Je však potrebné brať do úvahy fakt, že v dátovej sade je veľký počet entít a mnohé z nich sa nenachádzajú v znalostnej báze nástroja SEC, preto je v tabuľke uvedený aj počet entít, ktoré sú súčasťou znalostnej báze nástroja SEC, a tie mali byť rozpoznané. Chybne rozpoznané entity predstavujú tie, u ktorých došlo k nájdeniu inej entity na mieste entity v dátovej sade a tvoria 3,2 % výsledku.

Z 359 733 nenájdených entít existujúcich v znalostnej báze nástroja SEC bola v 209 622 (58,3 %) prípadoch identifikovaná chyba neznámej kotvy, ktorú je možné opraviť pridaním nového alternatívneho názvu entity do znalostnej bázy. U 49 489 (13,76 %) entít nastala chyba rozpoznania časti entity. Vo zvyšných prípadoch sa jednalo o chybu kontextu.

Pomocou 286 500 chybne rozpoznaných entít bolo v znalostnej báze nájdených 2 765 (0,96 %) duplicít, 9 481 (3,3 %) entít so starými URI, 1 054 (0,36 %) entít, ktoré je potrebné rozšíriť o ďalšie odkazy získane z dátovej sady, a 2 257 (0,79 %), ktoré nebolo možné namapovať, je u nich teda možná zhoda a ponúka sa znova možnosť týmto entitám pridať odkazy z dátovej sady. V 154 229 (53,82 %) prípadoch sa jednalo o entitu neznámu nástroju SEC, anotátor ale na jej mieste na základe kontextu rozpoznal nejakú entitu, ktorú mal v svojej znalostnej báze. Znalostnú bázu je možné rozšíriť o tieto entity. U zvyšných entít došlo k chybe kontextu, pri ktorej nástroj mal entitu z dátovej sady v svojej znalostnej báze, na jej mieste však na základe kontextu rozpoznal inú.

Okrem štatistík nástroj samozrejme poskytuje detailné informácie ku každej entite. Vo výpise 7.1 je zobrazená časť výstupného súboru obsahujúceho informácie o chybných entitách. Je vo formáte TSV, kde na každom riadku sú informácie o práve jednej entite. Obsahuje text entity z dátovej sady a text vrátený anotátorom, odkazy do znalostných báz a typy z dátovej sady a anotátora, odkaz na dokument, kontext, v ktorom sa entita nachádzala v texte, a identifikovanú chybu spolu s možným riešením. V prípade 1. entity je uvedená chyba REDIRECT MATCH označujúca chybu presmerovania a riešenie UPDATE WIKIPEDIA URI IN KB, ktoré hovorí, že URI z Wikipédie v znalostnej báze je potrebné aktualizovať na novšiu verziu z dátovej sady.

```
Sharm el Sheikh Sharm el Sheikh http://en.wikipedia.org/wiki/
  Sharm_el-Sheikh|http://dbpedia.org/resource/Sharm_el-Sheikh|
  http://www.freebase.com/m/0266kj|http://www.geonames.org
  /349275  location  http://en.wikipedia.org/wiki/
  Sharm_El_Sheikh    the projects were identified in a large
  stakeholders conference in Sharm el Sheikh during 2004 and
  fine-tuned afterwards.   http://en.wikipedia.org/wiki/
  South_Sinai_regional_development_programme REDIRECT MATCH
  UPDATE WIKIPEDIA URI IN KB
Kulin Kulin http://en.wikipedia.org/wiki/Shire_of_Kulin|http
  ://dbpedia.org/resource/Shire_of_Kulin|http://www.freebase.
  com/m/026bhln|http://www.geonames.org/7839596  location  http
  ://en.wikipedia.org/wiki/Ban_Kulin  person  Kulin [52] Kulin'
  s rule was marked the start of a controversy http://en.
  wikipedia.org/wiki/South_Slavs  WRONG ENTITY  CONTEXT PROBLEM
```

Výpis 7.1: Výpis výstupného súboru s chybnými entitami

Keďže sa jedná o dátovú sadu, ktorá nebola kontrolovaná manuálne, môže táto sada obsahovať rôzne chyby. Veľa chýb kontextu vzniká práve z dôvodu, že sa jedná o webové stránky, u ktorých po vertikalizácii mohli ostať rôzne artefakty (napr. z infoboxu). Vo výsledku sú teda podstatné hlavne entity, u ktorých boli odhalené chyby, a možnosť opravy týchto chýb v znalostnej báze. Percentuálna úspešnosť nástroja môže byť kvôli chýbam v sade dosť skreslená, preto sa na takéto porovnanie používajú kontrolované dátové sady.

# Kapitola 8

# Záver

Cieľom tejto práce bolo navrhnúť a implementovať nástroj umožňujúci porovnávanie anotačných nástrojov oproti rôznym dátovým sadám. Na začiatku práce boli priblížené anotačné nástroje spolu s nástrojom SEC, dostupné riešenia pre porovnávanie, ich nedostatky a dostupné dátové sady. Ďalšie časti sa zaoberali návrhom nového nástroja a jeho samotnou realizáciou.

Navrhnutý nástroj sa podarilo úspešne implementovať a otestovať. Výsledný nástroj umožňuje prácu s dátovými sadami vo formáte NIF a poskytuje knižnice pre ich vytvorenie. Je možné si zvoliť jeden z piatich anotátorov, ktoré sú dostupné prostredníctvom nástroja SEC, v ktorom pre nich boli vytvorené rozhrania pre komunikáciu. Do budúcna sa ponúka rozšírenie pridať rozhrania aj pre ďalšie anotátory.

Pri porovnávaní je možné si voliť medzi silnou a slabou kontrolou pozície entít, kontrolou typov a odkazov do znalostných báz. Vo výsledku nástroj poskytuje štatistické informácie ale aj obšírnejšie informácie ku všetkým entitám z dátovej sady. Dôležitou časťou nástroja je taktiež identifikácia chýb určujúca chyby, ktorých sa anotátor dopustil, a k tomu aj možné riešenia ich opravy. Pri testovaní bolo zistené, že identifikácia chýb predlžuje celý proces porovnávania, preto nástroj umožňuje túto funkciu vypnúť. V budúcnosti by bolo vhodné rozšíriť možnosti identifikácie chýb pridaním mapovania odkazov do ďalších KB, čím by sa odstránili nepotvrdené zhody entít. Ďalej aj pridať identifikáciu dôvodu rozpoznania nesprávnej entity v rámci kontextu a upraviť spôsob mapovania odkazov do Wikipédie pre urýchlenie celej identifikácie.

Testovanie rýchlosti spracovania vstupu ukázalo, že nový nástroj kladie značne nižšie požiadavky na operačnú pamäť a je približne trikrát rýchlejší než dostupné riešenia. Tento fakt je dôležitý hlavne pri práci s veľkými dátovými sadami.

Pomocou vytvoreného nástroja sa uskutočnilo porovnanie anotátora NER s dátovou sadou Wikipédie, pri ktorom bolo identifikovaných približne dvestotisíc možných opráv v znalostnej báze. V blízkej dobe bude uskutočnené porovnávanie s dátovými sadami Wikilinks, ClueWeb a CommonCrawl a bude potrebné odstrániť prípadne vzniknuté chyby.

# Literatúra


[1] Belshe M.; BitGo; Peon R.; a ďalší. *RFC 7540*, [online, cit. 2017-04-27].
URL https://tools.ietf.org/html/rfc7540

[2] Centrum zpracování přirozeného jazyka. *Popis vertikálů*, [online, cit. 2017-04-27].
URL https://nlp.fi.muni.cz/cs/PopisVertikalu

[3] Fielding, R.; UC Irvine; Gettys J. a ďalší. *RFC 2616*, [online, cit. 2017-04-27].
URL https://tools.ietf.org/html/rfc2616

[4] A Kenneth Reitz Project. *Quickstart*, [online, cit. 2017-04-27].
URL http://docs.python-requests.org/en/master/user/quickstart

[5] Ecma International. *The JSON Data Interchange Format*, 2013, [online, cit. 2017-04-27].
URL http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

[6] Jan Doležal. *SEC API*, 2014, [online, cit. 2017-04-27].
URL http://sec.fit.vutbr.cz/sec_api.pdf

[7] Cornolti, M.: *BAT-Framework v0.1: A Quick Reference*, 2013, [online, cit. 2017-04-27].
URL http://acube.di.unipi.it/wp-content/uploads/2013/01/BAT-Framework-0.1-reference.pdf

[8] Cornolti, M.; Ferragina, P.; Ciaramita, M.: A Framework for Benchmarking Entity-Annotation Systems. In *Proceedings of the International World Wide Web Conference (WWW) (Practice & Experience Track)*, 2013.

[9] Hellman, S.: *NLP Interchange Format (NIF) 2.0*, [online, cit. 2017-04-27].
URL http://persistence.uni-leipzig.org/nlp2rdf/specification/core.html

[10] Martelli, A.; Ravenscroft, A.; Holden, S.: *Python in a Nutshell*. O'Reilly Media, tretie vydání, 2017, ISBN 978-1-4493-9292-5.

[11] MediaWiki: *API:Main page*, [online, cit. 2017-04-27].
URL https://www.mediawiki.org/wiki/API:Main_page

[12] Panda, P.: *Common Crawl - Free Database Of The Entire Web, Competition For Google*, 2013, [online, cit. 2017-04-27].
URL http://thetechpanda.com/2013/01/25/common-crawl-free-database-of-the-entire-web-competition-for-google/#.U9l25IBdVIc

[13] Röder, M.; Usbeck, R.; Ngomo, A.-C. N.: *GERBIL – Benchmarking Named Entity Recognition and Linking Consistently*, 2016, [online, cit. 2017-04-27].
URL http://www.semantic-web-journal.net/system/files/swj1577.pdf

[14] Starr, B.: *A Layman's Visual Guide To Google's Knowledge Graph Search API*, 2016, [online, cit. 2017-04-27].
URL http://searchengineland.com/laymans-visual-guide-googles-knowledge-graph-search-api-241935

# Prílohy

# Príloha A

# Obsah DVD

- */src* – zložka obsahujúca zdrojové kódy nástroja
- */SEC* – zložka obsahujúca nástroj SEC
- */corpproc* – zložka obsahuje vertikalizátor a nástroj pre deduplikáciu dátovej sady. Oba nástroje sú potrebné pre prípravu dátovych sád.
- */manual* – obsahuje návod na spustenie nástroja
- */doc_pdf* – zložka s touto technickou správou vo formáte PDF
- */doc_tex* – zložka so zdrojovými kódmi tejto technickej správy v LaTeX