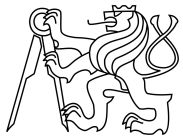
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Metoda pro sumarizaci a hodnocení významnosti informací na Webu dat
Student:	Bc. Marek Filteš
Vedoucí:	Ing. Milan Dojčinovski
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Web dat (angl. Web of Data) obsahuje množství informací, které popisují charakteristiky entit z různých domén. Je však obtížné posoudit, které z těchto informací jsou důležitější než ostatní. Hlavním cílem této práce je navrhnout a naimplementovat metodu pro určení důležitosti informací, která bude založena na datech z Wikipedie a DBpedie. Dalším cílem je navrhnout a naimplementovat webovou aplikaci "prohlížeč" pro sumarizaci a prohlížení entit.
Pokyny:
- Seznamte se s existujícími metodami pro sumarizaci entit.
- Navrhněte metodu pro sumarizaci informací o entitách založenou na datech z Wikipedie a DBpedie. Jako hlavní zdroj dat použijte dataset DBpedia abstracts a ve Wikipedia článcích analyzujte výskyt důležitých informací.
- Ve vybraném prostředí naimplementujte webovou aplikaci pro prohlížení a sumarizaci entit. Aplikace má zobrazit top-N nejdůležitějších informací pro danou entitu.
- Ověřte funkčnost aplikace na vybrané podmnožině entit z DBpedie a proveďte evaluaci metody.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 7. dubna 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra softwarového inženýrství

Diplomová práca

Metóda pre sumarizáciu a hodnotenie významnosti informácií na Webe dát

Bc. Marek Filteš

Vedúci práce: Ing. Milan Dojčinovski

8. mája 2017

Poďakovanie

Chcem sa poďakovať vedúcemu tejto diplomovej práce Ing. Milanovi Dojčinovskému za jeho cenné poznatky, rady a pripomienky, ktorými mi bol nápomocný pri jej tvorbe.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 8. mája 2017 .....

Odkaz na túto prácu

Filteš, Marek. *Metóda pre sumarizáciu a hodnotenie významnosti informácií na Webe dát.* Diplomová práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Táto práca sa zaoberá sumarizáciu entít sémantického webu. Najprv sa rieši otázka informácií, hodnotenia miery dôležitosti informácií ako aj všeobecne sumarizácie enít. Prechádza sa k sumarizácií entít sémantického webu. V praktickej časti sa zaoberá návrhom modelu a implementáciou nástroja pre sumarizáciu entit na základe datasetu DBpedia abstracts. Vygenerovaná báza znalosti je integrovaná vrámci implementácie webového prehliadača.

Klíčová slova sumarizácia entít, dôležitosť informácií, sémantický web, DBpedia

Abstract

This work deals with entity sumarization of semantic web. Firstly, the issue of information, evaluation the importance of information as well as the general summation of entities. It goes to entity sumarization of semantic web entities. The practical part deals with design of the model and implementation of the entity summary tool based on the DBpedia abstracts dataset. The generated knowledge base is integrated within the implementation of the web browser.

Keywords entity summarization, importance of information, semantic web, DBpedia

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Web dát (angl. Web of Data) obsahuje veľké množstvo informácií, popisujúce charakteristiky entit z rôzných domén. Je zložité vyhodnotiť, ktoré informácie sú dôležitejšie než ostatné. Problém dôležitosti informácií, do určitej miery, limituje použitie zdrojov prepojených dát (angl. Linked Data) uložených v databázach ako sú DBpedia, Freebase, Wikidata a ďalšie. DBpedia ako jedná z najdlhšie existujúcich, aktuálne vo verzií 2016-04, obsahuje informácie pre približne 6 milónov zdrojov (angl. resource) a 1,3 miliárd základných faktov (RDF tripletov)[7]. Každý zdroj preto obsahuje v priemere približne 216 RDF trojíc [7]. Hľadanie informácií človekom v takýchto zdrojoch, si vyžaduje veľké množstvo času. Tento stav nie je vhodný. Vo všeobecnosti má užívateľ pri hľadaní niekoľko možných cieľov. Napríklad chce rýchlo nájsť zdroj, ktorý obsahuje informáciu, ktorú potrebuje. Zároveň potrebuje preskočiť zdroje, ktoré pre jeho hľadanie niesu relevatné. Medzi ďalší možný cieľ patrí zoskupenie zdrojov na základe ich kľúčových informácií. V rámci týchto a iných možných cieľov hlďania, je bez stanovenia dôležitosti informáciam, prehľadavanie zdrojov nefektívne. Preto sa v tejto práci budeme venovať návrhu metódy pre satnovanie doležitosti informáciam a ich prehliadaniu.

Hlavným cieľom tejto práce je poskytnuť používateľovi n najdôležitejších informácií jednotlivých zdrojov DBpedie na základe príslušnosti do určitej skupiny entít. Tieto informácie chceme získať z neštrukturovanej časti textu Wikipedie, tzv. abstraktu. Zo získaných vzťahov chceme vytvoriť bázu znalosti. Naše vysledky budu pre používateľa prehľadné pomocou webovej aplikácie, ktorá bude používať vytvorenú bázu znalosti.

Preto sa budeme v tejto práci venovať riešeniu problému, nazývaného sumarizacia entít (angl. entity summarization). V rámci sumarizácie entit budeme identifikať mieru dôležitosti jednotlivých informácií entít. Pokúsime sa na základe existujúcich informácií vybudovať bázu znalosti, pomocou ktorej budeme môcť inetrepretovať dôležité informácie. V prvej, teoretickej, časti práce sa zoznámime s existujúcimi riešeniami sumarizácie entít. Popíšeme základne pojmy, ktoré potrebujeme pochopiť. Identifikujeme hlavné technologie,

potrebné pre sumarizáciu entít, ktoré budeme používať. V časti návrhu si popíšeme proces návrhu metódy. Ná základe metódy vytvoríme spustiteľný nástroj. Pomocou nástroja si vytvoríme bázu znalosti, ktorú použijeme pri tvorbe webového prehliadača. V časti porovnania a vyhodnotenia, získane výsledky miery dôležitosti porovnáme voči výsledkom iných riešení. V závere zhodnotíme dosiahnuté výsledky práce, možne použitie tohto riešenia, nedostatky a prípadne vylepšenia.

Výsledkom práce sa stane nástroj pre sumarizáciu entít a identifikáciu dôležitých informácií, báza znalosti ako aj webová aplikácia pre prehliadanie dôležitých informácií zdrojov DBpedie.

Kapitola 1

Teoretická časť

Predtým ako sa dostaneme k hlavnej časti práce si musíme rozobrať potrebné predpoklady pre našu sumarizáciu entít. Rozoberme teoretické vedomosti. Ako stanoviť mieru dôležitosti informácií. Pozrime sa na to čo znamená sumarizácia entít. Čo v rámci jej riešenia musíme pochopiť. S akými technológiami budeme musieť pracovať. Zároveň si prezrime aktuálne dostupné riešenia sumarizácie entit a stanovenia miery dôležitosti informáciam.

1.1 Informácia

Informácie sú hnacou silu spoločnosti. Informácie sa už od dávných dôb predávali medzi ľudmi v rámci vzájomnej komunikácií. Vďaka výmene informácií sa ľudstvo vyvinulo k technologický vyspelej civilizácií. Pozrime sa na pojem informácia v širšom a taktiež v užšom zmysle, potrebnom pre našu prácu.

Pojem informácia je veľmi široký a ťažko definovateľný. Predná definícia pojmu informácia neexistuje. Slovo informácia má pôvod v latinčine, kde znamená *„uvádzať do tvaru"*. Uveďme si aspoň dve základne prístupy chápania informácie. Existuje samozrejme viacero možnosti chápania informácií.

- Informácia ako oznámenie. Ide o komunikačný prostriedok, ktorý má význam z hľadiska príjemcu.
- Informácia ako správa. Súčasť technického prenosu.

Potrebu skúmať informácie aj z hľadiska technického, prinieslo až využívanie elektrickej energie. Z počiatku sa však vedci venovali viac problému prenosu, než otázke čo znamená informácia. Až publikácia *"Přenos informací"* od R. Hartleyho z roku 1928, prinesla jednu zo základných myšlienok technického chápania informácie. V základe popisuje prenos správy ako postupnosti symbolov. Väčšina neskorších prác a prístupov technického chápania informácií je založená na tomto princípe.

Keďže informácia, v technickom chápaní, vznikla abstrakciou, nemá hmotnú podobu. Vždy je ale spojená, s určitým fyzikálnym pochodom. Napríklad signálom. Vyjadrením informácie, ako postupnosti znakov, je správa. Správa môže byť textom, číslom a pod. Všetky správy majú svojú skladbu (syntaxi), obsah (sémantiku) a dôležitosť (pragmatickosť):

- Sémantický obsah vyjadruje význam správy. Správy s rovnakým množstvom informácií je možne zapísať v rôznych jazykoch.
- Pragmatický obsah vyjadruje dôležitosť (významnosť, užitočnosť) a prioritu pre príjemcu. Je zrejme, že správa *"Peter je doma."* má inú dôležitosť pre jeho otca a inú pre osobu, ktorá ho nepozná.
- Syntaktický obsah je vzájomné usporiadanie znakov. Syntaktický obsah nie je závislí na existencií sémantického, či pragmatického obsahu. Sémantický obsah správa získava smerom k objektu a pragmatický smerom k príjemcovi.

Syntaktické usporiadanie znakov patrí pod kvantitatívnu vlastnosť informácie. Sémantický a pragmatický obsah spolu vytvárajú kvalitatívnu vlastnosť informácie. [8]

Je podstatné rozlíšiť tri pojmy: *dáta*, *informácia* a *znalosť*. Dáta predstavujú rozpoznateľné, vzájomne oddelené znaky. Tie zvyčajne bývajú uložené v pamäti. Systém, ktorý ich dokáže akceptovať a spracovať, je im schopný priradiť určitý význam. V takom prípade je ich možné, v rámci daného systému, považovať za informácie. Pokiaľ majú dané informácie pre systém organizačný význam, je možne ich nazvať znalosťami. [9]

1.2 Relevatnosť informácií

Najskôr sa pozrime ako popisujú problém relevantnosti autori prác o medziľudskej komunikácií. Podľa Jany Hoffmannovej sa človek snaží predávať adresátovi len relevantné informácie. Príjemcovia venujú svoju pozornosť, len takým informáciám, ktoré pokladajú za podstatné a v daný moment použiteľné. V rámci vnímania stále očakávame prospešné informácie. Ide o očakávanie, ktoré slúži ako kritérium, podľa ktorého vyberáme z množiny alternatívnych interpretácií, tú ktorá je najpodstatnejšia a najzávažnejšia. [10] V tomto tvrdení sa síce jedná o medziľudskú komunikáciu, avšak podobné princípy hodnotenia platia všeobecne pri vyhľadávaní informácií, pretože informácie majú hlavný význam pre človeka. Človek má svoje očakávania, na základe ktorých hodnotí relevantnosť získaných informácií.

Informácie môžu mať určitý význam jedine v definovanom kontexte. Podľa Roberta Stalnakera, je kontext jednoducho definovateľný ako určitý celok dostupných informácií. Tieto informácie sú dostupné všetkým členom konverzácie. Kontextová množina predstavuje možné situácie, ktoré sú kompatibilné

	Splnenie špecifikácií	Naplnenie očakávaní používateľa
Kvalita produktu	Spoľahlivé informácie: charakteristiky dodaných informácií spĺňajú štandard informačnej kvality.	Užitočné informácie: charakteristiky dodaných informácií spĺňajú používateľove potreby.
Kvalita služby	Dôveryhodné informácie: transformácia dát na informácie spĺňa štandardy.	Použiteľné informácie: transformácia dát na informácie spĺňa používateľove potreby.

Tabuľka 1.1: Aspekty modelu PSP/IQ [5]

s dostupným celkom informácií. Na základe toho môžeme reprezentovať kontext ako množinu určitých svetov. Tieto svety predstavujú otvorené možnosti vzhľadom k tomu, čo je pevne dané. [11]

Mieru relevancie novej informácie pre jednotlivca musíme hodnotiť pomocou, určitého zlepšenia, ktoré je výsledkom získania novej informácie v danom kontexte. Ľudia vymedzujú mieru relevancie, spôsobom, kde je myšlienkový predpoklad relevantný v kontexte, len ak má kontextuálny účinok, na tento kontext. Psychické procesy, zahrňajúce úsilie pre dosiahnutie relevancie predstavuje negatívny efekt, vzhľadom ku efektu dosiahnutej relevancie. Myšlienkový predpoklad je pre človeka relevantný, v danej dobe, ak je relevantný aspoň v jednom kontexte. [12]

1.2.1 Relevatnosť informácií sémantického webu

Problém dôležitosti informácií sémantického webu (viac o sémantickom webe nájdete v kapitole 1.4.1) je zložitejši a je možne ho riešiť z rôzných hľadísk. Pokúsime sa postupne prepracovať k čo možno najvhodnejšiemu riešeniu, ako chápať relevatnosť informácií v prostredi sémantického webu.

Beverly K. Kahn a kolektív v práci "Information quality benchmarks: product and service performance" popisujú model hodnotenia informačnej kvality. Kvalitu informácií popisujú ako vhodnosť (angl. fitness) pre používateľa. Extistujú však aj ďalšie konkrétnejšie pohľady. Patrí sem napríklad: dokonalosť, splnenie špecifikácie alebo naplnenie očakávani používateľa. Vo svojom modeli PSP/IQ (product and service performance model for information quality) pristupujú ako k splneniu špecifikácie a naplnenie očakávaní používateľa. [5] Základné dimenzie modelu sú zobrazené v tabuľke 1.1.

Pomocou modelu PSP/IQ poskytujú popis dimenzií kvality informácií. Tento popis je nápomocný k lepšiemu pochopeniu požiadaviek pre poskytovanie vysoko kvalitných informácií. Tento popis je v tabuľke 1.2. Tieto dimenzie poukazujú na vysokú mieru komplexnosti problému relevancie informácií.

Relevancia informácií v prostredí sémantického webu môže byť ďalej určená pomocou hodnotenia kvality dát. Kvalita dát so zameraním na kvalitu datového produktu. Takáto kvalita môže byť posudzovaná v rôznej úrovni granularity [13]:

Dimenzia	Popisuje
Dostupnosť	Do akej miery sú informácie k dispozícii alebo ľahko a rýchlo vyhľadateľné.
Primerané množstvo informácií	Do akej miery je objem informácií vhodný pre danú úlohu.
Dôveryhodnosť	Do akej miery sú informácie považované za pravdivé a dôveryhodné.
Úplnosť	Do akej miery informácie nechýbajú a majú dostatočnú šírku a hĺbku pre danú úlohu.
Stručné zastúpenie	Do akej miery sú informácie zjednodušene reprezentované.
Dôsledné zastúpenie	Do akej miery sú informácie prezentované v rovnakom formáte.
Jednoduchá manipulácia	Do akej miery sa s informáciami ľahko manipuluje a uplatňujú na rôzne úlohy.
Bezchybovosť	Do akej miery sú informácie správne a spoľahlivé.
Interpretovateľnosť	Do akej miery sú definície jasné a informácie v príslušných jazykoch, symboloch a jednotkách.
Objektivita	Do akej miery sú informácie nezaujaté, bezprecedentné a nestranné.
Relevancia	Do akej miery sú informácie použiteľné a užitočné pre danú úlohu.
Reputácia	Do akej miery je informácia odporúčaná, pokiaľ ide o jeho zdroje alebo obsah.
Bezpečnosť	Do akej miery je prístup k informáciám primerane obmedzený, aby sa zachovala jeho bezpečnosť.
Včasnosť	Do akej miery sú informácie dostatočne aktuálne pre danú úlohu.
Pochopiteľnosť	Do akej miery sú informácie ľahko pochopiteľné.
Pridaná hodnota	Do akej miery sú informácie prospešné a poskytujú výhody z ich využívania.

Tabuľka 1.2: Dimenzie kvality informácie [5]

- Úroveň objektu (inštancie). Hodnotením relevancie inštancií tried v RDF databáze.
- Úroveň dokumentu. Pomocou analýzy obsahu a štruktúry prepojení.
- Úroveň podgrafov. Pomocou analýzy obsahu založenej na ontológií v kontexte výsledkov dotazu a následne výpočtom dôveryhodnosti na základe kontextu.

Pojem relevancia je často chápaný s trocha odlišným významom. Viacerí autori ho preto rozoberajú vo svojich prácach a snažia sa ho zadefinovať. Ide o komplexný a zároveň multidimenzionálny pojem. V oblasti informatiky sa objavuje určitý súlad v jeho chápaní. Kagolovsky Y. uvádza, že revelancia je vo všeobecnosti rozdelená do dvoch hlavných kategorií:

- Tématická relevancia. Je objektívna a týka sa terminológie.
- Používateľský orientovaná relevancia. Je subjektívna, závisi na potrebách používateľa.

Z hľadiská úrovne, medzi ktorými informácie prechádzajú pri ich získavaní, sa relevancia dá riešiť na viacerých z nich. Nižšie úrovne rieššia interakcie s informačným systémom. Naproti tomu vyššie úrovne zabezpečujú interakcie používateľa. V rámci vyšších úrovni existujú ďalšie aspekty. Patrí sem:

- Kognitívny aspekt
- Afektívny aspekt
- Situačný aspekt
- Kontextový aspekt

[14][15] Kognitívny aspekt relevancie hovorí o zložitosti a množstve biochemických procesov potrebných k dosiahnutiu pochopenia informácií používateľom. V prípade, že rozoznáva málo relevatné informácie je táto kognitívna záťaž vyššia. Vyššia kognitívna záťaž predstavuje negatívny stav pri vyhľadávaní. [16] Afektívny aspekt relevancie je vzťahom medzi zámerom, cieľom a motiváciou používateľa a zdroja. Situačný aspekt relevancie predstavuje vzťah medzi situáciou, úlohou alebo problémom a zdrojom. [14] Kontextový aspekt bol spomenutý vyššie. Hovori o množine svetov v rámci dostupného celku informácií.

R. Lachica s kolektívom ďalej, za účelom jasnejšieho chápania problému relevance, popisujú definície postulátov pre pojmy:

Kvalita Odzrkadľuje vnútornú hodnotu informácie. V prípade, že sú informácie nespoľahlivé alebo nepochopiteľné sú pokladané za nekvalitné. Naďalej môžu byť označené za relevatné alebo dôležité.

Relevancia Odráža hodnotu informacie z potrieb používateľa. Ide o vnímanie používateľom.

Dôležitosť Odzrkadľuje hodnotu informácie zo širšieho hľadiska. Z pohľadu ľudí, ktorý majú poznatky o tom čo môže používateľ potrebovať. Tieto poznatky nemusí mať samotný používateľ.

[14]

1.3 Sumarizácia entít

Sumarizácia entít predstavuje aktuálne rozsiahlú oblasť výskumu. Ide o všeobecný problém analýzy obsahu entít. Keďže väčšina informácií je reprezentována pomocou textov, je najčastejšou sumarizáciu, sumarizácia textov. Dôvodom, prečo sa tomuto problému začala venovať pozornosť, je nárast informovanosti. Ide predovšetkým o veľke množstvo informácií zdieľaných pomocou webových technológií. Preto je sumarizácia entít naobšírnejšie spracovaná z pohľadu sumarizácie dokumentov na webe. Pozrime sa na základy sumarizácie entít na webe ako aj na konkretnejší problém sumarizácie entít sémantickéhi webu.

Sumarizáia dokumentov webu môže byť chápana, ako text ktorý vznikol z jedného alebo viacerých textov. Poskytuje podstatné informácie z pôvodných textov. Nemal by však byť dlhší než je polovica celkového textu. Hlavným cieľom sumarizácie je prezentácia hlavných ideí v dokumente vo výrazne menšom rozsahu. Ak by mali všetky elementárne časti textu rovnakou mieru dôležitosti,

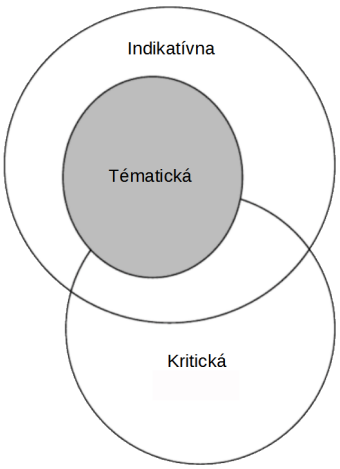
nebola by sumarizácia efektívna. V praxi je však väčšina textov rozdelená do určitých logických časti, pričom je vždy možné posúdiť ich významnosť. Výsledok sumarizácie je znovupoužitím časti textu (vety, slovné spojenia, slová a ďalšie), prípadne ich modifikovanou podobou. Existuje niekoľko typov sumarizácii, ktoré je možné definovať z rôznych nezávislých hľadísk:

Z hľadiska formy reprezentácie výsledku existuje extrakčná a abstrakčná sumarizácia.

- Extrakčná sumarizácia. Vo výsledku sumarizácie sú priamo vety z pôvodného textu.
- Abstrakčná sumarizácia. Výsledok nemusí obsahovať vety z pôvodného textu. Môže ich parafrázovať, zovšeobecňovať prípadne konkretizovať. Vyžaduje pokročilejšie metódy spracovania.

Z hľadiska účelu existuje indikatívna, tématická a kritická sumarizácia entít. Ich vzajomný vzťah je znázornený na obrázku 1.1.

- Indikatívna sumarizácia (angl. indicative summary) identifikuje o čom text informuje. Poskytuje skrátenu verziu obsahu.
- Tématická sumarizácia (angl. topic-oriented summary) sa zamierava na požadované témy čitateľa. Takáto sumarizácia odzrkadľuje pohľad čitateľa.
- Kritická sumarizácia poskytuje hlavné zistenia systematického prieskumu. Výsledkom je posudok o kvalite prieskumu a platnosti výkladu poznatkov.



Obr. 1.1: Vzťah medzi indikatívnou, tématickou a kritickou sumarizáciou [1]

Samotný proces sumarizácie môže obsahovať rôzne čiastkové procesy. Vo všeobecnosti je možné tieto procesy zaradiť do 4 základných skupiín:

1. Extrakcia - identifikácia podstných častí textu.
2. Abstrakcia - preformulovanie časti na nové pojmy
3. Fúzia - kombinovanie extrahovaných častí
4. Kompresia - eliminovanie nepotrebných častí

[17] [1]

Od kedy bola publikovaná prvá metóda automatickej sumarizácie textu, vzniklo množstvo ďalších. Sumarizačné metódy môžeme rozdeliť do troch skupín:

1. Klasické metódy:
   - Heuristické metódy. Predstavujú najstaršie sumarizačné metódy. Jednou z nich bola metóda frekvencie termov vytvorená z roku 1958. Jej hlavnou myšlienkou je výber najčastejšie sa vyskytujúcich slov textu.
   - Štatistické metódy. Medzi typickú štatistickú sumarizáciu patrí Bayesova klasifikácia kde sa klasifikátor snaží o každej vete dokumentu rohodnúť či patrí do výsledku, alebo nepetrí.
2. Metódy so spracovaním prirodzeného jazyka:
   - Metódy využívajúce súvislosti v texte. Patrí sem napr. metóda lexikálnych reťazcov.
   - Metódy modifikujúce pôvodný text.
3. Ďalšie matematické metódy:
   - Grafové metódy. Tieto metódy sú založené na algoritme PageRank, prípadne jeho modifikáciach TextRank a LexRank.
   - Algebrické metódy. Patrí sem napr. metóda latentnej sémantickej analýzy (LSA), ktorej princípom je rozklad matice na singulárne hodnoty.

[18]

Medzi najčastejšie metódy hodnotenia poradia dokumentov na webe patrí PageRank. PageRank vivinul Larry Page na Stanfordovej univerzite. PageRank patril ku kľúčovým hodnoteniam výsledov vo vyhľadávači Google. Základnou myšlienkov hodnotenia dokumentu je počet iných dokumentov, ktoré na neho odkazujú. Týmto smerom mu pridávajú dôležitosť. Na druhej strane

je jeho dôležitosť zníženia počtom vlastných odkazov na iné dokumenty. Definícia PageRanku: nech u je webová stránka a nech F_u je množina stránok u. Ďalej nech B_u je množina stránok odkazujúcich na u a $N_u = |F_u|$ je množina odkazov z stránky u [2]. Potom:

$$R(u) = c \sum_{v \in B(u)} \frac{R(v)}{N(v)}$$

Obr. 1.2: Definícia PageRank [2]

Jedným z modelov, ktorý implementuje PageRank je model náhodneho surfera. Surfista sa pohybuje medzi jednotlivými uzlami a s rovnakou pravdepodobnosťou vyberá, ktorú hranu na prechod použije. Zároveň s malou pravdepodobnosťou, skočí do náhodného uzla. Poradie uzlov sa získava pomocou stacionárneho rozloženia Markovského reťazca. Uzol s vyššou pravdepodobnosťou dosahu surfera je hodnotený vyšším skóre. Týmto spôsobom sa predpokladá, že najvyššie zaradené uzly zachytia hlavné témy pôvodných dát. [2]

1.4 Základné technológie sémantického webu

V tejto časti sa zoznámime s podstatnými technológiam pre realizáciu našej metódy sumarizacie entít. Pohybujeme sa vo webových technológiach. Presnejšie technológiach spojených s oblasťou sémantického webu.

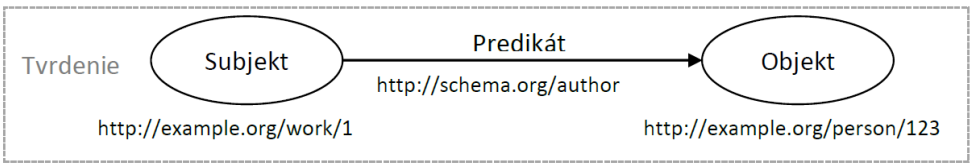
1.4.1 Sémantický web

Sémantika je náuka o význame jednotlivých slov, morfému a iných znakov, prípadne tiež ich vzťahu ku skutočnosti, ktorú označujú. Slovo sémantika ma pôvod v gréckom slove "séma", ktoré znamená význam alebo nosič významu. [19] V roku 2001 vyslovil Tim Berns-Lee myšlienku sémantického webu. Upozornil na skutočnosť, že súčasný internet je len chaotická zmes webových stránok a domén, ktorá začína rapídne narastať. Zároveň je stále zložitejšie nájsť správne relevantné informácie. Základnom sémantického webu je popis informácií pomocou vzťahov medzi entitami. Sémantický web je založený na technológií Resource Description Framework (RDF). [20]

1.4.2 Resource Description Framework

Resource Description Framework (RDF) je obecným frameworkom určeným pre popis, výmenu a znovu použitie metadat. Poskytuje jednoduchý model zápisu a zároveň nie je závislý na konkrétnej implementácií. Informáciu o objekte poskytuje tzv. tvrdenie (angl. statement). Základom tvrdenia je syntaktické vytvorenie trojice (angl. trilpes) medzi subjektom a objektom pomocou pedikátu (angl. predicate). Subjekt predstavuje zdroj, ktorému chceme priradiť

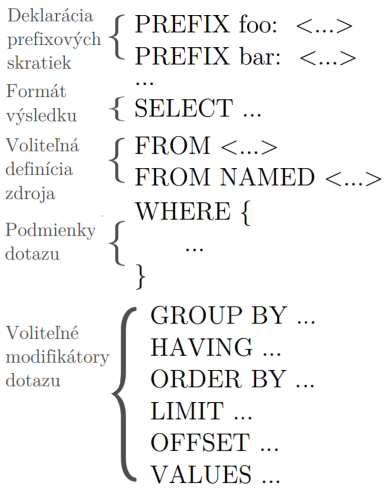
určitú vlastnosť. Predikát je priradením vlastnosti subjektu. Vlastnosť nadobúda hodnotu definovanú objektom. Týmto spôsobom je možne vyjadriť väčšinu potrebných informácií. Priradená hodnota vlastnosti môže predstavovať ďalší zdroj. Ak sa nejedná o ďalší zdroj, takýto objekt sa nazýva literál (angl. literal). Najlepšie ako si to je možne predstaviť, je znázornenie pomocou grafovej štruktúry. Subjekt spolu s objektom tvoria uzli grafu. Predikát je orientovanou úsečkou spájajúcou subjekt s objektom. Príklad je znazornený na obrázku 1.3. Dané tvrdenie udáva informáciu, že dielo identifikované pomocou `http://example.org/work/1` má autora (podľa slovníka `http://schema.org`) identifikovaného `http://example.org/person/123`. [21]



Obr. 1.3: RDF trojica

1.4.3 SPARQL

SPARQL predstavuje štruktúrovaný dotazovací jazyk, určený pre prácu s RDF uložiskami (tzv. triplestore). Ide o jednoducho použiteľný dotazovací jazyk. Existujú celkovo 4 základne konštrukcie dotazov, a to pomocou príkazov SELECT, ASK, CONSTRUCT a DESCRIBE.Názorná štruktúra dotazu je na obrázku 1.4



Obr. 1.4: Názorná ukážka štruktúry SPARQL dotazu.

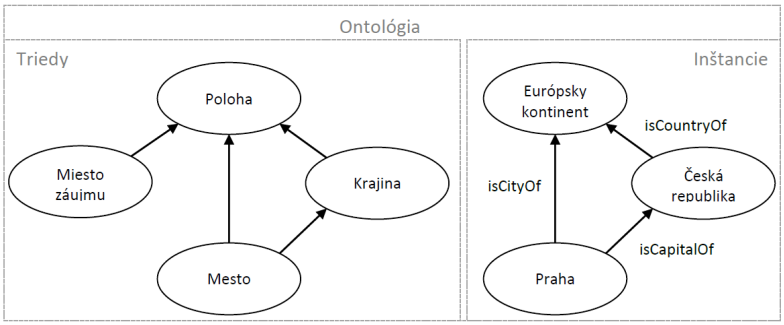
Na začiatku SPARQL dotazu je možné deklarovať skrátené prefixy, ktoré zastupujú v zvyšku dotazu uvedenú URI. Po nich, nasleduje definícia formátu výsledku pomocou kľúčových slov, ktoré už boli spomenuté vyššie. Kľúčové slová FROM poskytujú možnosť definovať obmedzenia na zdrojové grafy, z ktorých budú dáta pri tvorbe odpovede čerpané. Časť WHERE obsahuje podmienkové telo dotazu. V podmienkach sa používajú premenné, ktoré musia začínať symbolom "?". Pri tvorbe dopovede sa v nadom RDF zdroji dát na tieto premenné mapujú dostupné hodnoty, pričom musia byť splnené všetky podmienky z tela WHERE. Takéto hodnoty sa potom použiju pre tvorbu odpovede na základe požadovaného formátu výsledku. V rámci tvorby podmienok platí, že konjunkcia podmienok je vytvorená bodkov ".". OPTIONAL operátor slúži pre pripojenie časti (pravej) podmienky k predošlej (ľavej) podmienke, tak aby to nepoškodilo vyhodnotenie splnenia predošej časti. UNION operátor slúži k zlučeniu časti podmienok. Operátor MINUS odstráni výsledky získané podmienkov na pravej časti. Špeciálny operátor FILTER umožňuje redukovať výsledok len na tie ktoré spĺňajú danú podmienku. Tento operátor podporuje viaceré funkcie, ako sú logické, textové, číselné a pod. Okrem iného taktiež umožnuje filtrovanie trojích, ktorých objekty sú literálmi. [22]

1.4.4 Ontológia

Ontológia bola prebratá z filozofických vied, predstavovala náuku o bytí. Vo vzťahu k sémantickému webu sú ontológie chápané ako definície pojmov a vzťahov medzi nimi. Účelom je popis domény určitého ľudského záujmu. Takáto oblasť obsahuje jednotlivé triedy, ktoré sú prepojené vzťahmi. Objekty obsiahnuté v tejto doméne a ich prepojenie ontológia popisuje pomocou štyroch prvkov: entita, trieda, atribút a väzba. U niektorých definícii sa uvádza taktiež piaty prvok udalosť. Najznámejšou definício pojmu ontológia je definícia podľa T. Grubera: *„ontológia je explicitná špecifikácia konceptualizacie“*. Cieľom ontológie je definovať spoločné chápanie určitých tried a pojmov. Mala by podporovať porozumenie medzi viacerými ľuďmi, z rôznych profesií, komunikáciu počítačových systémov a taktiež zjednodušiť návrh znalostne orientovaných aplikácií. [23] Príklad jednoduchej ontológie je na obrázku 1.5

1.4.5 Linked Data

Pojem linked data uviedol v roku 2006 Tim Berners-Lee. Linked data definuje spôsob publikovania štruktúrovaných dát. Tieto dáta sú ďalej medzi sebou prepojené, čím navzájom získavajú pridanú hodnotu. Medzi základne technológie, ktoré sú pri linked data použité patria štandardné webové technológie: HTML, RDF a URI. Vďaka tomu sa rozširuje využitie publikovaných dát, strojovo čitateľných a ďalej zobraziteľných v rôznych podobách. Hlavným účelom linked data je sprístupnenie dát v jednoducho spracovateľnej forme. [24] Linked data používajú súbor overených postupov, pre publikovanie a prepoje-



Obr. 1.5: Príklad jednoduchej ontológie

nie štruktúrovaných dát na webe. Postupy boli prijaté v priebehu niekoľkých rokov, narastajúcim počtom poskytovateľov dát. To viedlo k vytvoreniu celosvetového dátového priestoru, pozostávajúceho z viac ako miliardy dát. Vďaka dodržiavaniu štandardov a overených postupov je umožnená interoperabilita1 dát a tzv. znovu použitie dát na webe. [25]

1.4.6 Wikipedia

Cieľom Wikipédie je vytvorenie voľne dostupnej a spoľahlivej encyklopédie. Wikipédiu založil Jimmy Wales a charakterizuje ju ako *"snahu o vytvorenie a distribuovanie voľne dostupnej encyklopédie každej osobe na planéte v jej rodnom jazyku, a to v čo možno najlepšej kvalite."* Je založená a základe softvéru wiki. Wikipedia je systematicky tvorená kolaboratívnou činnosťou svojich užívateľov. Predstavuje jeden z najväčších znalostných zdrojov. Anglická verzia obsahuje viac než 5 miliónov, článkov. Články Wikipedie obsahujú predovšetkým voľný text. Obsahuje taktiež niekoľko rôznych typov štruktúrovaných informácií:

- informačné boxy
- kategorizačné informácie
- mulitemédia
- geologické informácie
- odkazy na interné aj externé zdroje

[26]

1.4.7 DBpedia

DBpediu založili vedci z Free University of Berlin a The University of Leipzig v spolupráci s OpenLink Software. Prvá verzia bola vydaná v roku 2007. DBpediu je možné definovať viacerými spôsobmi. Je možné sa na tento projekt dívať ako na komunitu ľudí, venujúcich sa získavaniu informácií a znalosti

z Wikipedie. Alebo ako na službu, pomocou ktorej používatelia vedia vyhľadávať informácie z Wikipedie oveľa sofistikovanejším spôsobom. DBpedia obsahuje sadu nástrojov určených pre extrakciu štruktúrovaných informácií z Wikipedie. Tieto informácie tvoria bázu prepojených a strojovo čitateľných dát. DBpedia umožňuje jednoduchým spôsobom získať odpovoveď na otázky, které nie je jednoduché získať pomocou textového hľadania. Napríklad získať zoznam všetkých účastníkov zimných olympijskych hrier, ktorý sa narodili po roku 1980.[27, 28]

1.4.7.1 Extrakcia informácií z Wikipedie

Extrahované sú predovšetkým informácie z informačných boxov. Informačné boxy Wikipédie obsahujú základne fakty, zvyčajne v tabuľkovom formáte. Tabuľka priradzuje atribútom ich hodnoty. Extrahovanie informácií z informačných boxov prebieha pomocou dvoch hlavných paralelných metód:

1. Generické extrahovanie informačných boxov. Cieľom generickej extrakcie je veľké pokrytie. URI zdroja DBpedie je tvorená z URL článku Wikipedie. Táto URI je postupne použitá ako subjekt pre všetky extrahované triplety. Napríklad pre článok `https://en.wikipedia.org/wiki/Mazda` je jeho URL použitá pre vytvorenie zdroja `http://dbpedia.org/resource/`Mazda. URI predikátov sú vytvorené zreťazením menného priestoru `http://dbpedia.org/property/` a názvu atribútu informačného boxu. Napríklad atribút Revenue má potom vo výsledku URI predikátu `http://dbpedia.org/property/revenue`. Objekty sú tvorené z priradených hodnôt atribútov. Hodnoty sú predspracované a transformované na podporované reprezentácie hodnôt v RDF. Ide zvyčajne o objekty typu literál.

2. Mapovacie extrahovanie informačných boxov. Cieľom mapovania je lepšia kvalita dát. Rieši sa problém synonymných názvov atribútov a viacerých šablón, ktoré sú použité pre rovnaké typy. Šablóny Wikipedie sú mapované na ontológiu DBpedie. Ontológia je tvorená manuálne, usporiadaním najviac použitých šablón informačných boxov do hierarchie menších premís. Z šablon sa mapujú atribúty na vlastnosti ontológie. RDF trojice tejto metódy obsahujú predikát so základom `http://dbpedia.org/ontology/` .

1.5 Existujúce riešenia sumarizácie entit sémantického webu

Medzi podstatnú časť analýzy patrí oboznámenie sa s existujúcimi prácami. Relevantnými pre nás sú tie, ktoré riešia problém sumarizácie entít, stanovenia

dôležitosti informáciam, alebo aspoň niektorú jeho časť. Preto sa v nasledujúcej časti tejto práce pozrieme na niektoré aktuálne dostupné riešenia sumarizácie a identifikácie doležitostí informácií entit v prostredí sémantického webu.

1.5.1 Dôležitosť získaných informácií pomocou fuzzy sémantických sietí

V práci sa R. Lachica s kolektívom venoval hodnoteniu zdrojov použitím sémantickej vzdialenosti. Vzdialenosť určili pomocou prechodov medzi jednotlivými asociáciami množiny tém. Celková vzdialenosť je určená od témy používateľa. Asociácie pokladajú za dôležité. Súlužia ako vstupné body do sémantického webu. Konečný výsledok hodnotenia je založený na sémantickej vzdialenosti a kvalite hodnotenia zdroja. Asociácie majú rôzne váhy. Pri ceste do podkategórie triedy sa váha znižuje výraznejšie.

V prípade konextu špecifickej témy sa poradie počíta podľa všetkých výstupných ciest z počiatočnej témy. Váha relevancie je znižovaná každým prechodom. Všetky témy nad určitou prahovou hodnotou sú ohodnotené ako relevantné. Tento princíp vyplýva z techniky CSA (The Constrained Spreading Activation). Pomocou druhého prechodu je zlepšene umiestnenie zdrojov, ktoré sú spojené s témov používateľa. U ostatných kontextových tém sa hodnotenia automicky prispôsobia. Získaným riešeniam je pomocou sumarizácie priradené skóre relevantnosti a zároveň priradené skóre kvality zdroju riešení.

V prípade vyhľadávania pomocou kľúčových slov, sa každé kľúčové slovo porovnáva s témamy z množiny tém. Syntaktické zúženie termínov sa používa na vyhľadávanie zodpovedajúcich tém. Zväčšenie sémantického množstva termínov sa vykoná taktiež použitím CSA. Vďaka tomu sa získa zoznam subjektov, ktoré sa zhodujú s kľúčovými slovami z dotazu. Proces pokračuje získaním zdrojov. Poradie zdrojov sa stanoví pomocou najkratšej cesty. Cesty sú vypočítané pomocou algoritmu pre hľadanie najkratšej cesty v neorientovanom váženom grafe. V práci odkazujú na použitie Bellman-Fordovho algoritmu. Algoritmus slúži pre získanie najkratších ciest z počiatočného uzla do všetkých ostatných uzlov grafu. Ak neobsahovuje cyklus zápornej dĺžky. Ak by ho graf obsahoval, algoritmus ho dokáže detekovať. [14]

1.5.2 Sumarizácia entít pomocou vedomostnej hry

Vo svojej práci sa v roku 2012 A. Thalhammer a ďalší venovali sumarizácií entít na základe kvízovej hry. Kviz bol zameraný na jednú konkrétnu doménu. Doménu filmov. Kvíz podľa predpokladu mal poskytnút kvalitnejšiu sumarizáciu než boli automaticky generované sumarizácie. Kvíz obsahoval filmy, ktorých RDF grafy boli získane z Freebase. [1] Dataset obsahoval veľke množstvo

[1]Freebase API bola súčasťou služieb poknýkaných spločnosťou Google. Aktuálne je Freebase API ukončená. Presmerovali podporu znalostí z Freebase na službu Wikidata.

Pozícia	Freebase predikát	Pozícia	Freebase predikát
1	prequel	14	production company
2	film series	15	runtime
3	sequel	16	music
4	parodied	17	award
5	adapted original	18	actor
6	subject	19	story writer
7	genre	20	editor
8	initial release date	21	event
9	director	22	cinematographer
10	rating	23	budget
11	writer	24	film festival
12	featured song	25	film casting director
13	featured filming location		

Tabuľka 1.3: Tabuľka výsledkov dôležitosti vlastnosti filmov [6]

otvorených dát, čo bolo pre ich výber datsetu kľúčove. Medzi ďalšie zvažované alternatívy patrili DBpedia a Linked Movie Database (LinkedMDB).

Pre kvíz si zvolili 60 najlepšie hodnotených filmov na základe rebríčka IMDb. Princípom kvízu bolo zodpovedať otázku, výberom správnej odpovede. Implementovali 2 druhy otázok. S jedinou správnou odpoveďou a s n správnými odpoveďmi. Jednotlivé otázky boly zostavené na základe existujúcich tripletov v datasete. Príkladom pre vytvorenie otázky s jedinou správnou odpoveďou je použitie tripletu:

fb:en.pulp_fiction test:hasActor fb:en.john_travolta .

Nesprávne možnosti boli generované na základe nájdenia subjektov, ktoré maju priradený rovnaký predikát ako triplet otázky *(test:hasActor)*. Zároveň objekt tripletu otázky nie je v žiadnom triplete nesprávného subjektu. Otázka znela: *"John Travolta je hercom vo filme:"*. Otázka skúma mieru dôležitosti daného predikatu. Každa správna odpoveď pridáva skóre hráčovi. Výsledky hry boli spracované v zmysle percentuálnej úspešnosti zodpovedania otázky každého predikátu. Prehľadný výsledok ich metódy je v tabuľke 1.3. [6]

1.5.3 API rozhranie pre sumarizáciu prepojených dát

V roku 2015, A. Thalhammer a S. Stadtmüller riešili návrh a implementáciu Web API pre sumarizáciu entít. Účelom práce bolo poskytnutie nástroja, krorý dokáže spracovať sumarizáciu entít, dodaných klientom z rôzných zdrojov. Sumarizácia je založená na hodnotení prichádzajúcich odkazov z Wikipedie. Výsledom ich implementácie sa stala webová aplikácia summaServer a klientska knižnica summaClient.

V rámci implementácie riešili niekoľko problémov. Ako popisujú, znalostné bázy zvyčajne obsahujú podporu viacerých jazykov pre označenie zdrojov. Pridaním podpory jazykov je možné znížiť zbytočné požiadavky na iné zdroje alebo znalostné bázy. Pre potreby pokrytia širšieho aspektu vlastnosti, využili princíp n-árných vzťahov. Takto umožnili definovať maximálny počet prechodov k získaniu cieľového objektu. Na základe zadanej množiny predikátov je možné získať prepojenia, ktoré sú zamerané len na určitú oblasť. Vo výsledku je hodnotenie priradené skupine tvrdení (angl. statement). Uvádzajú, že rozdiel medzi RDF trojicou, ktorá obsahuje prepojenie na ďalší zdroj a literálom je pri tvorbe sumarizácie v účelnosti využitia. Literály poskitujú užitočné informácie o špecifickej entite naproti tomu prechod na iný zdroj umožňuje preskúmanie vzťahov danej entity. Pre každé RDF prepojenie typu *a link b* je možné vytvoriť obrátené prepojenie *b linkBy a*. Uvadzanie týchto obrátených prepojení vo výsledku sumarizácie má zmysel, pretože zahŕňa informácie o danom zdroji. Výsledná sumarizácia obsahuje skóre založene na slovníku vRank. [3]

Pulp Fiction

film director	Quentin Tarantino
distributor	Miramax
producer	Lawrence Bender
editing	Sally Menke
cinematography	Andrzej Sekula
series of	Mia Wallace
Wikipage disambiguates of	Vincent
Wikipage disambiguates of	Pulp fiction
Wikipage disambiguates of	Jules
Wikipage disambiguates of	Coolidge

Obr. 1.6: Ukážka výsledku summaClient[3]

1.5.4 Centralizovaná sumarizácia založená na príbuznosti a informovanosti

V práci z roku 2011 od G. Cheng, T. Tran a Y. Qu je riešený návrh a implementácia varianty metódy náhodného surfera. Využíva vzťahy popisov prvkov pre hodnotenie. Model je grafovo orientovaným datovým modelom, ktorý spĺňa RDF štandard.

Problém sumarizácie chápu ako výber n najlepšie hodnotených vlastnosti zdroja. Centralizovane orientované hodnotenie pokladajú za úspešne v rámci sumarizácie textov a ontológií. Preto tento spôsob hodnotenia aplikovali pri návrhu ich modelu. Centralizovane orientované hodnotenie si vyžaduje konštrukciu gafu, kde uzli zodpovedajú prvkom, ktoré majú byť hodnotené. Každý pár súvisiacich uzlov je spojený pomocou neorientovaných alebo orientovaných hrán. Vo výsledku sú uzly zoradené podľa ich centrality v grafe. Najčastejšie

vypočítané pomocou PageRank algoritmu. Pri použití náhodného surfera identifikovali dva problémy. Pojem centrálnosti môže byť príliš všeobecný. Identifikácia hlavných tém pôvodného popisu entity nebol jediným cieľom. Hľadala sa sumarizácia, ktorá najlepšie charakterizuje entitu a pomáha rozlišovať entitu od ostatných. To znamená, že pri meraní centrality by sa malo zvážiť, koľko informácií prináša určitá vlastnosť, ktorá môže prispieť k identifikácii entity. Druhý problém predstavuje hodnotenie susedných uzlov uzla za rovnako dôležité pri výbere cesty. Tento výber ma pre každý susedný uzol rovnakú pravdepodobnosť. Model neposkytuje stupeň príbuznosti na jemnozrnnejšej úrovni. Táto nepresnosť môže viesť k neoptimálnym výsledkom.

Vytvorili model podobný ako PageRank, ktorý simuluje správanie náhodného surfera pomocou dvoch druhov akcií. Surfer je cieľovo orientovaný. Prechádza množinou vlastnosti v snahe popísať požadovanú entitu. Surfer vykoná buď relačný pohyb s vyššou mierou pravdepodobnosti do uzla, ktorý obsahuje súvisiace informácie o téme, ktorá sa sumarizuje. Druhou akciou je informačný skok, pri ktorom s vyššou mierou pravdepodobnosti preskočí do uzla, ktorý poskytuje väčšie množstvo nových informácií. Tieto voľby sú reprezentované dvomi nerovnomerne rozdelenými pravdepodobnosťami, pričom jedna je daná súvislosťou medzi uzlami a druhá informatívnosťou uzlov. [29]

1.5.5 Zhodnotenie

V skratke sa pokúsime zhodnotiť získané informácie na základe prieskumu existujúcich riešení. Spôsob hodnotenia získaných informácií na základe fuzzy sématických sieti je viac zameraný na hodnotenie výsledkov, než na samotný proces ich získavania. Poskytuje návod pre stanovenie dôležitosti výsledku pomocou grafovej štrukúry. Pre našu prácu je viac informačný. Podáva však informácie o dôležitosti sumarizácie skupín entít. Sumarizácia entít pomocou kvízu, predstavuje zaujímavý nápad, ktorý zároveň dokáže poskytuje kvalitné informácie. Tie sú však závislé priamo na interakcií používateľa. Jej ďalšou nevýhodou je šírka pokrytia domén. Pre potreby vytvorenia bázy znalosti všetkých domén by si metóda vyžadovala veľké množstvo respondentov. Prístup k priradeniu miery dôležitosti predikátu pomocou jeho úspešnosti je možne v pozmenenej forme použiť aj v rámci nášho riešenia. Sumarizácia pomocou verejného API pre ľubovoľný RDF zdroj poskytuje vysokú mieru pokrytia. Pomocou neho je možné automatické spracovanie ľubovoľných zdrojov. Vo výsledkoch sa však objavujú duplicitné predikáty čo znižuje prehľadnosť dôležitých informácií. Táto skutočnosť poukazuje na to, že je vhodné pokúsiť sa tieto duplicity redukovať. Spôsob sumarizácie existujúcich zdrojov vieme ďalej použiť, ako určitú podpornú časť procesu našej sumarizácie. Práca zameraná na centralizovanú sumarizáciu na základe príbuznosti a informovanosti poskytuje potrené informácie pre analýzu vzťahov pri prechode grafom. Pokladá vhodný návod ako odlíšiť akcie pohybu grafom v rámci sumarizácie.

Kapitola 2

Metóda pre sumarizáciu a hodnotenie významnosti informácií

V úvode sa pozrime na všetky technológie s ktorými budeme pracovať. Ďalej si predstavme základnú myšlienku a návrh nášho modelu. V implementačnej časti sa pozrieme na proces vytvorenia nástroja, ktorý je založený na našom modeli. V poslednej časti tejto kapitoly si popíšeme spôsob práce s nástrojom a uvedieme príklady ako sme vygenerovali našu bázu znalosti.

2.1 Zdroje dát pre sumarizáciu entít

V našom modeli budeme používať ako hlavný zdroj informácií pre sumarizáciu datset NIF DBpedia abstracts. Budeme uvažovať dataset vo formáte NLP Interchange Format (NIF). Pre vyhľadanie informácií o zdrojoch na DBpedii pužijeme DBpedia SPARQL enpoint. Preto sa v úvode návrhu modelu zoznámime s vyššie uvedenými technológiami.

2.1.1 DBpedia abstracts

Wikipedia je najdôležitejším a najkomplexnejším zdrojom otvorených, encyklopedických poznatkov. Obsahuje články, ktoré predstavujú obrovský zdroj otvoreného textu v prirodzenom jazyku. Ako sme už uviedli vyššie, extrakcia informácií do DBpedie spočíva v mapovaní informačných tabuliek, šablón a ďalších ľahko identifikovaných štruktúrovaných údajov. Textový popis a údaje, ktoré môže obsahovať, nie sú súčasťou extakcie, hoci predstavujú najväčšiu časť dokumentu. Z celého popisu je extrahovaná len prvá kapitola. Táto kapitola je celá extrahovaná a predaná do DBpedie pod označením *abstract*. Exisuje taktiež skrátena verzia tohto abstraktu. Prvý odstavec abstraktu je mapovaný ako hodnota predikátu *rdfs:comment* [30] Pomocou sumarizácie

založenej na použití abstraktov preto dokážeme pokryť takmer všetky zdroje jednotlivých entít. V aktuálnej verzií DBpedia 2016-04 obsahuje približne 77% zdrojov s predikátom `http://dbpedia.org/ontology/abstract`.[7]

2.1.2 NLP Interchange Format

Formát NLP Interchange Format (NIF) je založený na RDF/OWL formáte. Jeho účelom je dosiahnutie interoperability medzi nástrojmi na spracovanie prírodného jazyka (NLP), jazykovými zdrojmi a anotáciami. Hlavnou časťou NIF je slovník, ktorý umožňuje reprezentovať reťazce ako zdroje RDF. Pomocou špeciálného dizajnu URI je umožnené presné určenie anotácií časti dokumentu. Tieto identifikátory URI je možné použiť na pripojenie ľubovoľných anotácií k príslušnej sekvencii znakov. Použitím týchto identifikátorov URI môžu byť anotácie zverejnené na webe ako prepojené údaje. NIF sa skladá z štrukturálnej, koncepčnej a prístupovej interoperability. Štrukturálna interoperability znamená, že URI sa používajú na pripojenie anotácií v dokumentoch pomocou fragmentu identifikátorov. Koncepčná predstavuje vetne štruktúrovanú ontológiu (SSO), ktorá bola špeciálne vyvinutá pre napojenie existujúcich textových ontológii. Pomocou toho je možné pripojiť bežné anotácie k URI. Prístupová predstavuje popis REST rozhrania. URI NIF-1.0 je konštruovaný z URI zdroja. Jeho URI je použitá ako prefix. Vo všeobecnosti je to jednoduchý a spoľahlivý prístup ako anotovať webové zdroje existujúcich zdrojov. Pre NIF URI je doporučené použitie ukončenia prefixu pomocou lomky "/", mriežky "#"prípadne dotazovou komponentou (napr. ?Nif-id=123). [31] [32]

2.1.3 NIF DBpedia Abstracts

DBpedia sa v súčasnosti primárne zameriava na zastupovanie faktických poznatkov, ktoré sú obsiahnuté v informačných boxoch Wikipedie. Za účelom možnosti získania štrukturovaných údajov z neštruktúrovaných textov, DBpedia poskytuje všetky informácie[2] extrahované zo zdrojového kódu HTML. Tieto informácie sú rozdelené do 3 datsetov [4]:

- nif-context - obsahuje text stránky ako kontext (vrátane indexov začiatku a konca). Príklad obsahuje je na obrázku 2.1
- nif-page-structure - obsahuje štruktúru stránky v sekciách a odsekoch (názvy, kapitoly a iné). Príklad obsahuje je na obrázku 2.2
- nif-text-links - obsahuje všetky odkazy na iné zdroje DBpedie, ako aj externé odkazy. Príklad obsahuje je na obrázku 2.3

Príklady obsahu jednotlivých datsetov, pre prehľadnosť sú vynechané typové označenia literálov:

[2]V rámci prvej publikácie (verzia 2016-04, z 12.10.2016) boli zverejnené len texty abstraktov. Až od verzie 2016-10 budú zverejnené všetky texty vo formáte NIF.[4]

```
dbr:Anthropology?dbpv=2016-04&nif=context a nif:#Context ;

nif:isString "Anthropology is the study of humanity...";

nif:beginIndex  "0" ;
nif:endIndex    "634" ;
nif:sourceUrl   <http://en.wikipedia.org/wiki/Anthropology> ;
nif:predLang    <http://lexvo.org/id/iso639-3/eng> .
```

Obr. 2.1: Ukážka obsahu nif-context datsetu.[4]

```
dbr:Anthropology?dbpv=2016-04&nif=context nif:hasSection dbr:Anthropology?
    dbpv=2016-04&nif=section_0_634 .

dbr:Anthropology?dbpv=2016-04&nif=section_0_634 a nif:Section ;
nif:beginIndex          "0" ;
nif:endIndex            "634" ;
nif:referenceContext    dbr:Anthropology?dbpv=2016-04&nif=context ;
nif:hasParagraph        dbr:Anthropology?dbpv=2016-04&nif=paragraph_0_330 ;
nif:hasParagraph        dbr:Anthropology?dbpv=2016-04&nif=paragraph_331_634 ;
nif:firstParagraph      dbr:Anthropology?dbpv=2016-04&nif=paragraph_0_330 ;
nif:lastParagraph       dbr:Anthropology?dbpv=2016-04&nif=paragraph_331_63 .

dbr:Anthropology?dbpv=2016-04&nif=paragraph_0_330 a nif:Paragraph ;
nif:beginIndex          "0" ;
nif:endIndex            "330" ;
nif:referenceContext    dbr:Anthropology?dbpv=2016-04&nif=context ;
nif:superString         dbr:Anthropology?dbpv=2016-04&nif=section_0_634 .

dbr:Anthropology?dbpv=2016-04&nif=paragraph_331_634 a nif:Paragraph ;
nif:beginIndex          "331" ;
nif:endIndex            "634" ;
nif:referenceContext    dbr:Anthropology?dbpv=2016-04&nif=context ;
nif:superString         dbr:Anthropology?dbpv=2016-04&nif=section_0_634     .
```

Obr. 2.2: Ukážka obsahu nif-page-structure datasetu.[4]

```
dbr:Anthropology?dbpv=2016-04&nif=word_29_37  a nif:Word ;
nif:referenceContext    dbr:Anthropology?dbpv=2016-04&nif=context ;
nif:beginIndex          "29" ;
nif:endIndex            "37" ;
nif:superString         dbr:Anthropology?dbpv=2016-04&nif=paragraph_0_634 ;
<http://www.w3.org/2005/11/its/rdf#taIdentRef>  dbr:Human ;
nif:anchorOf            "humanity" .

dbr:Anthropology?dbpv=2016-04&nif=phrase_65_84  a nif:Phrase  ;
nif:referenceContext    dbr:Anthropology?dbpv=2016-04&nif=context ;
nif:beginIndex          "65" ;
nif:endIndex            "84" ;
nif:superString         dbr:Anthropology?dbpv=2016-04&nif=paragraph_0_634 ;
<http://www.w3.org/2005/11/its/rdf#taIdentRef>    dbr:Social_anthropology ;
nif:anchorOf            "social anthropology" .
```

Obr. 2.3: Ukážka obsahu nif-text-links datsetu.[4]

2.1.4 DBpedia SPARQL enpoint

Okrem webovej reprezentácie jednotlivých zdrojov existuje ďalší verejne dostupný spôsob pre získavanie informácií z DBpedie. Predstavuje ho SPARQL endpoint, ktorý je dostupný na adrese `http://dbpedia.org/sparql`. SPA-

RQL endpoint je poskytovaný pomocou služby OpenLink Virtuoso. [33] Tento endpoint je vhodný, pre použitie v prípade, že používateľ, alebo klientska aplikácia vie vopred, aké informácie potrebuje. Okrem štandardného protokolu SPARQL endpoint podporuje niekoľko rozšírení. Vyhľadávanie v celkom texte na vybraných RDF predikátoch. Agregačných funkcí, ako napríklad *COUNT*. V rámci zabezpečenia má implementované obmedzenia pre určité dotazy. Napríklad dotaz, ktorý požaduje získanie celého obsahu úložiska. Ten je zamietnutý ako príliš náročný. Výsledky dotazov SELECT sú redukované na maximálne 1000 záznamov. [34]

2.2 Analýza sumarizácie entít

V našom návrhu sa pokúsime navrhnúť prístup pre sumarizáciu entít sémantického webu, ktorý bude možne použiť ako model pre ďalšie použitie. Vychádzajúc z poznatkov o údajoch, ktoré sú obsiahnuté na Wikipédii, sa zameriame na neštruktúrované textové informácie. V tomto prístupe nám bude nápomocný dataset DBpedia abstrakt.

Hlavnou myšlienkou prečo využiť informácie obsiahnuté v textovom popise je tá, že obsahujú pravdepodobne značne množstvo užitočných informácií o daných zdrojoch. Vyplýva to okrem iného, z množstva času, ktorý bol k vytvoreniu popisov venovaný ľudmi. Ďalším dôležitým faktom je usporiadanie samotného textu. Každý článok na Wikipédií je štrukturovaný do určitej miery podobne ako nejaká výstupná práca. Týka sa to v hlavnej miere úvodu, ten predstavuje abstrakt, ktorý poskytuje skrátený popis postatných informácií daného článku. Napríklad ak sa pozrieme na článok popisujúci mesto Praha, bude v prvej časti textu určite uvedená informácia, že Praha je hlavným mestom Českej republiky. Rovnako to bude pri iných hlavných mestách. Medzi ďalšie skutočnosti patrí to, že sú jednotlivé články medzi sebou prepojené odkazmi. To znamená, že Praha v abstrakte obsahuje odkaz na Česku republiku. Ak si to zovšeobecníme na všetky články, je možné, že pomocou analýzy vzťahov, ktoré sú v abstraktoch odhalíme ďalšie podstatné informácie, ktoré doposiaľ niesu extrahované. Ak sa pozrieme na typ vzťahov, ktoré môžu existovať v rámci odkazovania sa medzi článkami, môžeme identifikovať dve typy. Prvým je výstupne prepojenie zo zdrojového článku. Príkladom je už spomenutý vzťah článku Praha k článku Česká republika. Druhým typom je vzťah opačný. Sumarizácia zdrojov ma zmysel na úrovni typovej príslušnosti k určitej triede, definovanej pomocou ontológie. Výsledné hodnotenie bude v takom prípade priradené daným triedam ontológie.

Na základe identifikovaných tried ontológie bude potrebné zvoliť pre každú triedu dostatočný počet zdrojov. Ešte pred voľbou počtu zástupcov tried, je potrebné zvoliť spôsob výberu vhodných zdrojov. Je nutné definovať vhodnosť kandidátov. V tomto štádiu existuje viacero možností ako k výberu pristupovať. Uveďme si apoň niekoľko uvažovaných metód:

- Výber na základe najdlhšieho abstraktu.
- Výber na základe DBpedia PageRank
- Výber na základe najvyššieho počtu unikátnych predikátov

Varianty výberu na základe dĺžky abstraktu a počtu predikátov sú závisle na kompletnom prehľadaní zdrojov. Ide o optimalizačné vyhľadávanie. Navyše pri použití počtu predikítov je možné, že sa vhodnejšie zdroje dostanú na nižšie miesta. Zdroje totiž môžu obsahovať viacero "šumových"predikátov. Zároveň automatická identifikácia unikátnosti, predstavuje smotný veľký problém. Existuje viacero slovníkov, ktoré popisujú identické hodnoty pomocou odlišných predikátov. Výber na základe dĺžky abstraktu vyzerá síce vhodným kandidátom, ale ako sme už uviedli vyžaduje si kompletné prehľadanie zdrojov. Na druhej strane existuje vypracované hodnotenie DBpedia zdrojov pomocou PageRanku. Tvorbe PageRanku sa venovali A. Thalhammer a A. Rettinger. Vytvorili vlastný model PageRanku pre články Wikipédie a taktiež zdroje DBpedie, ktoré sú dostupné pomocou SPARQL dotazu. [35]

Ďalším problémom sú typické literálové vlastnosti. Vlastnosti, ktoré nemajú prepojenia. Tie sú obsiahnuté ako čisté časti textu. Takéto informácie nebude možné získať pomocou analýzov vzťhov článkov. Za účelom získania takýchto informácií budeme potrebovať rozšíriť proces hľadania prepojení. Znova pripadá do úvahy niekoľko prístupov ako to je možné realizovať. Pokúsme si načtnúť niekoľko možnosti ako doplniť literálové vlastnosti k vzťahovým vlastnostiam:

- Identifikáciou slov, ktoré sú okrem abstraktu obsiahnuté vo zvyšku textu.
- Aplikáciou metód dolovania dát z textov.
- Sumarizáciou tried obsiahnutých v DBpedií

Zaujímavým prítupom je identifikácia slov, ktoré sú použité aj vo zvyšnej časti textu. Existuje pritom niekoľko problémov. Prvým z nich je problém mapovania hodnôt na existujúce slovníky. Menším problémom je skutočnosť, že budeme pracovať s datestom NIF DBpedia abstrakt, pričom DBpedia aktuálne ešte nepublikovala podobný NIF dataset pre celé články z Wikipedie [3]. Spôsob sumarizácie na základe tried zdrojov aktuálne obsiahnutých, predstavuje jednnoduchší prístup, korý na druhej strane zaručí spoľahlivosť získaných informácií.

Za zmienku ešte stojí otázka počtu zástupcov jednotlivých tried pre sumarizáciu. Počet rozhodne nesmie byť príliž nízky a zároveň nemá zmysel zvoliť vysoký počet. Jednou z možnosti je použiť počet percentuálného zastúpenia v danej triede. Tu však nastáva problém s nerovnomerným rozložením zdrojov

[3] Až od verzie DBpedie 2016-10 budú zverejnené všetky texty vo formáte NIF.

medzi triedami. Akuálne DBpédia obsahuje približne 3,7 milóna zdrojov triedy *dbo:Agent* nadruhej strane len približne 22 tisic pre triedu *dbo:Disease*[36]. Nepredpokladá sa že by sa tento rozdiel v rozložení mohol niekedy zmeniť. Stanovením konštantného počtu sa zníži náročnosť, ale u tried s výrazne vyšším zastúpením je možná nižšia kvalita získaných výsledkov.

Možným problémom je, že vo výsledku sa objavia duplicitné hodnoty, ktoré sú mapované na iné predikáty slovníkov. Bude vhodné, ak sa pokúsime takéto duplikácie redukovať. Jednou z duplicít, ktoré sa najpravdepodobnejšie objavia je duplicita spôsobená dvojitým spôsobom extrakcie informácií z Wikipédie do DBpedie. Aj napriek komu, že hodnoty sú mapované na predikát s prefixom `http://dbpedia.org/ontology/`, sú často priradené aj pomocou predikátu `http://dbpedia.org/property/`. Ako bolo spomenuté vyššie v kapitole 14, hodnoty predikátátov typu *ontlogy* sú kvalitnejšie. Je možné, že existujú všetky 3 kombinácie výskytu týchto predikátov. Uvažujeme len kombinácie, kedy zdroj obsahuje aspoň jeden z týchto predikátov, alebo obsahuje obdidve. Príkladom je zdroj `http://dbpedia.org/page/Mazda`, ktorý obsahuje predikáty `http://dbpedia.org/ontology/owner` a zároveň `http://dbpedia.org/property/owner`. V tomto prípade ide o nežiadúcu duplicitu hodnôt.

Báza znalosti by mala obsahovať zoradenú množinu predikátov pre každú definovanú triedu v rámci zvolenej ontológie. Radenie je závislé na pridelenom hodnotení predikátov. Musíme si rozobrať možné spôsoby určenia významnosti predikátov:

- Pomocou precentuálného zastúpenia.
- Pomocou relatívnej pozícií v textoch.

Stanovenie dôležitosti pomocou percentuálného zastúpenia je jednoduchým spôsobom, ktorý je vhodnejší na situácie, kedy nedokážeme nájsť sofistikovanejší spôsob stanovenia významnosti. Identifikácia pozície v texte je priamo závislá na možnosti spätného zistenia umiestnenia odkazu v texte abstraktu.

2.3 Návrh modelu sumarizácie entít

V rámci definovaných vstupov, čiastkovch úloh a problémov uvedených v analýze si v tejto časti predstavýme konceptuálny model pre našu sumarizáciu entít.

Vďaka tomu, že existuje verejne dostupný dataset DBpedia abstrakt vo formate NIF, budeme môcť získať vztahy zdrojov priamo z neho. Model bude založený sa využití NIF DBpedia datasetu. Výber ontológie pre výber zdrojov daných tried je spojený so závislosťou na DBpedií. Budeme preto používať triedy ontológie DBpedie. Výber kontrétných zdrojov, pre sumarizáciu, budeme realizovať pomocou získania existujúceho hodnotenia DBpedia PageRank. Počet zvolených zdrojov stanovíme pomocou konštanty. Zlepšíme tým

rýchlosť spracovania, hoci vznikne riziko zníženia kvality výsledkov u nadpočetných tried. Nepredopkládame však, že by takéto zníženie bolo výrazne. Spôsob, ktorým doplníme literálové vlastnosti si pre náš model zvolíme sumarizáciu tried obsiahnutých v DBpedií. Bude sa jednať o ďalšiu doplnkovú sumarizáciu. Ďalšou doplňujúcou úlohou v rámci modelu bude redukcia duplicitných predikátov. Pokúsime sa ich na základe podobnosti zoskupiť pod jeden záznam vo výsledku sumarizácie. Dokážeme tak poskytnuť kvalitnejšiu bázu znalosti. Zároveň bude lepšie splnený cieľ pre zlepšenie prehľadnosti informácií o zdrojoch.

Model navrhnutý na základe analýzy obsahuje 4 hlavné komponenty a 3 pomocné. Model je znázornený na obrázku 2.4. Hlavnými komponentami sú:

Extraktor prepojení Má za úlohu na základe získaných zdrojov určených pre sumarizáciu, vyhľadať ich záznamy v NIF DBpedia abstrakt datasete. Jeho výstupom je množina prepojení, pre ktoré cez *Vyhľadávač predikátov* doplní predikáty, ktoré odpovedajú najdeným prepojeniam. Všetky získane informácie následne pomocou *Generátorá čiastkových výsledkov* uloží.

Generátor globálnej štatiskity Má za úlohu paralelne s generovaním štatisky zdrojov, vygenerovať jednu globálnu štatistiku literálov pre všetky triedy ontológie. Generuje ich na základe vstupných zdrojov a pomocou *Vyhľadávača predikátov* získa potrebné predikáty. Výsledok uloží v jednom súbore.

Identifikátor identických predikátov Má za úlohu pre všetky vstupné predikáty identifikovať tie, ktoré sú identické na základe stanovenej miery podobnosti. Výsledky priebežne ukladá do súboru.

Generátor bázy znalosti Má za úlohu spojiť všetky čiastkové štatistiky jednotlivých zdrojov, pripojiť globálnu štatistiku a redukovať identické predikáty.

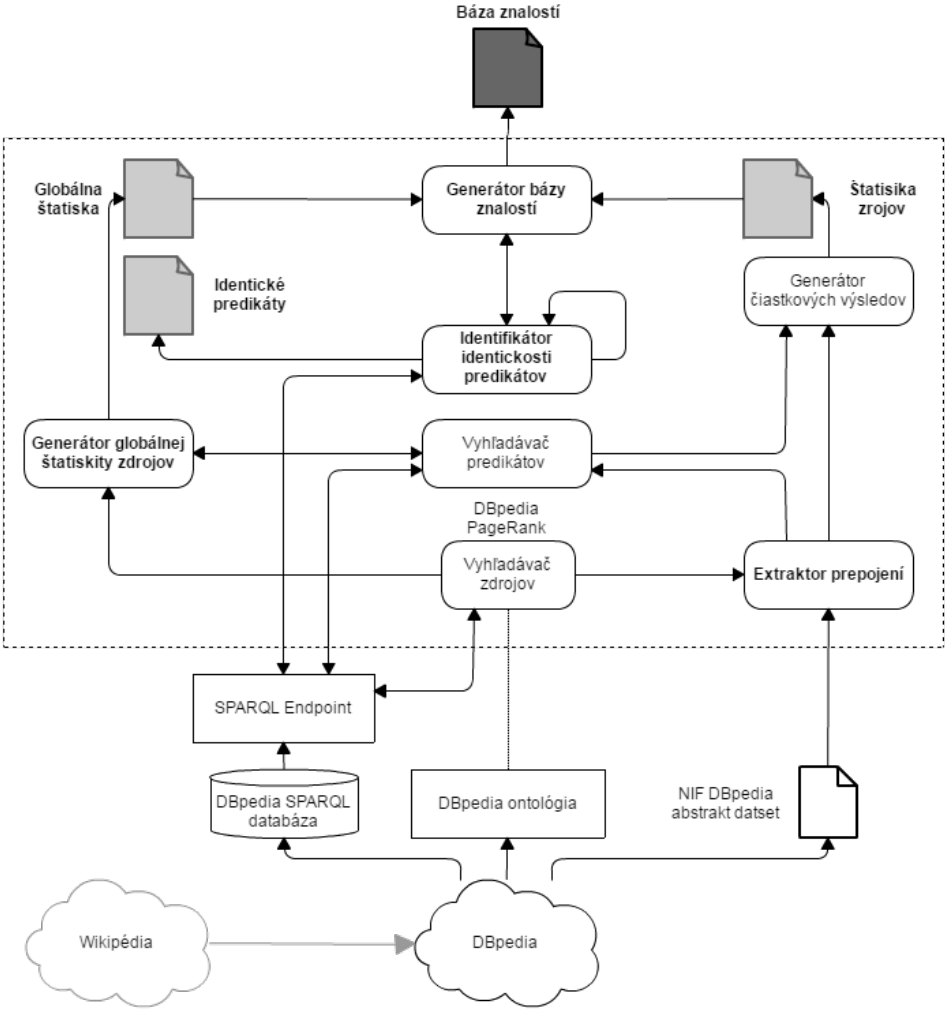
Pomocné komponenty:

Výhľadávač zdrojov Ma za ulohu na základe DBpedia PageRanku a triedy ontológie vrátiť zoradený požadovaný počet zdrojov.

Výhľadávač predikátov Má za úlohu na základe zadaných kritétrií vrátiť zoznam predikátov.

Generátor čiatkových výsledkov Má za úlohu uchovať všetky potrebné informácie daného zdroja pre následne spracovanie.

Ďalej si podrobnejšie popíšeme hlavné komponenty. Predovšetkým aké vnútorne procesy riešia a zároveň ako vypočítavajú jednotlivé hodnotenia.



Obr. 2.4: Model sumarizácie entít s využitím NIF DBpedia abstraktu

2.3.1 Extraktor prepojení

Koponenta bude pracovať s datasetom formátu NIF. Pre potrebu zabezpečenia zvládnutia veľkých datasetov s nim bude prácovať pomocou streamu. Rešpektuje dávkové spracovanie a pre zadanú množinu zdrojov vyhľadá potrebné údaje pomocou jediného prechodu datsetom. Potrebné údaje pre výstup sú definované na základe potrieb sumarizácie. Pre každý záznam zdroja vyhľadáva násobne (v zápise je použitý prefix nif:[4]):

[4]nif: zastúpuje http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#

- Prepojenie. Hodnota priradená v NIF triplete pomocou predikátu `http://www.w3.org/2005/11/its/rdf#taIdentRef`
- Indexy pozície v texte, pre text prepojenia. Hodnoty priradené v NIF triplete pomocou predikátu `nif:beginIndex` a `nif:endIndex`
- Paragraf výskytu prepojenia. Hodnota priradená v NIF triplete pomocou predikátu `nif:hasSection`

Výsledky úkladá ako čiatočné štatistiky každého zdroja. Rozlišuje 2 druhy získaných predikátov. Predikáty striktne viazané na daný zdroj a predikáty viazané na celú triedu ontológie. V rámci ukladania realizuje výpočet čiastkového skóre. To je vypočítane na základe relatívnej pozície odkazovaného textu v celom abstrakte. Zároveň je znížované pozíciou daného paragrafu v abstrakte. Nech A je množiná znakov abstraktu a D množina paragrafov abstraktu A potom počet znakov abstraktu A je rovný $c_A = |A|$ a počet paragrafov $c_D = |D|$. Ďalej nech je L množina degradácie dôležitosti vplyvom pozície paragrafu potom degradácia pre paragraf D_i je stanovená ako $L_{D_i} = 1 - \frac{i}{c_D} \times 0.1$ pričom $c_D > 0$. Nech pre odkaz o v paragrafe D_i je f index prvého znaku odkazovaného textu v abstrakte potom je pozičná váha odkazu o stanovená ako $W_o = 1 - \frac{f}{c_A}$ pričom $c_A > 0$. Výsledne čiastkové skóre pre predikát p odkazu o obsiahnutého v paragrafe D_i je definovaný ako súčin pozičnej váhy a degradácie pozíciou paragrafu. Vzťah je znázornený na obrázku 2.5

$$S_{T_p} = W_o \times L_{D_i}$$

Obr. 2.5: Definícia čiastkového skóre predikátu p triedy T

2.3.2 Generátor globálnej štatistiky

Komponenta je zodpovedná za vytvorenie doplnkovej štatisky sumarizáciou zdrojov DBpedie na základe príslušnosti k určitej triede. Pre každú triedu ontlógie získa potrebný počet jej zdrojov. Pre každý zdroj získava všetky jeho predikáty. Tie uchováva a navyšuje počet výskytu jednotlivých predikátov v rámci danej triedy. Mieru významnosti priradzuje na základe percentuálneho výskytu. Nech je C množina všetkých počtov výskytu predikátov potom C_p je počet výskytu unikátného predikátu p a S_p definujeme ako výsledné skóre predikátu p pre globálnu štatistiku danej triedy T. Vzťah je znázornený na obrázku 2.6.

$$G_{T_p} = \frac{C_p}{max(C,1)}$$

Obr. 2.6: Definícia globálného skóre predikátu p triedy T

2.3.3 Identifikátor identických predikátov

Účelom komponenty je redukcia duplicitných predikátov na základe priradených hodôt v podobe objektov. Vstupom pre komponentu je množina predikátov zoradených zostupne na základe aktuálne priradeného skóre. Overí pre všetky dvojice predikátov, počet identicky hodnôt priradených v rámci jedného zdroja. Počet vyšetrovaných predikátov je redukovaný na maximalný povolený počet. Nutné opatrenie pred neúmerne vysokým časom potrebným na vyšetrenie úplne všetkých kombinácií. Z redukovanej množiny predikátov sa vygeneruju všetky kombinácie dvojíc. Pre výpočet kritéria, ktoré rozhodne o príslušnosti dvojice k vzájomne identickým predikátom, je použité percentuálne zastúpenie. Pre každý predikát sa získa počet všetkých použití v rámci zdrojov. Nech je c_a počet všetkých použití predikátu a a c_b počet všetkých použití predikátu b. Nech o_a je objekt mapovaný pomocou predikátu a, objekt o_b je mapovaný pomocou predikátu b a c_{ab} počet spoločného použitia predikátov a a b v jednotlivých tripletoch, tak že objekty $o_a = o_b$ potom mieru identickosti predikátov definujeme ako podiel c_{ab} a maxima z celkového počtu použitia predikátov a a b. Vzťah je znazornený na obrázku 2.7.

$$I_{ab} = \frac{c_{ab}}{max(c_a, c_b, 1)}$$

Obr. 2.7: Definícia miery identickosti dvojice predikátov a, b

Na základe minimálnej miery identickosti sú dvojice ohodnotené, buď to ako identické alebo rôzne. Tieto výsledky sú priebežne ukladané. V prípade, že už pre danu dvojicu existuje zaradenie do jednej z týchto skupín a nie je explicitne zadaný požiadavok na ich opätovnu identifikáciu je dvojica vynechaná z procesu identifikácie.

2.3.4 Generátor bázy znalosti

Komponenta ma za úlohu spojenie čiastkých výsledkov jednotlivých zdrojov a globálnej štatistiky pre definovanú vstupnú triedu ontológie. Zároveň pri generovaní bázy znalosti aplikuje získané vedomosti o identickosti predikátov. Preto jednotlivé záznami osahujú množinu predikátov, ktorá v prípade identických obsahuje viacero predikátov. Tejto množine predikátov je stanovené jednotné výsledne skóre dôležitosti. Skóre dôležitosti je reálným číslom v intervale $(0, 1 >$, pričom hodnotenie 1 priradzuje maximálnu dôležitosť. Proces výpočtu výsledného skóre pozostáva z dvoch havných operácií.

Prvá operácia predstavuje spojenie čiastkových výsledkov. Pre každý predikát je sumované čiastkové skóre a ceklový počet výskytu vo vyšetrovaných zdrojoch. Nech c_p je počet výskytu predikátu p a S_{p_i} je čiastkové skóre predikátu p na pozícií i potom sumu celkového čiastkového skóre predikátu p definujeme ako sumu všetkých čiastkových skór. Vzťah je znázornený na obrázku 2.8

$$S_{T_p} = \sum_{i=1}^{c_p}(S_{p_i} \times \frac{1}{max(c_p,1)})$$

Obr. 2.8: Definícia celkového čiastkového skóre predikátu p triedy T

Druhá operácia načítava výsledky globálného skóre všetkých predikátov danej triedy a pripája ich k aktuálným celkovým čiastkovým výsledkom predikátov. Celkové výsledne skóre pre predikáty je stanovené spojením celkového čiastkového skóre a globálného skóre. Vzťah stanovenia výsledného skóre je znázornený na obrázku 2.9.

$$R_{T_p} = \begin{cases} \frac{S_{T_p} + G_{T_p}}{2} & , S_{T_p} > 0 \wedge G_{T_p} > 0 \\ max(S_{T_p}, G_{T_p}) & , inak \end{cases}$$

Obr. 2.9: Definícia výsledného skóre predikátu p triedy T

Tento vzťah definuje všeobecný spôsob stanovenia výsledného skóre. Na základe aktuálného návrhu však neuvažujeme iné než literálové predikáty v globalnej štatistike a opačne pri čiastkových výsledkov. Preto sa bude v praxi jednať o priradenie jednotlivých výsledkov.

Navrhovaný model poskytuje výslednú bázu znalosti, okrem toho poskytuje vedľašie výsledky. Medzi tie najpodstatnejšie patrí zoznam identických predikátov a globálna štatiska literálových predikátov. Okrem toho dokáže poskytnuť aj jednotlivé štatistiky zvolených zdrojov. Na základe týchto skutočnosti predpokládame, že model dokáže pokryť potrebné ciele tejto práce pre stanovenie miery důležitosti predikátov. Zároveň poskytne možné uplatnenie vedľajších výsledkov aj v rámci iných prác. Pre potreby čo možno najľahšieho použitia modelu bude nutná vhodná implementácia nástroja, ktorý aplikuje navrhnutý model. Problematiku implementácie nástroja navrhnutého modelu budeme riešiť v nasledujúcej kapitole.

2.4 Implementácia nástroja

V tejto kapitole sa budeme venovať celému procesu implementácie nástroja, ktorý aplikuje navrhnutý model. V úvode si popíšeme zvolený programovací jazyk a spôsob akým zabalíme nástroj v rámci použiteľného balíčka. Následne navrhneme štrukúru nástroja a uvedieme niektoré zaujímavé príklady z jeho implementácie.

2.4.1 Ruby

Ruby je skriptovací interpretovaný programovací jazyk. Vďaka tomu, že Ruby má celkom jednoduchú syntax, stále je však dosť výkonný na to, aby sa vedel vyrovnať známejším programovacím jazykom. Napríklad Perl a Python. Ruby je plne objektovo orientovaný. Autorom Ruby je len jeden človek, Yukihiro Matsumoto (známi ako „Matz“). Nevyhovovali mu ostatné programovacie jazyky, a preto sa rozhodol vziať z viacerých jazykov ich užitočné stránky a vytvoriť svoj vlastný programovací jazyk. Tak v roku 1995 vzniká prvá verzia Ruby. Štandartná implementácie Ruby („Matz“ interpreter) je napísaná v jazyku C, pričom má podobu klasického interpretra. Existuje niekoľko zaujímavých alternatív. Patrí medzi ne napríklad nativná implementácia v jazyku Java. JRuby umožňuje beh kódu Ruby v JVM prostredí. To samozrejme poskytuje všetky výhody JVM. [37]

Spôsob akým umožňuje jazyk Ruby vytvoriť ucelenú časť kódu v rámci určitého balíčka je vytvorenie tzv. gemu (angl. gem). Gemy sú knižnice pripraveného Ruby kódu, ktorý rieši určitú funkcionalitu. Pre správu gemov existuje ruby baličkovací manažér *RubyGems*. Tento manažer poskytuje štandartný spôsob distribúcie Ruby programov, jednoduchú inštaláciu a správu gemov. RubyGems používa sémantické verzovánie. Verzia se uvádza ako reťazec bodkami oddelenými troch nezáporných čísiel. Jednotlivé čísla sa definujú ako "major"."minor"."fix". [38] Každý gem obsahuje tieto základne časti:

- Zdrojový kód. Obsahuje výkonú časť funkcionality gemu, vytvorenú vývojarom.
- Gemspec súbor. Súbor ktorý obsahuje informacie o geme ako sú napríklad verzia, platforma, popis gemu. Obsahuje taktiež definície pre načítanie všetkýxh potrebných súborov z časti zdrojového kódu. Predatavuje súbor na základe, ktorého je riadený proces vytvárania gemu (angl. build gem).
- Súbor README.md. Osahuje základný návod pre použitie daného gemu.
- Súbor LICENSE.md. Obsahuje informácie o licencií.

[38]

2.4.2 Architektúra nástroja

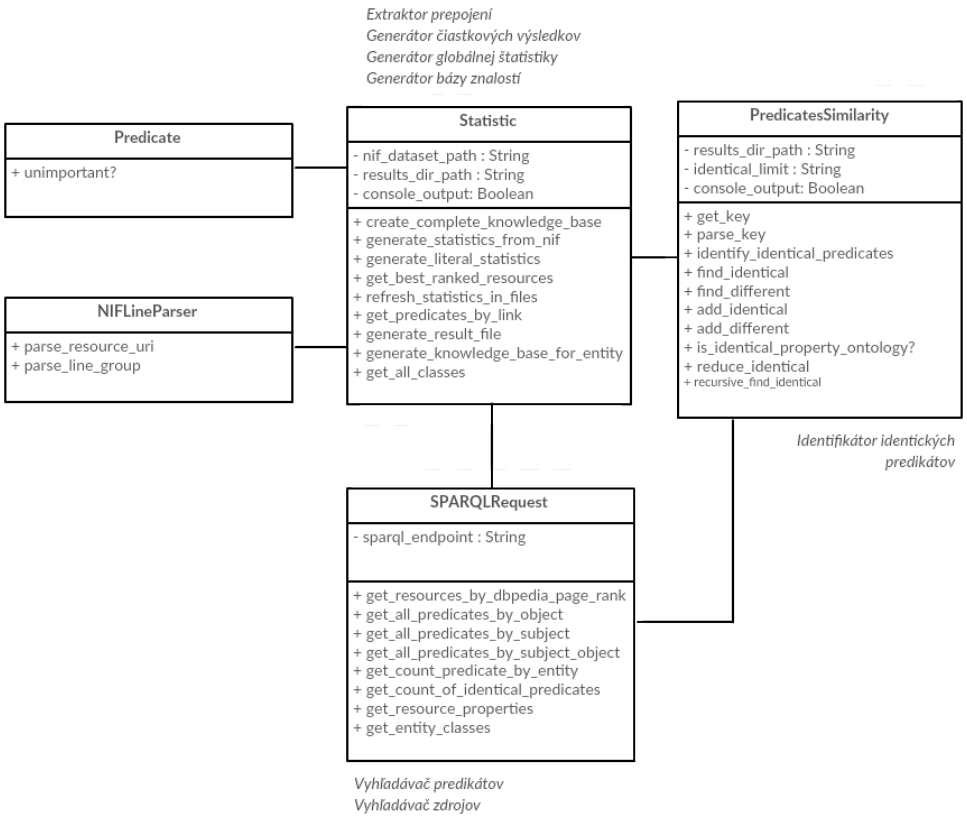
V zmysle definovaného modelu, v tejto časti predstavíme architekrúru vytvoreného nástroja. Hlavnými časťami nástroja sú triedy. Rozlišujeme pomocné triedy a aplikačné triedy, ktoré zodpovedajú za hlavnú funkcionalitu nástroja. Pomocné triedy maju na starosti hlavne prácu s ukladaným dočasných súborov, parsovním určitých štrukúr a pod. Diagram aplikačných tried je znázornený na obrázku 2.10. Nástroj obsahuje 3 hlavné aplikačné triedy, ktoré v

sebe majú implementované všetky komponenty z navrhnutého modelu. Sú to triedy:

Statistic Trieda v sebe obsahuje funkčnosti spojené s generovaním všetkých časti štatistik ako aj výslednej bázy znalosti.

PredicateSimilarity Trieda ma na starosti funkcionality pre komponentu identifikátor identických predikátov. Vyhodnocuje identickosť a stará sa o uchovanie získaných informácií o identickosti.

SPARQLRequest Predstavuje triedu, ktorá má na starosti komunikáciu s SPARQL endpointom. Ma definované metódy pre získanie potrebných výsledkov. Jednotlivé metódy aplikujú potrebne dotazy.



Obr. 2.10: Základný diagram tried nástroja

2.4.3 Charakteristika tried

Trieda *Statistic*, predstavuje najpodstatnejšiu triedu. Zabezpečuje zároveň najnáročnejšie operácie v rámci nastroja. Jedným z problémov, ktorý sa v

rámci jej implementácie objavil bol spôsob použitia veľkého datasetu. Aj napriek tomu že v rámci jazyka existuju gemy, ktoré sú určené pre prácu s RDF datasetmi, nedokázali sme získať taký, ktorý by neriešil použitie datasetu spôsobom, kedy dochádza k načítaniu celého súbora do pamäte. Preto bolo nutné použiť vlastné riešenie pre vyhľadávanie tripletov v datsete. Nutnosťou bolo spracovanie pomocou streamu. Pri takomto použití načitavania sme získali postupne jednotlive riadky súboru. V rámci načitavania riadkov sme následne identifikovali jednotlivé tvrdnia (angl. statement) ako postupnosť sedmíc riadkov. Tieto skupiny riadkov boli pre jednoduchosť parsované pomocou Ruby metódy *scan*, pre stringové hodnoty, s využitím nami definovanými regulárnými výrazmi. Na základe týchto potrieb vznikla trieda *NIFLineParser*. Jej hlavnou úlohou je získanie potrebných údajov z jednotlivých skupín riadkov. Týmto sa vyriešil spôsob extrakcie informácií z NIF DBpedia datsetu. Po extrakcií informácií trieda pomocou metód triedy *SPARQLRequest* získa potrebné predikty. Pre predikáty vypočita čiastkové skóre. Všetky výsledky uloži v cieľovom adresari.

Trieda *PredicatesSimilarity* ma jedinú úlohu. Rozhodnúť o miere identickosti predikátov. Pre potreby rozhodnutia o príslušnosti k skupine na základe miery identickosti definovanej v 2.7 bolo nutne stanoviť prahovú hodnotu. V prvých testoch sme použili vyššiu hodnotu 0.9. Následne sme na základe výsledkov usúdili, že je táto hodnota príliš striktná a nezískali sme takmer žiadne identické predikáty. Znižili sme danú konštantu na úroveň 0.8. Pri takto nastavenej úrovni sa zvýšilo množstvo očakávaných identických predikátov, pričom zároveň sa počas testov neobjavili žiadne nevhodne priradené. Preto pokladáme hodnotu 0.8 sa prijateľnu pre naše potreby. Ďalšou dôležitou konštantov, ktorú bolo nutne zaviesť je maximlány počet predikátov k overeniu. Keďže proces identifikácie spočíva v overení každej kombinácií dvojíc, je nárast výpočetného času výrazný. Stanovili sme hornú hranicu predikátov na 250, odpovedá to 31 125 kombináciam dvojíc. Tento limit je aplikovaný pri každom voláni metódy *identify_identical_predicates*

V rámci triedy *Predicate* je definovaná jediná metóda, určená pre vylúčenie predikátov, ktoré sú nepodstatné. Táto príslušnosť bola definovaná manuálne. Nepredstavuje teda žiaden automatický proces. Predikáty sú vylúčene na základe myšlienky, že nepodávajú informácie k odlíšeniu jednotlivých tried medzi sebou. V rámci predikátov jednotlivých zdrojov, existujú také, ktoré môžu mať vysoku mieru informácie, nepridávajú však žiadnu pridanú hodnotu pre informovanie používateľa. Medzi takéto predikáty patria napríklad identifikátory Wikipedia článkov, z ktorých boli extrahované. Takéto predikáty potom obsahuje každý zdroj. Vplyvom na spôsob stanovenia výsledného skóre dôležitosti, sa vždicky tieto predikáty umiestňovali na prvých miestach. Preto sa zaviedla množina tzv. nedôležitých predikátov. Medzi nedôležité predikáty patria:

- `http://xmlns.com/foaf/0.1/primaryTopic`
- `http://dbpedia.org/ontology/wikiPageRedirects`

- `http://dbpedia.org/ontology/wikiPageDisambiguates`
- `http://dbpedia.org/ontology/wikiPageRevisionID`
- `http://dbpedia.org/ontology/wikiPageID`
- `http://www.w3.org/2002/07/owl#sameAs`
- `http://www.w3.org/2000/01/rdf-schema#seeAlso`
- `http://www.w3.org/2002/07/owl#differentFrom`
- `http://dbpedia.org/ontology/wikiPageExternalLink`
- `http://xmlns.com/foaf/0.1/depiction`

Na druhej strane sme zároveň identifikovali predikáty, u ktorých nemá zmysel uvažovať ich dôležitosť. Preto sú aj tieto predikáty vylúčené a nebudú použité v rámci sumarizácie. Patria sem hlavne základne popisné predikáty. Zvolili sme množinu týchto, ako takzvaných všeobecne dôležitých predikátov:

- `http://dbpedia.org/ontology/thumbnail`
- `http://xmlns.com/foaf/0.1/name`
- `http://www.w3.org/2000/01/rdf-schema#label`
- `http://dbpedia.org/property/name`
- `http://dbpedia.org/property/commonName`
- `http://dbpedia.org/property/title`
- `http://www.w3.org/2000/01/rdf-schema#comment`
- `http://dbpedia.org/ontology/abstract`

Ako sme už uviedli trieda *SPARQLRequest* obstaráva komunikáciu s endpointom. Má preto zároveň preddefinovanú štrukúru dotazov pre potrebné výsledky. Pre vyjasnenie spôsobu získavania jednotlivých informácií z DBpedia SPARQL endpiontu, si predstavíme podstatné dotazy.

Získanie zdrojov na základe DBpedia PageRank. Vrámci tohto dotazu je potrebné definovať parameter *entity* a *limit* pričom entity definuje pre akú triedu budú získané zdroje. Napríklad je to `http://dbpedia.org/ontology/Person`. Limit je dafaultne nastavený na hodnotu 10. Dotaz je znázornený na obrázku 2.11.

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX vrank:<http://purl.org/voc/vrank#>

SELECT ?entity ?rank
FROM <http://dbpedia.org>
FROM <http://people.aifb.kit.edu/ath/#DBpedia_PageRank>
WHERE {
  ?entity rdf:type #{entity_type}.
  ?entity vrank:hasRank/vrank:rankValue ?rank.
}
ORDER BY DESC(?rank) LIMIT #{limit}
```

Obr. 2.11: SPARQL dotaz pre získanie zdrojov na základe DBpedia PageRank

```
SELECT DISTINCT ?property

WHERE {
  #{subject} ?property #{object}.
}
```

Obr. 2.12: SPARQL dotaz pre získanie predikátov zdroja na základe extrahovaného prepojenia

Získanie predikátov zdroja na základe extrahovaného prepojenia. Vstupom pre tento dotaz je vyšetrovaný zdroj ako *subject* a extrahované prepojenie ako *object*. Výsledok je použitý ako tzv. striktné predikáty vyšetrovaného zdroja. Dotaz je znázornený na obrázku 2.12.

Získanie predikátov triedy zdroja na základe extrahovaného prepojenia. Získanie týchto predikátov je logicky rozdelené do dvoch zvláš-ných dotazov v prvom sa získavú kompletne všetky predikátky z tripltov, v ktorých je *object* extrahované prepojenie. V druhom dotaze sa pre každý predikát vyhľadá počet všetkých použití predikátu v triede ako $entity_class$. Uvažuje sa vstupné aj výstupné prepojenie. Dotaz je znázornený na obrázku 2.15 respektíve 2.14.

```
SELECT DISTINCT ?property

WHERE {
  ?subject ?property #{object}.
}
```

Obr. 2.13: SPARQL dotaz pre získanie všetkých predikátov na základe extrahovaného prepojenia

Získanie predikátov pre globálnu štatistiku triedy. Vstupom pre dotaz je zdroj ako *subject*. V prípde že sú požadované iba predikaty v rámci

```
SELECT DISTINCT COUNT(?subject) as ?count

WHERE {
  ?subject a #{entity_class} .
  {?subject #{predicate} ?a .} UNION {?b #{predicate} ?subject .}
}

ORDER BY DESC(?count)
```

Obr. 2.14: SPARQL dotaz pre získanie počtu použití predikátu triedy.

tripletov s literalmi, doplní sa do dotazu filtračne kritérium only_literal ako *FILTER(isLiteral(?object))*. Dotaz je znázornený na obrázku 2.15.

```
SELECT DISTINCT ?property

WHERE {
  #{subject} ?property ?object.
  #{only_literal}
}
```

Obr. 2.15: SPARQL dotaz pre získanie predikátov globálnej štatistiky triedy

2.5 Výstup nástroja

Výstupom nástroja sú 4 typy súborov v zmysle navrhnutého modelu. Všetky súbory sú vo formáte JSON. Voči modelu je pozmenený spôsob ukladaných hodnôt. Pre jednotlivé predikáty v rámci všetkych typov okrem výslednej bázy znalosti, sú priradené počty použitia. V prípade aplikácie výpočtu by sa uchovali len relatívné hodnoty, ktoré pri zmenách a pokusoch nebude možne spätne prepočítať na pôvodne počty. Preto sa všetky typy skór prepočítavaju až pri finálnom generováni bazy znalosti. Ďalej si ukážeme štrukúru jednotlivých výsledných súborov.

Súbor s globálnov štatistikov. Súbor obsahuje JSON objekt so štrukútov kľúč *trieda ontolígie* a hodnota objekt s štruktúrov kľúč *predikát* a hodnota *počet všetkých výskytov*. Štruktúra je znázornená na obrázku 2.16.

Súbor s identickými predikátmi. Súbor obsahuje pole reťazcov. V rámci jedného reťazca sú uchováne pridikáty, ktoré sú si navzájom identické. Každý predikát je ohraničený znakmi «ä »". Štruktúra je znázornená na obrázku 2.17.

Súbor s čiastkovým výsledkom zdroja. Súbor obsahuje na prvej úrovni 2 základne kľúče *resource_uri* s hodnotou URI zdroja, ktorému patria

```
{
 "City": {
           "http://dbpedia.org/ontology/PopulatedPlace/areaMetro": 415,
           ...
         }
 ...
}
```

Obr. 2.16: Štruktúra súboru s globálnou štatistikou

```
[
 "<http://dbpedia.org/ontology/raceHorse><http://dbpedia.org/property/
     horses>",
 ...
]
```

Obr. 2.17: Štruktúra súboru s identickými predikátmi

výsledky a *nif_data*, ktorý obsahuje pole s objektmi pre každé nájdene prepojenie. Objekt prepojenia obsahuje kľúče *link* s hodnotou nájdeného prepojenia v NIF datasete, *anchor* s hodnotov reťazca, pre ktorý je dané prepojenie mapované, *indexes* obsahuje pole s počiatočným a koncocým indexom daného reťazca, *section* s hodnotov z akého paragrafu v abstrakte je získane prepojenie, *weight* s hodnotou čiastkového skóre (čiastkové skóre je zhodné pre všetky predikáty obsiahnute v rámci daného objektu prepojenia, výpočet je založený na definícií 2.5). Ďalej obsahuje *properties* a *strict_properties*, pričom prvému sú priradené hodnoty predikátov nájdené pomocou daného prepojenia pre celú triedu a druhému su priradené predikáty, nájdené pomocou prepojenia priamo k vyšetrovanému zdroju *resource_uri*. Obe priradenia sú zároveň zanorené pod triedu, do ktorej patrí daný zdroj. Štruktúra je znázornená na obrázku 2.18.

Súbor s bázov znalosti. Súbor obsahuje objekt s kľúčmi pre jednotlivé triedy. Každej triede je priradené pole objektov. Pole predstavuje zoradený zoznam na základe miery dôležitosti každého záznamu. Objekty obsahujú výsledné skóre dôležitosti *score* a pole *predicates* s predikatmi, ktoré sú priradené na danu pozíciu. Štruktúra je znázornená na obrázku 2.19.

Nástroj bol vo finálnej verzií zabalený ako gem na základe štandardu RubyGems. Zároveň bol pre potreby neskoršieho použitia publikovaný pomocou RubyGems manažéra. Gem je dostupný na adrese `https://rubygems.org/gems/browser_web_data_entity_sumarization`. Spôsob ako je možne gem získať je jeho inštalácia. Spôsob intalácie a zavedenia gemu pre použitie je znazornený na obrázku 2.20. Po vykonaní *require* je celá funkcionalita gemu sprístupnená.

```
{
 "resource_uri": "http://dbpedia.org/resource/Abraham_Lincoln",
 "nif_data": [
  {
   "link": "http://dbpedia.org/resource/Whig_Party_(United_States)",
   "anchor": "Whig Party leader",
   "indexes": ["588","605"],
   "section": "paragraph_435_1442",
   "properties": {
     "Person": {
        "http://dbpedia.org/ontology/occupation": 329669.0,
        "http://dbpedia.org/ontology/otherParty": 3584.0,
        "http://dbpedia.org/ontology/politicalPartyInLegislature": 4.0,
        "http://dbpedia.org/property/otherparty": 3620.0,
        "http://dbpedia.org/property/party": 78771.0
     }
   },
   "weight": 0.8715,
   "strict_properties": {
     "Person": {
        "http://dbpedia.org/ontology/party": 80846.0
     }
   }
  },
  ...
  ]
}
```


Obr. 2.18: Štruktúra súboru s čiastkovým výsledkom zdroja


```
{
 "Game": [
    {
     "score": 0.8912,
     "predicates": [
      "http://dbpedia.org/ontology/designer",
      "http://dbpedia.org/property/designer"
    ]
    },
    ...
 ],
 ...
}
```


Obr. 2.19: Štruktúra súboru bázy znalostí

```
V CMD:
gem install browser_web_data_entity_sumarization

V Ruby projekte - gemfile:
gem 'browser_web_data_entity_sumarization'

V Ruby skripte:
require 'browser_web_data_entity_sumarization'
```

Obr. 2.20: Spôsob inštalácie nástroja

2.6 Použitie nástroja

V tejto kapitole si popíšeme, ako sme generovali výslednú bázu znalosti. Poskytneme takto stručný návod použitia nástroja.

Pre potreby vygenerovania bázy znalosti v zmysle sumarizácie sme analyzovali dostupné NIF datasety. Vychadzajúc zo štruktury tak, ako je uvedená na obrázkoch 2.1,2.2 a 2.3 sme za kľúčový dataset zvolili *nif-text-links*. Zvolený typ datsetu sme získali z publikácie DBpedia 2016-04. Dataset je dostupný na adrese `http://downloads.dbpedia.org/2016-04/ext/nif-abstracts/en/`. Samotný dataset ma približnú veľkosť 45 GB. V ďalšom kroku sme potrebovali vybrať, pre ktoré triedy ontológie budeme generovať výsledky. Ako sme už uviedli uvažujeme triedy z ontologie DBpedie. Hierarchia ako aj zoznam týchto tried je verejne dostupný na `http://mappings.dbpedia.org/server/ontology/classes/`. V úvodných testoch sme spracovali len triedy `http://dbpedia.org/ontology/City` a `http://dbpedia.org/ontology/Person`. V závere, po vyladení nedostatkov, sme sputili generovanie pre všetky triedy tejto ontológie. V čase generovania obsahovala 685 tried. Reálne však neboli pre každú definovanú tried nájdené zdroje na základe ich DBpedia PageRanku. Vo výsledku sme spracovali sumarizáciu pre 401 tried. Dôležitým výberom bol počet zástupcov každej triedy. Zvolili sme si konštantný počet 100 zdrojov pre každú triedu. Ako sme už uviedli, pre niektoré triedy neboli získaný zástupcovia. Zároveň u niektorých tried bol počet získaných zástupcov nižší než bol stanovený požadovaný počet. Vo výsledku sme teda reálne spracovali 32 060 zdrojov. V prieme sa teda reálne pre každu triedu použilo približne 80 zrdojov.

Spôsob akým sa spúšťa generovania bázy znalosti, je volanie metódy *create_complete_knowledge_base* inštancie triedy *Statistic*. Metóda na vstupe vyžaduje hash s parametrami:

- entity_types - hodnotou je pole tried ontológie, pre ktoré budú následne získaný zástupci.
- entity_count - hodnotou je počet, koľko zástupcov ma byť pre každú triedu získany.
- demand_reload - označenie, či sa požaduje pregenerovanie čiastkových výsledkov pre zdroje, ku ktorým už aktálne sú uložené.
- identify_identical_predicates - označenie, či sa požaduje v závere identifikácia identických predikátov.

Nami vygenerována báza znalosti mala aplikovane veštky konštrukcie a výpočty, ktoré sme si v tejto kapitole definovali. Pokladáme ju preto za kompletnú a vhodnú pre použitie v rámci webovej aplikácie za účelom zlepšenia prehľadania informácií. Preto sa v nasledujúcej kapitole budeme venovať použitiu tejto bázy znalosti.

Kapitola 3

Webová aplikácia

V rámci tejto kapitoly si popíšeme proces návrhu implementácie a zavedenia webovej aplikácie, ktorá dokáže inerpretovať vygenerovanu bázu znalosti. V prvej časti sa pozrieme na spůsob intergacie bázy znalosti v rámci webovej aplikácie. Ďalej si popíšeme návrh a implementáciu.

3.1 Integrácia výsledkov sumarizácie entít

Báza znalosti je reprezentovana vo formate JSON, preto nepredstavuje integrácia žiaden problém. Pre zjednošujenie načítavania bázy znalosti je obsiahnutá v rámci vytvoreného nástroja. Tento nástroj je použitý na strane severa webovej aplikície. Server využíva niektoré jeho pomocné metódy a získava spomínanú bázu znalosti. Nástroj ma preto implementovanú pomocnú triedu *CacheHelper*, pomocou ktorej okrem iného načítava bázu znalosti. O tento proces sa stará metóda *load_knowledge*, ktorá na základe požadovaného typu výsledného súboru ho načíta z priečinku */knowledge*. Okrem už spomínanej bázy znalosti je v rámci tohto priečinu uložená hierarchia tried ontológie na základe, ktorej bola báza znalosti vygenerovaná a súbor so zoznamom všeobecne dôležitých predikátov. Všetky tieto výsledky sú práve potrebné pre aplikovanie výsledkov vo webovej aplikácií.

3.2 Návrh prehliadača

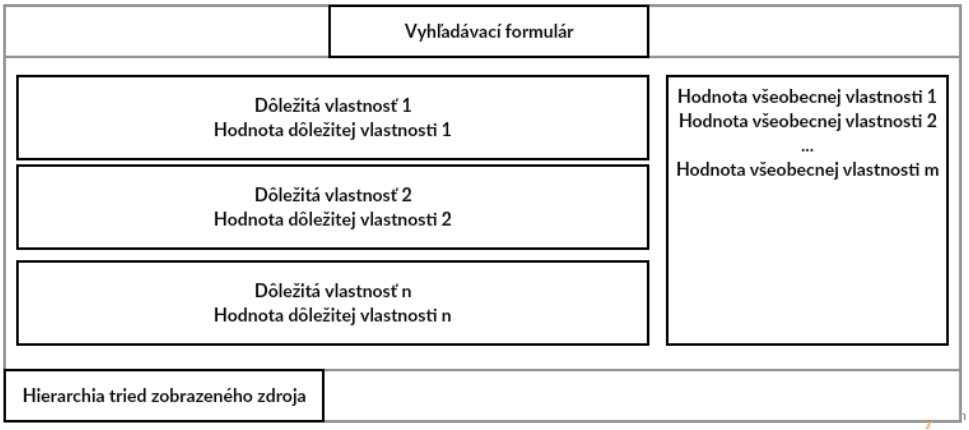
Pri našom návrhu prehliadača musíme vychádzať zo stanoveného cieľa práce a síce, sprehľadnenie informácií o zdrojoch DBpedie. Preto budeme potrebovať vytvoriť užívateľský jednoduchú aplikáciu, ktorá však musí zároveň poskytnúť dostatočnú účinnosť prehliadania informácií. Z hľadiska grafického návrhu budeme potrebovať užívateľský intuitívne prostredie. Aby bolo prostredie dostatočne intuitívne, potrebujeme si stanoviť aký je hlavný účel našej aplikácie. Hlavným účelom, bude samozrejme vyhľadávanie. Uvažujme teda používateľa

našej aplikácie, ktorý je posadený pred našu aplikáciu. Komponenta, ktorú bude najviac očakávať je vyhľadávacie pole (formulár), kde uvedie aký zdroj ma v záujme vyhľadať. V tomto štádiu nastáva určitý problém, ktorý sme si uvedomili. A síce ako umožniť používateľovi zadanie iba kľúčového výrazu, bez nutnosti poznania URI jednotlivých zdrojov. Bude jednoznačne veľkým prínosom ak mu poskytneme určitý našepkávač (angl. autocomplete) zoznam s návrhmi zdrojov. Na tento problém budeme myslieť v ďalšej časti implementácie. Máme teda základnú komponentu celej aplikácie, určenej k interakcií s používateľom. Mala by sa umiestniť v priestore, kde bude ľahko nájditeľná a zároveň používateľom očakávaná. Uvažujme teda základnú navigačnú lištu, ktorej dominantnou komponentov bude spomínaný vyhľadávací formulár. Pre ďalšie potreby uvažujme, že tam ďalej bude prepínanie základných stránok aplikácie.

Potom, ako používateľ odošle žiadosť, pre získanie informácií zvoleného zdroja, mali by sa zobraziť v hlavnej časti stránky. Samotná voľba by mala byť, čo možno najrýchlejšia. Bez nutnosti stláčať tlačidlo a pod. Povedzme teda, že sa tak stane po výbere navrhovaného zdroja, ktorý zaručene existuje na DBpedií. Jednotlivé informácie budú predstavovať určité dvojice. Ide o dvojice typu tvrdenie a význam. Vždy to bude názov vlastnosti a jej hodnota. Tieto výsledné informácie budú zobrazené pod navigačnou lištou.

Je ešte potrebne uvažovať, či existuje nejaký rozdiel v jednotlivých možných informáciách, ktoré sa budú zobrazovať. Jedným takým rozdielom môže napríklad byť účel informácie. Väčšina zdrojov obsahuje špecifické informácie, ktoré sú vlastné danému typu triedy, prípadne tried, do ktorých patrí. Avšak takmer u každého zdroja sú taktiež tzv. spoločné typy informácií. Myslíme tým typ informácií, ktoré sa dajú nájsť u každého zdroja. Ak by sme si dokázali zvoliť malé množstvo takýchto jednoznačných informácií mohli by sme návrh zobrazenia informácií upraviť na rozdelenie dvoch sĺpcov. V pravom budú zobrazene len všeobecné hodnoty. Medzi všeobecné hodnoty, bude zaradený veľmi malý počet informácií. Budú to informácie, ktoré vždy účinné popisujú dané zdroje. Medzi takéto informácie patrí názov, alebo titulok. Ďalšou informáciu je náhľadový obrázok a popis. O týchto informáciách nemá zmysel diskutovať či sú podstatné pre zobrazenie.

Za účelom demonštrácie zmeny dôležitosti informácií, pri zmene konkrétnosti zaradenia zdroja k typu triedy, bude v spodnej časti obrazovky zobrazená hierarchia tried. Hierarchia bude odpovedať aktuálne zobrazenému zdroju. Pričom bude možne, pomocou hierarchie meniť spôsob interpretácie informácií o zdroju podľa zvolenej triedy. Automaticky bude vždy aplikované zobrazenie pre najkokretnejšiu triedu hierarchie. Schématický návrh vizuálnej podoby prehliadača je zobrazený na obrázku 3.1.



Obr. 3.1: Návrh dizajnu webovej aplikácie

3.3 Implementácia prehliadača

Implementácia prehliadača spočíva z dvoch základných častí. Ide o serverovú a klientsku časť. Pre vytvorenie servera sme použili jazyk Ruby a framework *Sinatra* a klient je založený na JavaScript technologií *React*. Server bude obsluhovať požiadavky klienta. Ako prvé budeme potrebovať vyriešiť základne zobrazenie prehliadača. Ruby poskytuje jedeno z rozhraní určených pre jednoduchú konfiguráciu webového servera, rozhranie *Rack*. Základnou konfiguráciu pre Rack je konfigurácia súboru *config.ru* v koreňovom priečinku aplikácie. Ide o vstupný súbor, ktorý mapuje URL požiadavky z webového prehliadača na server. Okrem iného je v tomto súbore zvyčajne riešene načítanie všetkých potrebných tried a konfigurácií.

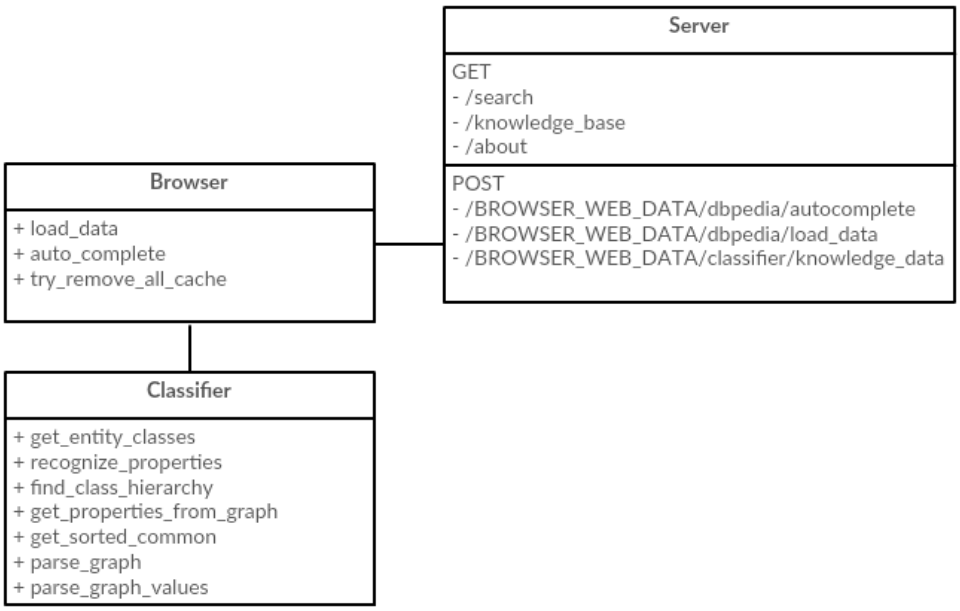
3.3.1 Architektúra servera

Pre lepšie pochopenie aké funkčnosti sú implementované na strane serveru, si v tejto časti uvedieme popis základných tried. Diagram tried servera je zobrazený na obrázku 3.2.

3.3.2 Charakteristika tried servera

Hlavnou triedou servera je trieda *Server* ktorá definuje samotný server našej aplikácie. V rámci neho sú definované tri hlavné požiadavky typu POST. Vstupom pre všetky požiadavky je JSON so základnou štrukúrov zobrazenou na obrázku 3.3. Na prefixe adresy */BROWSER_WEB_DATA/* sú dostupné tieto volania:

dbpedia/autocomplete Vstupom požiadavku je JSON základnej štruktúry s kľúčom *find_value* a hodnotou reťzca, pre ktorý majú byť nájdené ná-



Obr. 3.2: Základný diagram tried servera

```
{
  "data": {
    "body": {
      KEY: VALUE
    }
  }
}
```

Obr. 3.3: Základná štruktura JSON požiadavku na server

vrhy. Návratovou hodnotou je JSON základnej štruktúry s kľúčom *items* a hodnotou je pole objektov. Tieto objeky obsahujú kľúče a hodnoty *label* s reťazcom pre označenie zdroja a *uri* s reťazcom URI zdroja.

dbpedia/load_data Vstupom požiadavku je JSON základnej štruktúry s kľúčom *dbpedia_resource* a hodnotou reťzca, s URI zdroja, pre ktorý majú byť zobrazené dôležité informácie. Návratou hodnotou je JSON základnej štrukúry s kľúčmi a hodnotami *entity* s reťazcom najkonkretnejšej triedy ontologie, do ktoréj daný zdroj patrí, *entities* s zoradeným poľom reťazcov hierarchie tried ontológie daného zdroja, *properties* s zoradeným poľom výsledkov pre zobrazenie a *common* s zoradeným poľom všeobecne dôležitých informácií pre zobrazenie.

classifier/knowledge_data Vstupom požiadavku je prázdny JSON základnej štruktúry. Návratovou hodnotou je báza znalosti vo formáte JSON,

tak ako bola popísaná na obrázku 2.19.

V rámci triedy *Browser* sú definované 2 hlavné metódy ako implementácia obstarávania požiadavkov *dbpedia/autocomplete* a *dbpedia/load_data* na servery. Trieda zároveň poskytuje použitie metód inštancie triedy *Classifier* pomocou atribútu *classifier*. Trieda zároveň vytvára dočasné súbory (angl. cache) s výsledkami jednotlivých zdrojov, pre zníženie záťaže a zrýchlenie prehľadávania.

Trieda *Classifier* rieši všetky potrebné operácie potrebné k získaniu tripletov požadovaného zdroja, aplikácií výberu dôležitých informácií a konštrukcie výsledku v podobe vhodnej pre zobrazenie na strane klienta. V rámci potrebných dotazov na stranu DBpedia endpointu využíva existujúcu triedu nástroja *SPARQLRequest*. V rámci tejto triedy nástroja boli pre potreby preľadača definované ďalšie dotazy:

Získanie všetkých tried zdroja. Vstupom pre dotaz je zdroj ako *resource*. Vrátia sa všetky triedy aké ma zdroj priradené.

```
SELECT DISTINCT ?entity_class
WHERE {
  #{resource}   a ?entity_class .
  ?entity_class a owl:Class .
}
```

Obr. 3.4: SPARQL dotaz pre získanie tried zdroja

Získanie všetkých vlastnosti zdroja. Vstupom pre dotaz je zdroj ako *resource*. Vrátia sa všetky predikáty ako *predicate* s označením *predicate_label* a hodnoty ako *value* s označením *value_label*. Dotaz zároveň umožnuje filtrovanie na základe jazyka, parameter *lang*, pričom základnou hodnotou je *en*.

```
PREFIX dbo:   <http://dbpedia.org/ontology/>
PREFIX dbp:   <http://dbpedia.org/property/>
SELECT DISTINCT ?predicate, ?predicate_label, ?value, ?value_label
WHERE {
  { #{resource} ?predicate ?value . } UNION { ?value ?predicate #{
      resource} . }

  OPTIONAL{
    ?value rdfs:label ?value_label .
    FILTER (lang(?value_label) = '#{lang}')
  }

  ?predicate rdfs:label ?predicate_label .
  FILTER (lang(?predicate_label) = '#{lang}')
}
```

Obr. 3.5: SPARQL dotaz pre získanie vlastnosti zdroja

V rámci návratovej hodnoty pre získanie vlastnosti zdroja sú jednotlivé výsledky mapované na všetky triedy zdroja, ku ktorým patrí. Týmto spôsobom sa pri jednom požiadavku na stranu klienta dostanú všetky údaje potrebné k prepínaniu pod akou triedou sú zobrazené výsledky zdroja. Hlavným dôvodom tohto riešenia je fakt, že v procese získavania vlastnosti už metóda pozná všetky vlastnosti. Preto je veľmi jednoduché ich namapovať na všetky triedy. Týmto sa zároveň zrýchľuje prezentácia výsledkov na strane klienta.

3.3.3 Architektúra klienta

Klient webovej aplikácie je postavený predovšetkým na technológií *JavaScriptu* s využitím *REACTu*. Kaskadové štýly sú vytvorené pomocou frameworku *Sass*. Pre potreby požitia ďalších balíčkov je použitý balíčkovacieho nástroj *npm*. JavaScriptové súbory sú písané s použitím JSX harmony. Použité sú aj niektoré konštrukcie vyplývajúce z definície *ECMAScript6*. Keďže je klientský projekt vzhľadom na použité technológie zložitejší je aj jeho štruktúra rozdelená na vývojovú a produkčnú časť. V rámci vývojovej časti sú predovšetkým jednotlivé REACT komponenty a kaskadové štýly. Pri tvorbe štýlov je použitý prístup návrhu *Mobile First* pre zabazpečenie podpory zobrazenia webovej aplikáciena rôzných zariadeniach. Medzi ďalšie použité frameworky patrí taktiež Bootstrap. JavaScriptové súbory sú pomocou nástroja *babel* kompilované na čistý interpretovateľný JavaScript. Výsledne súbory sú minifikované. Pre prácu s úlohami kompilácie a minifikácie je použitý nástroj *Grunt*. Zobrazenie základných komponent klienta je na obrázku 3.6.

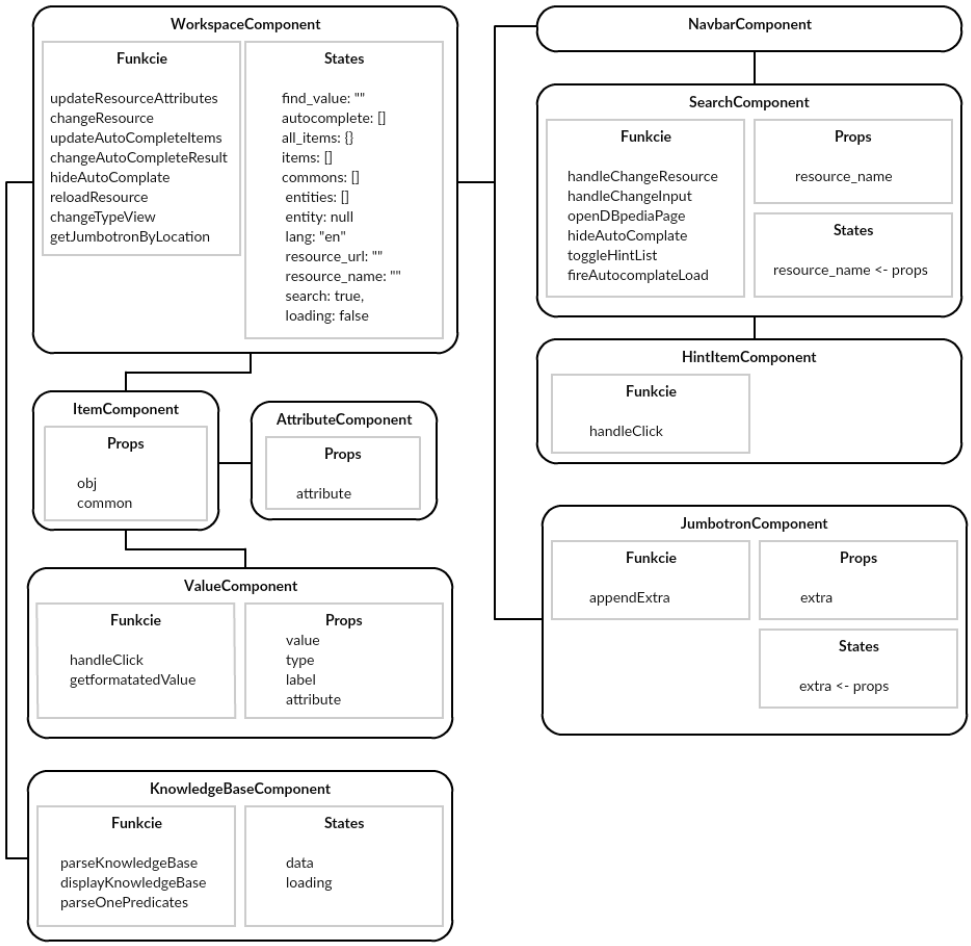
3.3.4 Charakteristika komponent klienta

Každá z komponent uvedených na obrázku 3.6 zabezpečuje vykreslenie určite časti prehliadača.

WorkspaceComponent Predstavuje celú oblasť prehliadača. Zabečpečuje pripájanie a vykreslenie podradených komponent na základe aktuálných potrieb. V prípade, že nie sú zobrazené informácie žiadného zdroja, je zobrazená informačná komponenta *JumbotronComponent*. V spodnej časti je zobrazená hierarchia tried zdroja. Zároveň je umožnené prepínanie medzi danými triedami.

NavbarComponent Definuje len spôsob vykreslenia navigačnej lišty. V naviagčnej lište sú prepnutia stránok a vyhľadávace pole *SearchComponent*.

SearchComponent Komponenta umožnuje hlavnú interakciu s používateľom. Obsahuje vstupné pole pomocou, ktorého vyhľadáva zdroj na DBpedií. Poskytuje zároveň našepkávač pre zadaný výraz. Našepkávač je vytvorený zo získaných možnosti na základe callback volania požiadavkom *dbpedia/autocomplete* na server.



Obr. 3.6: Základná štruktúra komponent klienta

HintItemComponent Predstavuje jednu položku z našepkávača. Zobrazuje textové označenie, ktoré je priradené určitej URI zdroja. Ak si používateľ zvoli zdroj na základe tejto koponenty, premietne sa URI zdroja do komponenty *SerarchComponent* a automatiky sa spustí vyhľadávanie.

KnowledgeBaseComponent Zabezpečuje vykresenie bázy znalosti. Sú zobrazené príslušne triedy ontológie a v rámci nich predikáty spolu s ich hodnotením dôležitosti.

ItemComponent Predstavuje jednotlivé záznami vlastnosti určené pre vykreslenie. V rámci komponenty sa rozlišuje, či sa jedná o všeobecne dôležitú komponentu tzv. *common*. Ak je to *common* bude vykreslená v pravej časti stránky, pričom nebude vykreslený názov vlastnosti, ale len samotná hodnota. V prípade, že je to bežná vlastnosť bude vykreslená

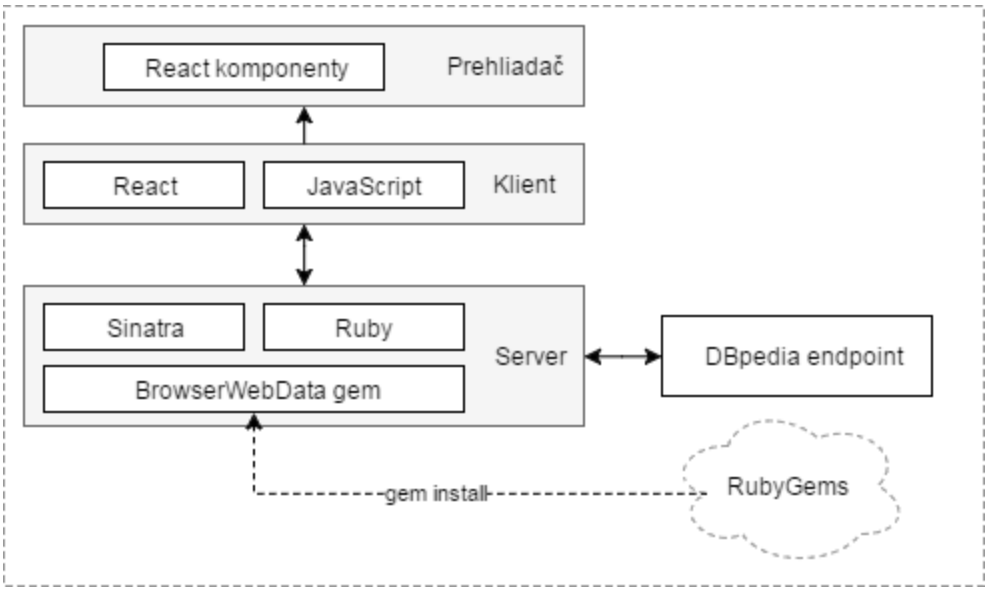
v hlavnej časti. Komponenta vykresľuje označenie vlastnosti pomocou komponenty *AttributeComponent* a hodnoty pomocou *ValueComponent*.

AttributeComponent Zabezpečuje spôsob vykreslenie označenia vlastnosti.

ValueComponent Zabezpečuje spôsob vykreslenie hodnôt vlastnosti. Pre jednu vlastnost môže byť priradeno viacero hodnôt, v takom prípade sa vždy zobrazí len ptvých 5 hodnôt. Všetky ďalšie sú skryté. Ich zobrazenie je možne pomocou tlačidla. Vykreslenie hodnôt zároveň aplikuje formatovanie na základe prideleného typu (napr. číslo, dátum a ďalšie).

3.4 Architektúra prehliadača

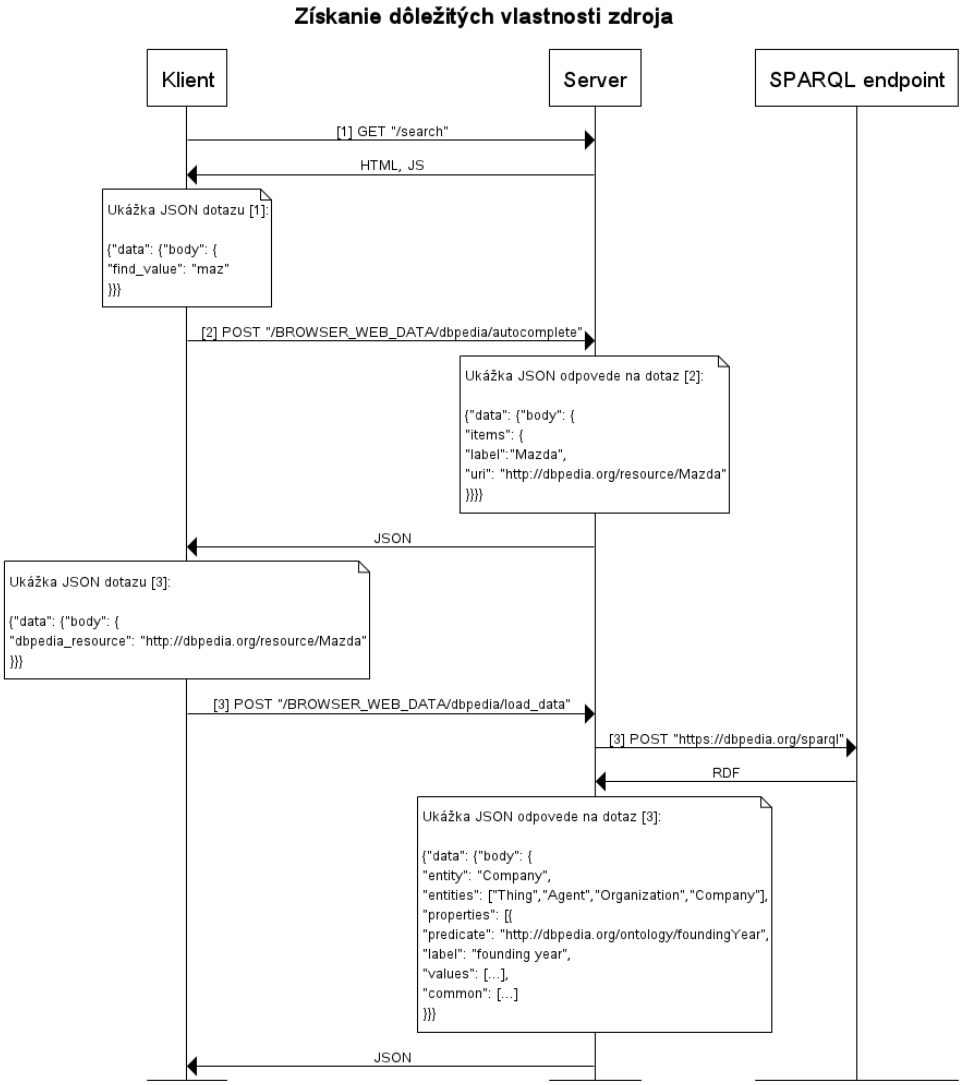
Prehliadač obsahuje 3 základne stránky. Úvodnou a hlavnou stránkou je */search*, ktorá umožnuje prehliadanie informácií zdrojov. Ďalšou stránkou je */knowledge_base*. Na tejto stránke je prehľadnou formou zobrazená používana báza znalosti. Poslednou stránkou je stránka */about* so základnými informáciami o prehliadači ako aj celom projekte. Základna architektúra celej webovej aplikácie je znazornená na obrázku 3.7.



Obr. 3.7: Základna architektúra webového prehliadača

Základna komunikácia medzi klientom a serverom za účelom získania dôležitých informácií je zobrazená na obrázku 3.8. Webová aplikácia bola, okrem lokálného testovania, nasadená pomocou služby *Heroku*. Tento proces bol riadený pomocou technológie *git*. Repozitár s aplikáciou, ktorý bol vytvorený v

službe Heroku sa pri operácií *push* nasadil ako celá webová aplikícia. Tým sa zároveň overilo jej reálne použitie v praxi.



Obr. 3.8: Základna komunikácia klienta so serverom

Kapitola 4

Experimenty a vyhodnotenie

V tejto kapitole sa pozrieme na získané výsledky. Hlavným výsledkom je vygenerovaná báza znalosti. Obsahuje zoradené predikáty na základe ich skóre. Na základe získaných hodnotení predikátov vykonáme porovnanie voči iným existujúcim riešeniam. Budeme hlavne porovnávať pozície jednotlivých predikátov.

Pozičné porovnanie si vyžiadalo zvoliť prístup akým budú jednotlivé umiestnenia naších výsledkov porovnané voči iným riešeniam. Porovnanie bude vychádzať z prechodu zoradeným zoznamom predikátov iných riešení. Pre každý predikát bude nájdená pozícia v rámci našej bázy znalosti. Pri niektorých riešeniach bude nutné premapovanie predikátov. Znamená to, aký predikát zdrojov DBpedie odpovedá svojím popisom predikátu daného riešenia. Pre stanovenie skóre identickosti umiestnenia sme si definovali vzťah, v ktorom je znížovaná presnosť v miere relatívneho vzdialenia sa nášho predikátu. Nech je c_o počet overovaných predikátov cudzieho riešenia, i_{p_a} index umiestnenia predikátu p cudzieho riešenia a i_{p_b} index umiestnenia predikátu p nášho riešenia potom skóre miery identickosti umiestnenia nášho predikátu v báze znalosti definujeme pomocou vzťahu znazornenom na obrázku 4.1.

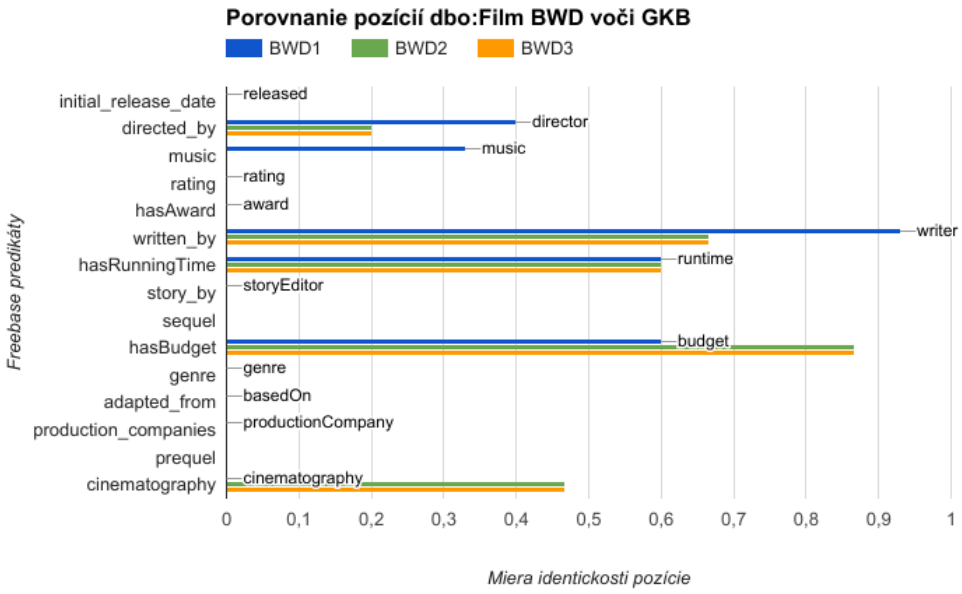
$$S_p = 1 - \frac{|i_{p_a} - i_{p_b}|}{c_o}$$

Obr. 4.1: Definícia miery identickosti pozície predikátu p.

Všetky porovnania sú realizované jednotným spůsobom. Porovnania sú reprezentované pomocou grafov. Prvým typom grafu je graf porovnania miery identickosti pozície predikátov. Druhým je graf porovnania miery identickosti hodnotenia dôležitosti predikátov. Za kompletne porovnanie je možne považovať premernú hodnotu z identickosti pozícií predikátov. Budeme ho preto označovať pod názvom *kompletné hodnotenie*, budeme ho uvádzať ako percentá. Vrámci experimetu sme zároveň porovnávali 3 druhy vygenerovaných báz znalosti. Prvou je kompletná báza znalosti so všetkým procesmi. Druhou

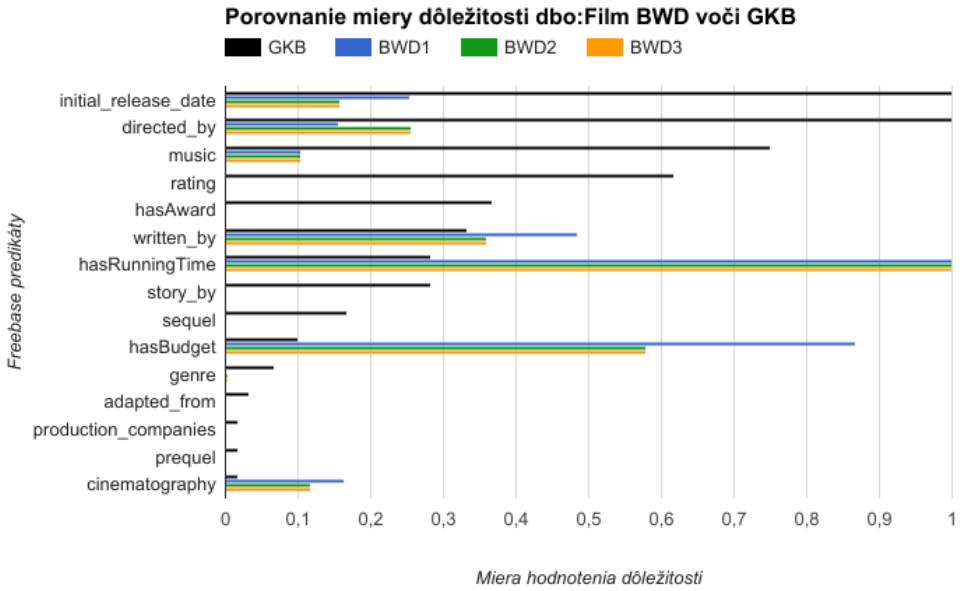
je báza znalosti vygenerovaná na základe iba striktných predikátov a identifikácií identických predikátov. Poslednou je pri použití striknych predikátov a triedných predikátov (podobne ako v hlavnej báze znalosti), ale bez identifikácie identických predikátov. Porovnávalo sa prvých 15 predikátov. Výsledky našej práce sú označené ako BrowserWebData (ďalej ako BWD).

Prvým overením výsledkov je porovnanie voči projektu založenom na sumarizácií entít pomocou vedomostnej hry (popísaná v kapitole 1.5.2). Riešenie používalo predikáty z datasetu Freebase. Prehľadná tabuľka výsledkov danej práce je na obrázku 1.3. Tie niesú zhodné s predikátmi použitými v DBpedií. Vytvorili sme si preto mapovanie ktoré je zobrazené v tabuľke B.1 prílohy B. Porovnávanými výsledkymi sú hodnotenia triedy `http://dbpedia.org/ontology/Film`. Vo svojej práci overovali výsledky voči *Google knowledge base*. Preto sme pomocou ich práce realizovali porovnanie voči ich výsledkom pod označením UBES a Google knowledge base pod označením GKB. Porovnanie pozícií BWD voči GKB je na obrázku 4.2, porovnanie miery dôležitosti BWD voči GKB je na obrázku 4.3. Kompletným hodnotením je 22%, je rovnakým pre všetky 3 druhy uvažovaných báz.

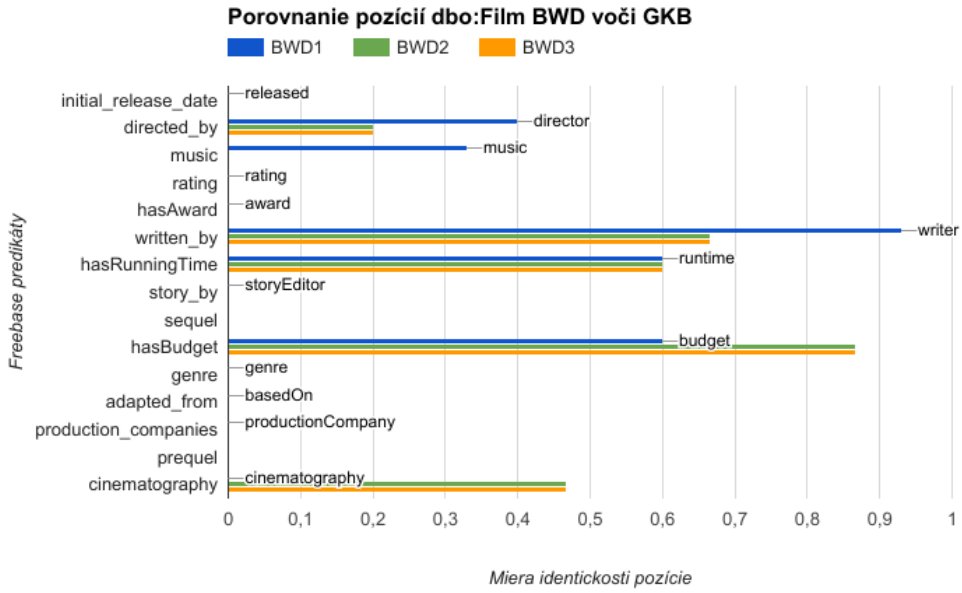


Obr. 4.2: Graf porovnania pozícií BWD voči GKB pre triedu Film

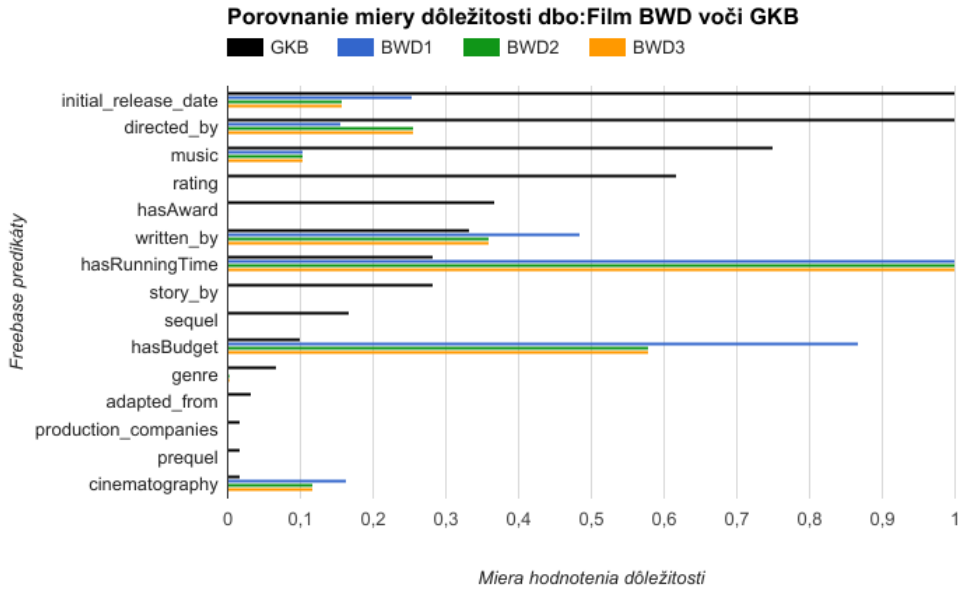
Porovnanie pozícií BWD voči UBES je na obrázku 4.4, porovnanie miery dôležitosti BWD voči UBES je na obrázku 4.5. Kompletným hodnotením je 37%, pre kompletnú bázu znalostí a 19% pre ďalšie 2 z experimentálných báz.



Obr. 4.3: Graf porovnania miery dôležitosti BWD voči GKB pre triedu Film



Obr. 4.4: Graf porovnania pozícií BWD voči UBES pre triedu Film



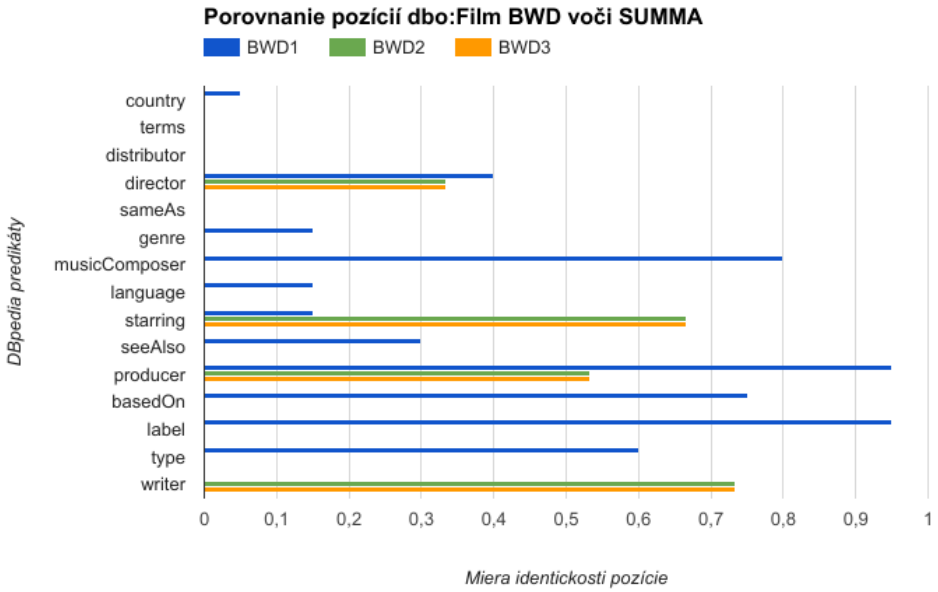
Obr. 4.5: Graf porovnania miery dôležitosti BWD voči UBES pre triedu Film

Poslednou prácou, ktorej výsledky porovnáme s našími je API rozhranie pre sumarizaciu (popísaná v kapitole 1.5.3). Tieto výsledky sú ďalej označené ako SUMMA. Pomocou ich API sme postupne získali hodnotenia zvolených zdrojov. Porovnanávali sme viacero tried. Zvolili sme si rovnakú triedu ako v prvom porovnaní `http://dbpedia.org/ontology/Film`. Ďalej sme si zvolili 2 triedy `http://dbpedia.org/ontology/Person` a jej podtriedu `http://dbpedia.org/ontology/President`. Porovnanie podtriedy je za účelom demonštrácie zmeny dôležitosti predikátov pri viac šepecifickom zaradení zdroja.

Porovnanie pozícií triedy `http://dbpedia.org/ontology/Film` BWD voči SUMMA je na obrázku 4.6, porovnanie miery dôležitosti BWD voči SUMMA je na obrázku 4.7. Kompletným hodnotením je 35%, pre kompletnú bázu znalostí a 15% pre ďalšie 2 z experimentálných báz.

Porovnanie pozícií triedy `http://dbpedia.org/ontology/Person` BWD voči SUMMA je na obrázku 4.8, porovnanie miery dôležitosti BWD voči SUMMA je na obrázku 4.9. Kompletným hodnotením je 12%, pre kompletnú bázu znalostí, 14% pre striktných predikátov a 11% u bázy bez použitie identifikácie identických perdikátov.

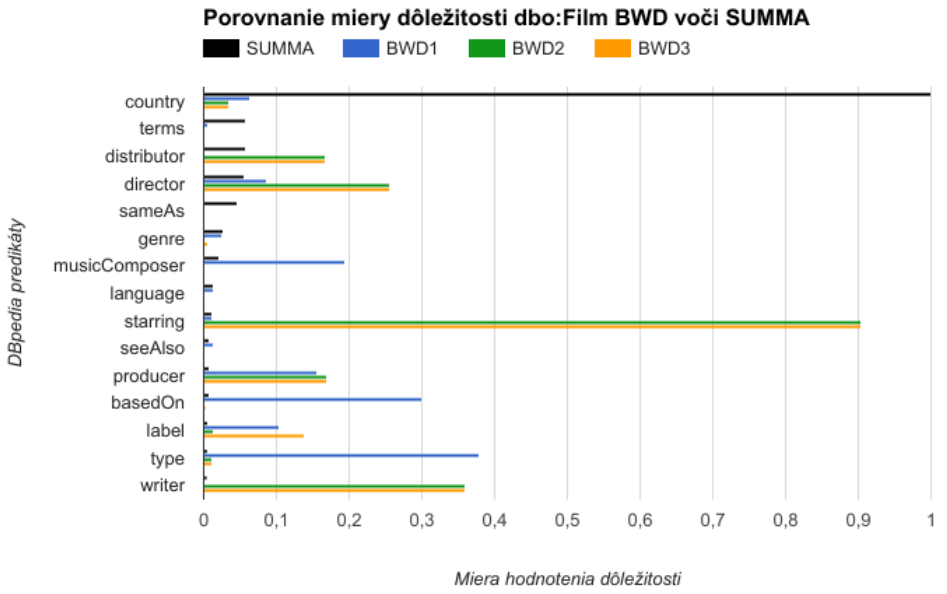
Porovnanie pozícií triedy `http://dbpedia.org/ontology/President` BWD voči SUMMA je na obrázku 4.10, porovnanie miery dôležitosti BWD voči SUMMA je na obrázku 4.11. Kompletným hodnotením je 12%, pre kompletnú bázu znalostí a striktné predikáty. Pre bázu bez použitia identifikácie identic-



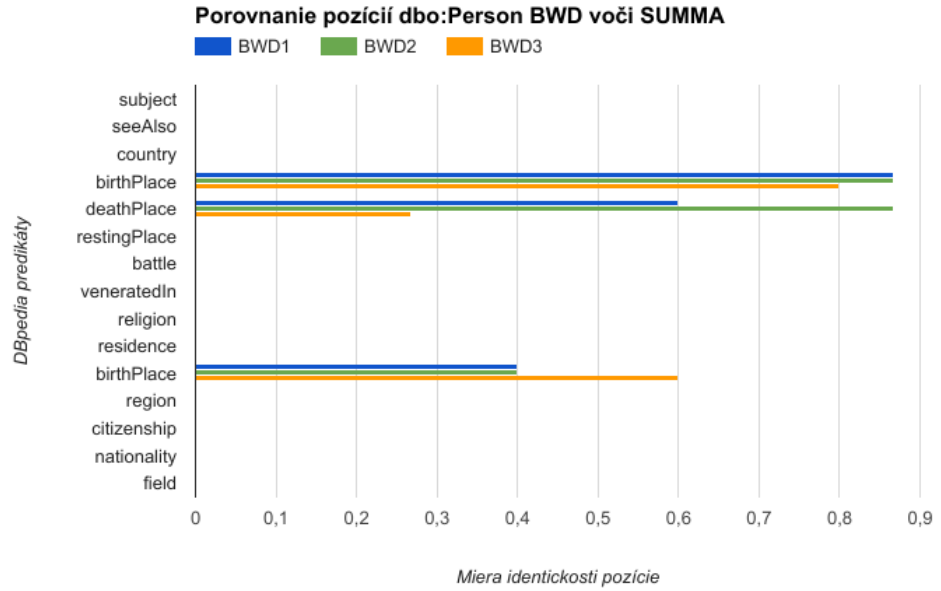
Obr. 4.6: Graf porovnania pozícií BWD voči SUMMA pre triedu Film

kých perdikátov 8%.

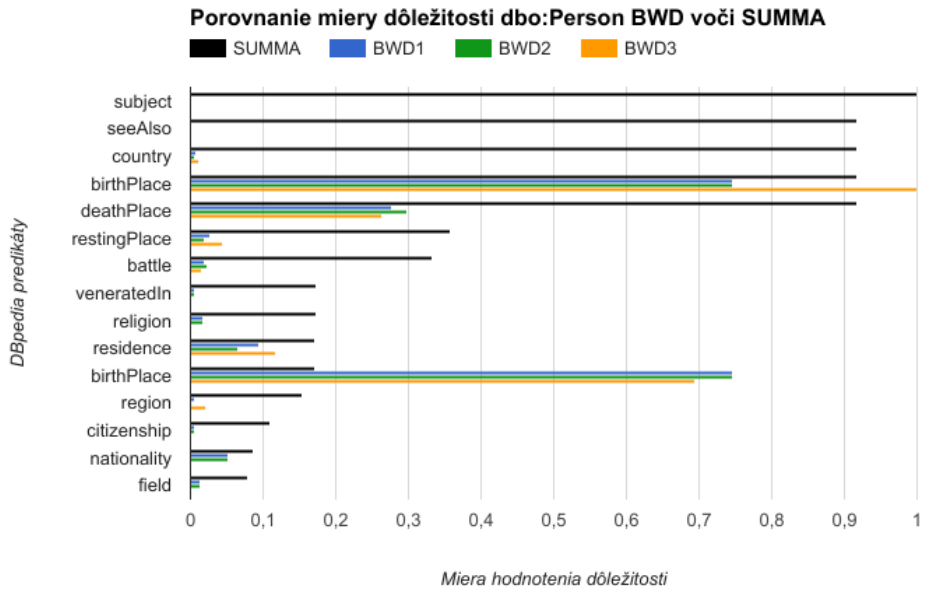
Zhrnutím celého porovnania je skutočnosť, že pri použití kompletnej bázy znalosti získavame identickejšie pozície voči iným riešeniam. Zároveň platí, že bez použitia identifikácie identických predikátov sa znižuje presnosť pozícií. Aj napriek nízkym kompletným hodnoteniam pokladáme naše riešenia za dostatočne úspešne. Zároveň napríklad u výsledkov SUMMA sú vo výsledkoch aj duplicity, ktoré negatívne vplývajú na porovnanie pozícií.



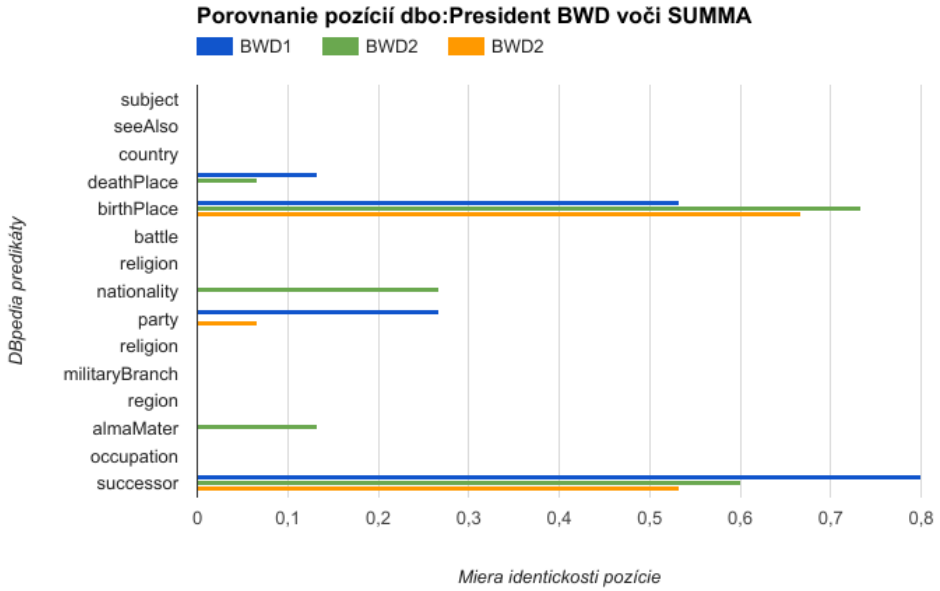
Obr. 4.7: Graf porovnania miery dôležitosti BWD voči SUMMA pre triedu Film



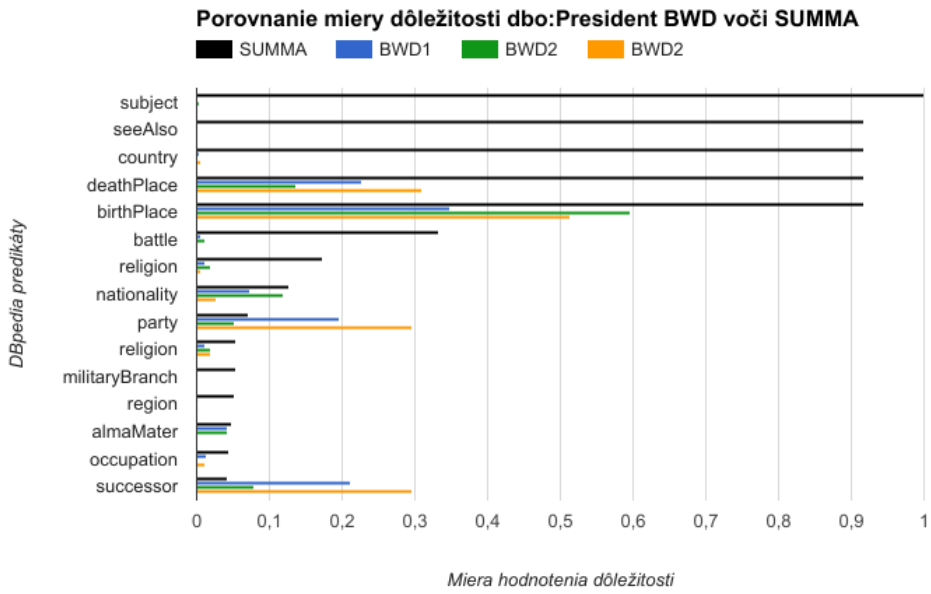
Obr. 4.8: Graf porovnania pozícií BWD voči SUMMA pre triedu Person



Obr. 4.9: Graf porovnania miery dôležitosti BWD voči SUMMA pre triedu Person



Obr. 4.10: Graf porovnania pozícií BWD voči SUMMA pre triedu President



Obr. 4.11: Graf porovnania miery dôležitosti BWD voči SUMMA pre triedu President

Záver

V závere práce by sme radi zhodnotli získané výsledky práce. Pozreli sme sa na problematiku sumarizácie entít, identifikovali podobné riešenia. V časti návrhu sme podrobnejšie rozobrali a navrhli vlastný model pre sumarizáciu entít na základe NIF DBpedia abstract datasetu. Model bol aplikovaný pri vytvorení nástroja. Nástroj sme publikovali pomocou verejného manažéra balíčkov *RubyGems*. Je dostupny na adrese `https://rubygems.org/gems/browser_web_data_entity_sumarization`. Spôsob ako použiť nástroj sme v krátkosti uviedli v tejto práci. Podrobnejší návod je súčasťou publikovaného kódu nástroja spolu s našími výsledkami na adrese `https://github.com/MarekFiltes/entity_sumarization`. Navrhli a implementovali sme webovú aplikáciu založenu na *Ruby* frameworku *Sinata*. Klient webovej aplikácie je založený na technológií *React*. Webová aplikácia aplikuje získane výsledky z bázy znalosti. Poskytli sme zároveň vedľajšie výsledky práce. Najpodstatnejším je zoznam identických predikátov. Pomocou neho je možne znížiť mieru šumových duplicitných predikátov.

Medzi zlepšenia práce patrí rozšírenie použitých tried ontológie o triedy z iných ontológií, napríklad triedy *yago*. Zlepšenie procesu identifikácie identických predikátov. Často sa aktuálne nezaradia ako identické, predikáty, ktorých hodnoty sú rozdielným spôsobom priradené, ale majú rovnaký význam. Možným rozšírením práce do budúcna je poskytnutie Web API pre získanie zoznamu dôležitých predikátov a ich hodnôt.

Všetky ciele, ktoré sme si v úvode stanovili pokladáme za splnené. Naše riešenie poskytuje model a nástroj určený pre zlepšenie prehľadnosti informácií zdrojov na DBpedií. Rieší problém identifikácie, ktoré z informácií sú dôležitejšie než iné. Pri prehliadaní informácií na základe našej bázy znalosti sa znižuje miera náročnosti identifikácie informácií používateľom. Zlepšuje sa použitie informácií zdrojov DBpedie pre bežného používateľa, ktorý nepotrebuje všetky informácie. Výsledná báza znalostí ma široký záber. Obsahuje predikáty mapované na všetky triedy DBpedia ontológie.

Literatúra

[1] Mani, I. *Automatic Summarization*. Natural Language Processing, John Benjamins Publishing Company, 2001, ISBN 9789027299109. Available from: https://books.google.cz/books?id=CSHxUOfb5bwC

[2] Page, L.; Brin, S.; Motwani, R.; et al. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999, previous number = SIDL-WP-1999-0120. Available from: http://ilpubs.stanford.edu:8090/422/

[3] Thalhammer, A.; Stadtmüller, S. *SUMMA: A Common API for Linked Data Entity Summaries*. Cham: Springer International Publishing, 2015, ISBN 978-3-319-19890-3, pp. 430–446, doi:10.1007/978-3-319-19890-3_28. Available from: http://dx.doi.org/10.1007/978-3-319-19890-3_28

[4] LLC, M. NIF Abstract Datasets. 2016, [cit. 2017-05-01]. Available from: http://wiki.dbpedia.org/nif-abstract-datasets

[5] Kahn, B. K.; Strong, D. M.; Wang, R. Y. Information Quality Benchmarks: Product and Service Performance. *Commun. ACM*, volume 45, no. 4, Apr. 2002: pp. 184–192, ISSN 0001-0782, doi:10.1145/505248.506007. Available from: http://doi.acm.org/10.1145/505248.506007

[6] Thalhammer, A.; Knuth, M.; Sack, H. *Evaluating Entity Summarization Using a Game-Based Ground Truth*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ISBN 978-3-642-35173-0, pp. 350–361, doi:10.1007/978-3-642-35173-0_24. Available from: http://dx.doi.org/10.1007/978-3-642-35173-0_24

[7] DBpedia version 2016-04. 2016, [cit. 2017-05-02]. Available from: http://wiki.dbpedia.org/dbpedia-version-2016-04/OnlineAccess

[8] Cover, T. M.; Thomas, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006, ISBN 0471241954.

[9] Šlapák, O. Data, informace, znalosti. *E-LOGOS ELECTRONIC JOURNAL FOR PHILOSOPHY*, 2003.

[10] Hoffmannová, J. *Stylistika a: současná situace stylistiky*. Trizonia, 1997, ISBN 9788085573671. Available from: https://books.google.cz/books?id=2qjlAAAAMAAJ

[11] Stalnaker, R. *Context and Content: Essays on Intentionality in Speech and Thought*. Oxford cognitive science series, Oxford University Press, 1999, ISBN 9780198237082. Available from: https://books.google.cz/books?id=OhhMRF2dz0IC

[12] Jaromír, J. *Psychologické základy verbální komunikace*. 2015, ISBN 9788024798509.

[13] Ding, L.; Pan, R.; Finin, T.; et al. *Finding and Ranking Knowledge on the Semantic Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, ISBN 978-3-540-32082-1, pp. 156–170, doi:10.1007/11574620_14. Available from: http://dx.doi.org/10.1007/11574620_14

[14] Lachica, R.; Karabeg, D.; Rudan, S. Quality, relevance and importance in information retrieval with fuzzy semantic networks. *Proc. TMRA*, 2008.

[15] Kagolovsky, Y.; Mohr, J. R. A new approach to the concept of"relevanceïn information retrieval (IR). *Studies in health technology and informatics*, , no. 1, 2001: pp. 348–352.

[16] Grešková, M. Empirickỳ vỳskum interakcie človeka s agentom. *Nová paradigma spracovania a využívania informácií*, 2007: p. 24.

[17] Radev, D. R.; Hovy, E.; McKeown, K. Introduction to the Special Issue on Summarization. *Comput. Linguist.*, volume 28, no. 4, Dec. 2002: pp. 399–408, ISSN 0891-2017, doi:10.1162/089120102762671927. Available from: http://dx.doi.org/10.1162/089120102762671927

[18] Steinberger, J.; Ježek, K. Text summarization and singular value decomposition. In *International Conference on Advances in Information Systems*, Springer, 2004, pp. 245–254.

[19] *Sémantika programovacích jazyků*. Skripta UK, 1997, ISBN 80-7184-327-X.

[20] Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Scientific American*, volume 284, no. 5, 2001: pp. 34–43, ISSN 0036-8733. Available from: http://www.scientificamerican.com/article.cfm?id=the-semantic-web

[21] RDF Primer. 2014, [cit. 2017-04-14]. Available from: https://www.w3.org/TR/2004/REC-rdf-primer-20040210/

[22] SPARQL Query Language for RDF. 2008, [cit. 2017-04-16]. Available from: https://www.w3.org/TR/rdf-sparql-query/

[23] Svátek, V. Ontologie a WWW. In *Sborník konference Datakon*, 2002, pp. 27–55.

[24] Berners-Lee, T. Linked Data. 2006, [cit. 2017-04-17]. Available from: http://www.w3.org/DesignIssues/LinkedData.html

[25] Bizer, C.; Heath, T.; Berners-Lee, T. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, volume 5, no. 3, 2009: p. 1–22.

[26] Wikipedie. 2014, [cit. 2017-04-16]. Available from: https://cs.wikipedia.org/wiki/Wikipedie

[27] Auer, S.; Bizer, C.; Kobilarov, G.; et al. Dbpedia: A nucleus for a web of open data. *The semantic web*, 2007: pp. 722–735.

[28] DBpedia. 2017, [cit. 2017-05-02]. Available from: https://en.wikipedia.org/wiki/DBpedia

[29] Cheng, G.; Tran, T.; Qu, Y. *RELIN: Relatedness and Informativeness-Based Centrality for Entity Summarization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ISBN 978-3-642-25073-6, pp. 114–129, doi:10.1007/978-3-642-25073-6_8. Available from: http://dx.doi.org/10.1007/978-3-642-25073-6_8

[30] Brümmer, M.; Dojchinovski, M.; Hellmann, S. DBpedia Abstracts: A Large-Scale, Open, Multilingual NLP Training Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, edited by N. C. C. Chair); K. Choukri; T. Declerck; M. Grobelnik; B. Maegaard; J. Mariani; A. Moreno; J. Odijk; S. Piperidis, Paris, France: European Language Resources Association (ELRA), May 2016. Available from: https://svn.aksw.org/papers/2016/LREC_DBpedia_Abstracts/public.pdf

[31] Hellmann, S.; Lehmann, J.; Auer, S. NIF: An ontology-based and linked-data-aware NLP Interchange Format. *Working Draft*, 2012.

[32] Hellmann, S.; Lehmann, J.; Auer, S.; et al. Integrating NLP using linked data. In *International Semantic Web Conference*, Springer, 2013, pp. 98–113.

[33] Accessing the DBpedia Data Set over the Web. 2016, [cit. 2017-05-01]. Available from: `http://wiki.dbpedia.org/OnlineAccess`

[34] Auer, S.; Bizer, C.; Kobilarov, G.; et al. *DBpedia: A Nucleus for a Web of Open Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, ISBN 978-3-540-76298-0, pp. 722–735, doi:10.1007/978-3-540-76298-0_52. Available from: `http://dx.doi.org/10.1007/978-3-540-76298-0_52`

[35] Thalhammer, A.; Rettinger, A. PageRank on Wikipedia: Towards General Importance Scores for Entities. In *The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 – June 2, 2016, Revised Selected Papers*, Cham: Springer International Publishing, Oct. 2016, ISBN 978-3-319-47602-5, pp. 227–240, doi:10.1007/978-3-319-47602-5_41. Available from: `http://dx.doi.org/10.1007/978-3-319-47602-5_41`

[36] DBpedia 2016-04 Statistics. 2016, [cit. 2017-05-03]. Available from: `http://wiki.dbpedia.org/dbpedia-2016-04-statistics`

[37] About Ruby. 2017, [cit. 2017-05-05]. Available from: `https://www.ruby-lang.org/en/about`

[38] What is gem. 2017, [cit. 2017-05-05]. Available from: `http://guides.rubygems.org/what-is-a-gem`

Dodatok A

Obsah priloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `exe` ..... adresár so spustiteľnou formou implementácie
- `src`
  - `impl` ..... zdrojové kódy implementácie
  - `thesis` ..... zdrojová forma práce vo formáte LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce vo formáte PDF

Dodatok B

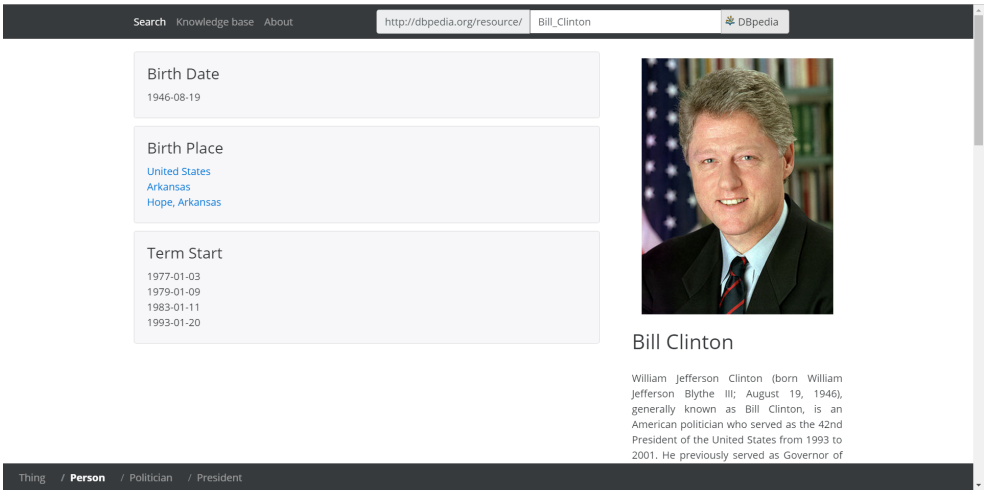
Doplňujúce informácie testovania

Tabuľka B.1: Tabuľka mapovania Freebase predikatov na DBpedia predikáty

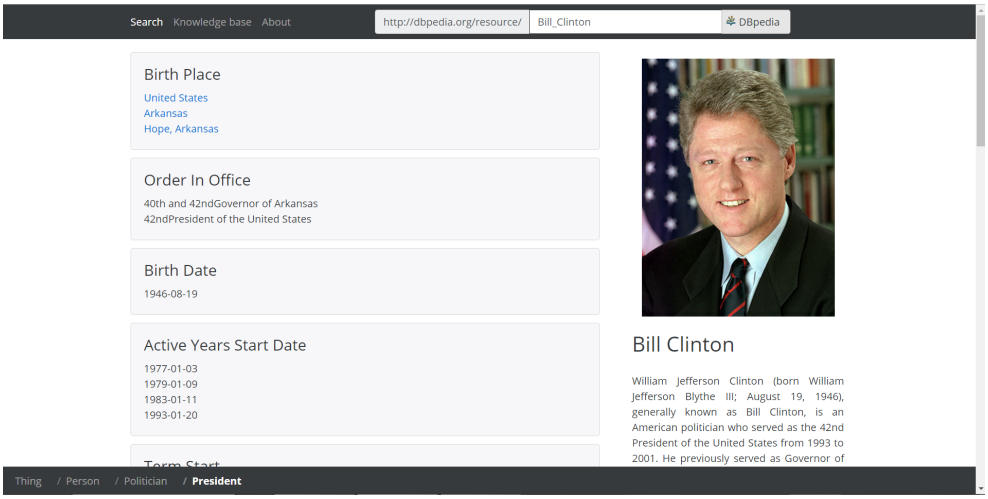
Predikáty Freebase	**Predikáty DBpedia**
http://rdf.freebase.com/ns/film.film.initial_release_date	http://dbpedia.org/ontology/releaseDate
http://rdf.freebase.com/ns/film.film.directed_by	http://dbpedia.org/ontology/director
http://rdf.freebase.com/ns/film.film.music	http://dbpedia.org/property/music
http://rdf.freebase.com/ns/film.film.rating	http://dbpedia.org/ontology/rating
http://test.com#hasAward	http://dbpedia.org/ontology/award
http://rdf.freebase.com/ns/film.film.written_by	http://dbpedia.org/ontology/writer
http://test.com#hasRunningTime	http://dbpedia.org/ontology/runtime
http://rdf.freebase.com/ns/film.film.story_by	
http://rdf.freebase.com/ns/film.film.sequel	
http://test.com#hasBudget	http://dbpedia.org/ontology/budget
http://rdf.freebase.com/ns/film.film.genre	http://dbpedia.org/ontology/genre
http://rdf.freebase.com/ns/media_common.adaptation.adapted_from	http://dbpedia.org/ontology/basedOn
http://rdf.freebase.com/ns/film.film.production_companies	http://dbpedia.org/ontology/productionCompany
http://rdf.freebase.com/ns/film.film.prequel	
http://rdf.freebase.com/ns/film.film.cinematography	http://dbpedia.org/ontology/cinematography
http://rdf.freebase.com/ns/film.film.subjects	http://purl.org/dc/elements/1.1/subject
http://rdf.freebase.com/ns/film.film.film_series	http://dbpedia.org/ontology/series
http://rdf.freebase.com/ns/film.film.film_festivals	
http://rdf.freebase.com/ns/film.film.film_casting_director	
http://rdf.freebase.com/ns/film.film.featured_film_locations	

Dodatok C

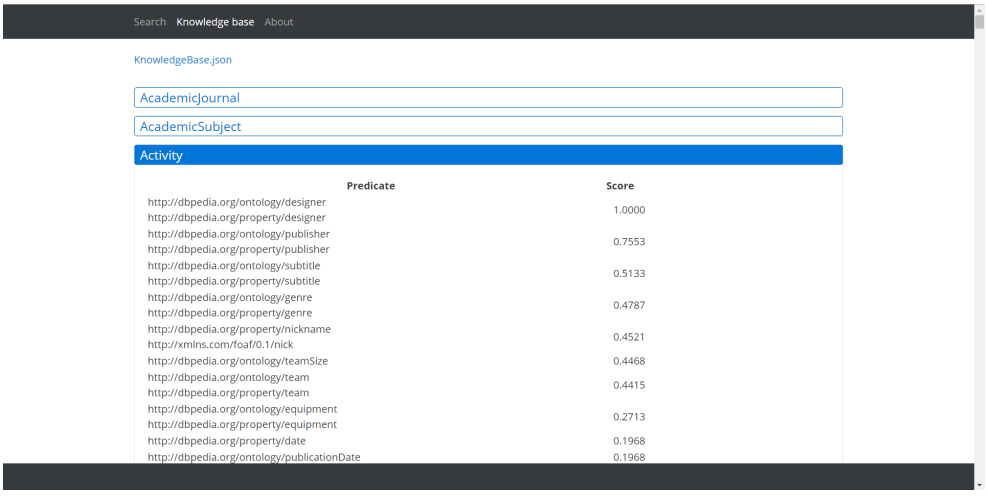
Ukážky webovej aplikácie



Obr. C.1: Ukážka zobrazenia zdroja triedy Person



Obr. C.2: Ukážka zobrazenia zdroja triedy President



Obr. C.3: Ukážka zobrazenia bázy znalosti

Dodatok D

Ukážky bázy znalostí

Tabuľka D.1: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Plant

Predikát	Skóre
http://dbpedia.org/property/r1Surface	1.0000
http://dbpedia.org/property/regnum	0.7658
http://dbpedia.org/ontology/synonym	0.7611
http://dbpedia.org/property/synonyms	0.7606
http://dbpedia.org/property/r2Surface	0.7148
http://dbpedia.org/ontology/genus http://dbpedia.org/property/genus	0.5003
http://dbpedia.org/ontology/binomialAuthority http://dbpedia.org/property/binomialAuthority	0.2135
http://dbpedia.org/property/statusSystem	0.1673
http://dbpedia.org/ontology/conservationStatusSystem	0.1673
http://dbpedia.org/ontology/conservationStatus	0.1661
http://dbpedia.org/ontology/status http://dbpedia.org/property/status	0.166
http://dbpedia.org/ontology/plant	0.1069
http://dbpedia.org/property/plant	0.0764
http://dbpedia.org/ontology/species http://dbpedia.org/property/species	0.0589
http://dbpedia.org/property/unrankedClassis	0.0525
http://dbpedia.org/property/unrankedDivisio	0.0493
http://dbpedia.org/ontology/ingredient	0.0474
http://dbpedia.org/property/subgenus	0.0408
http://dbpedia.org/ontology/associatedBand http://dbpedia.org/ontology/associatedMusicalArtist	0.0315
http://dbpedia.org/property/unrankedOrdo	0.0289

Tabuľka D.2: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Animal

Predikát	Skóre
http://dbpedia.org/property/regnum	1.0000
http://dbpedia.org/ontology/kingdom	0.8903
http://dbpedia.org/ontology/genus http://dbpedia.org/property/genus	0.8017
http://dbpedia.org/ontology/order	0.6659
http://dbpedia.org/ontology/family http://dbpedia.org/property/familia	0.5094
http://dbpedia.org/ontology/synonym	0.4756
http://dbpedia.org/property/synonyms	0.4754
http://dbpedia.org/ontology/conservationStatus	0.4651
http://dbpedia.org/ontology/status http://dbpedia.org/property/status	0.4636
http://dbpedia.org/property/statusSystem	0.4549
http://dbpedia.org/ontology/conservationStatusSystem	0.4549
http://dbpedia.org/ontology/phylum http://dbpedia.org/property/phylum	0.3532
http://dbpedia.org/property/classis	0.3187
http://dbpedia.org/property/ordo	0.214
http://dbpedia.org/ontology/binomialAuthority http://dbpedia.org/property/binomialAuthority	0.1694
http://dbpedia.org/ontology/class http://dbpedia.org/property/class	0.1100
http://dbpedia.org/property/subordo	0.0771
http://dbpedia.org/property/superfamilia	0.0671
http://dbpedia.org/ontology/species http://dbpedia.org/property/species	0.0192
http://dbpedia.org/property/filename	0.0143

Tabuľka D.3: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Person

Predikát	Skóre
http://dbpedia.org/ontology/birthDate http://dbpedia.org/property/birthDate	0.9341
http://dbpedia.org/ontology/birthPlace http://dbpedia.org/property/birthPlace	0.7464
http://dbpedia.org/ontology/cinematography http://dbpedia.org/property/cinematography	0.7182
http://dbpedia.org/ontology/deathDate http://dbpedia.org/property/deathDate	0.5573
http://dbpedia.org/ontology/activeYearsStartYear	0.4234
http://dbpedia.org/property/placeOfBirth	0.3589
http://dbpedia.org/ontology/birthYear	0.3474
http://dbpedia.org/ontology/birthName http://dbpedia.org/property/birthName	0.3271
http://dbpedia.org/property/voices	0.3253
http://dbpedia.org/ontology/voice http://dbpedia.org/property/voice	0.3051
http://dbpedia.org/ontology/deathPlace http://dbpedia.org/property/deathPlace	0.2765
http://dbpedia.org/ontology/starring http://dbpedia.org/property/starring	0.2518
http://dbpedia.org/property/yearsActive	0.2384
http://dbpedia.org/ontology/activeYearsEndYear	0.2236
http://dbpedia.org/ontology/deathYear	0.1984
http://dbpedia.org/property/termStart	0.1759
http://dbpedia.org/property/termEnd	0.1711
http://xmlns.com/foaf/0.1/givenName	0.159
http://dbpedia.org/ontology/occupation http://dbpedia.org/property/occupation	0.1454
http://dbpedia.org/property/reign	0.1321

Tabuľka D.4: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:President

Predikát	Skóre
http://dbpedia.org/ontology/orderInOffice	1.0000
http://dbpedia.org/ontology/birthDate http://dbpedia.org/property/birthDate	0.8971
http://dbpedia.org/ontology/activeYearsStartDate	0.8636
http://dbpedia.org/property/termStart	0.8627
http://dbpedia.org/ontology/activeYearsEndDate	0.8284
http://dbpedia.org/property/termEnd	0.8275
http://dbpedia.org/property/birthname	0.5152
http://dbpedia.org/ontology/deathDate http://dbpedia.org/property/deathDate	0.4706
http://dbpedia.org/property/shortDescription http://purl.org/dc/elements/1.1/description	0.3817
http://xmlns.com/foaf/0.1/surname	0.3568
http://xmlns.com/foaf/0.1/givenName	0.3568
http://dbpedia.org/ontology/birthPlace http://dbpedia.org/property/birthPlace	0.349
http://dbpedia.org/ontology/birthYear	0.3309
http://dbpedia.org/property/dateOfBirth	0.3309
http://dbpedia.org/property/order	0.3073
http://dbpedia.org/ontology/office http://dbpedia.org/property/office	0.2377
http://dbpedia.org/ontology/deathPlace http://dbpedia.org/property/deathPlace	0.2262
http://dbpedia.org/ontology/successor http://dbpedia.org/property/successor	0.2109
http://dbpedia.org/ontology/president http://dbpedia.org/property/president	0.2015
http://dbpedia.org/ontology/party http://dbpedia.org/property/party	0.1965

Tabuľka D.5: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Monarch

Predikát	Skóre
http://dbpedia.org/ontology/title	1.0000
http://dbpedia.org/ontology/activeYearsStartYear	0.939
http://dbpedia.org/property/reign	0.9317
http://dbpedia.org/ontology/activeYearsEndYear	0.8255
http://dbpedia.org/ontology/successor http://dbpedia.org/property/successor	0.5307
http://dbpedia.org/ontology/predecessor http://dbpedia.org/property/predecessor	0.4649
http://dbpedia.org/property/shortDescription http://purl.org/dc/elements/1.1/description	0.2365
http://dbpedia.org/property/fullName	0.2346
http://dbpedia.org/ontology/deathYear	0.1664
http://dbpedia.org/property/dateOfDeath	0.0974
http://dbpedia.org/ontology/birthYear	0.0945
http://dbpedia.org/property/dateOfBirth	0.094
http://xmlns.com/foaf/0.1/surname	0.0909
http://xmlns.com/foaf/0.1/givenName	0.0909
http://dbpedia.org/ontology/deathDate http://dbpedia.org/property/deathDate	0.0821
http://dbpedia.org/ontology/birthDate http://dbpedia.org/property/birthDate	0.0472
http://dbpedia.org/ontology/alias http://dbpedia.org/property/alias	0.0339
http://dbpedia.org/property/father	0.0326
http://dbpedia.org/ontology/deathPlace http://dbpedia.org/property/deathPlace	0.0315
http://dbpedia.org/ontology/placeOfBurial http://dbpedia.org/property/placeOfBurial	0.0272

Tabuľka D.6: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Artist

Predikát	Skóre
http://dbpedia.org/ontology/birthDate http://dbpedia.org/property/birthDate	0.9832
http://dbpedia.org/ontology/background http://dbpedia.org/property/background	0.8549
http://dbpedia.org/ontology/activeYearsStartYear	0.7691
http://dbpedia.org/property/yearsActive	0.732
http://xmlns.com/foaf/0.1/givenName	0.7132
http://dbpedia.org/ontology/birthPlace http://dbpedia.org/property/birthPlace	0.6269
http://dbpedia.org/ontology/birthName http://dbpedia.org/property/birthName	0.4192
http://dbpedia.org/ontology/associatedBand http://dbpedia.org/ontology/associatedMusicalArtist	0.3485
http://dbpedia.org/ontology/hometown http://dbpedia.org/property/hometown	0.3214
http://dbpedia.org/ontology/deathDate http://dbpedia.org/property/deathDate	0.3028
http://dbpedia.org/ontology/alias http://dbpedia.org/property/alias	0.301
http://dbpedia.org/property/placeOfBirth	0.2822
http://dbpedia.org/ontology/origin http://dbpedia.org/property/origin	0.2564
http://dbpedia.org/property/associatedActs	0.2353
http://dbpedia.org/ontology/genre http://dbpedia.org/property/genre	0.2327
http://dbpedia.org/ontology/activeYearsEndYear	0.2248
http://dbpedia.org/ontology/recordLabel http://dbpedia.org/property/label	0.2206
http://dbpedia.org/property/shortDescription http://purl.org/dc/elements/1.1/description	0.1835
http://dbpedia.org/ontology/deathPlace http://dbpedia.org/property/deathPlace	0.1819
http://dbpedia.org/ontology/birthYear	0.1734

Tabuľka D.7: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Film

Predikát	Skóre
http://dbpedia.org/ontology/runtime	1.0000
http://dbpedia.org/ontology/Work/runtime	1.0000
http://dbpedia.org/property/runtime	0.9997
http://dbpedia.org/ontology/starring http://dbpedia.org/property/starring	0.6816
http://dbpedia.org/ontology/gross	0.6582
http://dbpedia.org/property/gross	0.6582
http://dbpedia.org/ontology/budget http://dbpedia.org/property/budget	0.5786
http://dbpedia.org/ontology/writer http://dbpedia.org/property/writer	0.3796
http://dbpedia.org/ontology/producer http://dbpedia.org/property/producer	0.3009
http://dbpedia.org/ontology/imdbId	0.279
http://dbpedia.org/property/released	0.2539
http://dbpedia.org/property/id	0.2452
http://dbpedia.org/ontology/distributor http://dbpedia.org/property/distributor	0.194
http://dbpedia.org/ontology/releaseDate	0.158
http://dbpedia.org/ontology/director http://dbpedia.org/property/director	0.1553
http://dbpedia.org/property/screenplay	0.1236
http://dbpedia.org/ontology/musicComposer http://dbpedia.org/property/music	0.1033
http://dbpedia.org/ontology/Engine/length http://dbpedia.org/ontology/Infrastructure/length http://dbpedia.org/ontology/MeanOfTransportation/length http://dbpedia.org/ontology/Weapon/length http://dbpedia.org/ontology/length http://dbpedia.org/property/length	0.0861
http://dbpedia.org/ontology/language http://dbpedia.org/property/language	0.0856
http://dbpedia.org/ontology/editing http://dbpedia.org/property/editing	0.0707

Tabuľka D.8: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Country

Predikát	Skóre
http://dbpedia.org/property/conventionalLongName	1.0000
http://dbpedia.org/ontology/country http://dbpedia.org/property/country	1.0000
http://dbpedia.org/ontology/dissolutionYear	0.8837
http://dbpedia.org/property/yearEnd	0.8811
http://dbpedia.org/ontology/foundingYear	0.8709
http://dbpedia.org/property/yearStart	0.868
http://dbpedia.org/property/nativeName	0.6563
http://dbpedia.org/ontology/birthPlace http://dbpedia.org/property/birthPlace	0.5696
http://dbpedia.org/ontology/foundingDate	0.3981
http://www.georss.org/georss/point	0.3722
http://www.w3.org/2003/01/geo/wgs84_pos#geometry	0.3722
http://www.w3.org/2003/01/geo/wgs84_pos#long	0.3722
http://www.w3.org/2003/01/geo/wgs84_pos#lat	0.3722
http://dbpedia.org/property/placeOfBirth	0.3348
http://dbpedia.org/ontology/dissolutionDate	0.3273
http://dbpedia.org/property/nationalAnthem	0.1638
http://dbpedia.org/ontology/capital http://dbpedia.org/property/capital	0.152
http://dbpedia.org/ontology/leaderTitle	0.1468
http://dbpedia.org/ontology/location http://dbpedia.org/property/location	0.1455
http://dbpedia.org/ontology/nationality http://dbpedia.org/property/nationality	0.1449

Tabuľka D.9: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Company

Predikát	Skóre
http://dbpedia.org/ontology/foundingYear	1.0000
http://dbpedia.org/ontology/recordLabel http://dbpedia.org/property/label	0.8116
http://dbpedia.org/ontology/numberOfEmployees http://dbpedia.org/property/numEmployees	0.5023
http://dbpedia.org/property/website	0.5011
http://dbpedia.org/property/foundation	0.357
http://dbpedia.org/ontology/revenue http://dbpedia.org/property/revenue	0.3422
http://dbpedia.org/ontology/industry http://dbpedia.org/property/industry	0.2239
http://dbpedia.org/ontology/netIncome http://dbpedia.org/property/netIncome	0.2226
http://dbpedia.org/property/products	0.217
http://dbpedia.org/ontology/product http://dbpedia.org/property/product	0.19
http://dbpedia.org/property/keyPeople	0.1873
http://dbpedia.org/ontology/distributor http://dbpedia.org/property/distributor	0.1688
http://dbpedia.org/ontology/location http://dbpedia.org/property/location	0.1655
http://dbpedia.org/ontology/operatingIncome http://dbpedia.org/property/operatingIncome	0.1639
http://dbpedia.org/ontology/type http://dbpedia.org/property/type	0.1628
http://dbpedia.org/ontology/assets http://dbpedia.org/property/assets	0.1467
http://dbpedia.org/property/founded	0.1441
http://dbpedia.org/ontology/extinctionYear	0.1394
http://dbpedia.org/property/defunct	0.1262
http://dbpedia.org/ontology/fate http://dbpedia.org/property/fate	0.125

Tabuľka D.10: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Sport

Predikát	Skóre
http://dbpedia.org/ontology/sport http://dbpedia.org/property/sport	1.0000
http://dbpedia.org/property/nickname http://xmlns.com/foaf/0.1/nick	0.9884
http://dbpedia.org/ontology/teamSize	0.9767
http://dbpedia.org/property/sports	0.5461
http://dbpedia.org/ontology/team http://dbpedia.org/property/team	0.4826
http://dbpedia.org/ontology/equipment http://dbpedia.org/property/equipment	0.3772
http://dbpedia.org/ontology/sportGoverningBody http://dbpedia.org/property/union	0.2093
http://dbpedia.org/ontology/category http://dbpedia.org/property/category	0.1835
http://dbpedia.org/property/ball	0.1512
http://dbpedia.org/ontology/occupation http://dbpedia.org/property/occupation	0.0378
http://dbpedia.org/ontology/event http://dbpedia.org/property/event	0.0154
http://dbpedia.org/ontology/birthDate http://dbpedia.org/property/birthDate	0.0116
http://xmlns.com/foaf/0.1/surname	0.0116
http://xmlns.com/foaf/0.1/givenName	0.0116
http://dbpedia.org/property/dateOfBirth	0.0116
http://dbpedia.org/property/filename	0.0116
http://dbpedia.org/property/description	0.0116
http://dbpedia.org/property/mostChamps	0.0116
http://dbpedia.org/property/founded	0.0116
http://dbpedia.org/property/teams	0.0116

Tabuľka D.11: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Work

Predikát	Skóre
http://dbpedia.org/ontology/runtime	1.0000
http://dbpedia.org/ontology/Work/runtime	1.0000
http://dbpedia.org/property/runtime	0.805
http://dbpedia.org/ontology/releaseDate	0.8039
http://dbpedia.org/ontology/genre http://dbpedia.org/property/genre	0.569
http://dbpedia.org/property/firstAired	0.4665
http://dbpedia.org/property/showName	0.4356
http://dbpedia.org/property/numEpisodes	0.433
http://dbpedia.org/ontology/numberOfEpisodes	0.4318
http://dbpedia.org/ontology/producer http://dbpedia.org/property/producer	0.3835
http://dbpedia.org/property/gross	0.3519
http://dbpedia.org/ontology/gross	0.3515
http://dbpedia.org/ontology/numberOfSeasons	0.3508
http://dbpedia.org/ontology/budget http://dbpedia.org/property/budget	0.3307
http://dbpedia.org/property/lastAired	0.3033
http://dbpedia.org/ontology/completionDate	0.3024
http://dbpedia.org/property/numSeasons	0.3022
http://dbpedia.org/ontology/starring http://dbpedia.org/property/starring	0.2959
http://dbpedia.org/ontology/recordLabel http://dbpedia.org/property/label	0.2612
http://dbpedia.org/ontology/writer http://dbpedia.org/property/writer	0.231

Tabuľka D.12: Ukážka prvých 20 záznamov bázy znalosti pre triedu dbo:Eukaryote

Predikát	Skóre
http://dbpedia.org/property/regnum	1.0000
http://dbpedia.org/ontology/kingdom	0.7810
http://dbpedia.org/ontology/genus http://dbpedia.org/property/genus	0.7000
http://dbpedia.org/ontology/synonym	0.547
http://dbpedia.org/property/synonyms	0.5463
http://dbpedia.org/property/classis	0.5177
http://dbpedia.org/ontology/conservationStatus	0.3544
http://dbpedia.org/ontology/status http://dbpedia.org/property/status	0.3529
http://dbpedia.org/property/statusSystem	0.3452
http://dbpedia.org/ontology/conservationStatusSystem	0.3452
http://dbpedia.org/ontology/class http://dbpedia.org/property/class	0.3168
http://dbpedia.org/ontology/division	0.2591
http://dbpedia.org/ontology/binomialAuthority http://dbpedia.org/property/binomialAuthority	0.2365
http://dbpedia.org/ontology/phylum http://dbpedia.org/property/phylum	0.1605
http://dbpedia.org/ontology/order	0.1131
http://dbpedia.org/property/ordo	0.0801
http://dbpedia.org/ontology/species http://dbpedia.org/property/species	0.0525
http://dbpedia.org/property/divisio	0.0296
http://dbpedia.org/property/subgenus	0.0221
http://dbpedia.org/property/unrankedDivisio	0.0195