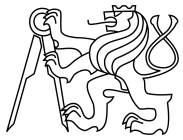
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Grafové algoritmy pro doporučování v Linked Data
Student:	Bc. Martin Chouň
Vedoucí:	Ing. Milan Dojčinovski
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

V posledních letech bylo podle principu "propojených dat", tzv. Linked Data, zveřejněno velké množství volně dostupných datasetů. Cílem diplomové práce je navrhnout a naimplementovat inteligentní webovou aplikaci pro prohlížení a doporučování Linked Data informací na základě traverzování dat. Doporučený obsah bude personalizován podle preferencí uživatelů.
Pokyny:
- Seznamte se s principy Linked Data.
- Proveďte rešerši v oblasti doporučovacích systémů založených na Linked Data principech.
- Analyzujte a identifikujte vhodné grafové algoritmy pro traverzování grafových struktur.
- Na základě analýzy navrhněte architekturu webové aplikace pro doporučování Linked Data.
- Vyberte jednu doménu, pro kterou aplikaci implementujete (např. doporučování filmů, hudby).
- Otestujte nově navrženou webovou aplikaci alespoň na dvou vybraných datasetech (např. DBpedia, DBLP, MovieLens).

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 27. ledna 2016

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

Diplomová práce

Grafové algoritmy pro doporučování v Linked Data

Bc. Martin Chouň

Vedoucí práce: Ing. Milan Dojčinovski

9. května 2016

Poděkování

Děkuji vedoucímu práce Ing. Milanu Dojčinovskému
za neocenitelné rady a pomoc při tvorbě diplomové práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 9. května 2016 .....

České vysoké učení technické v Praze
Fakulta informačních technologií

Odkaz na tuto práci

CHOUŇ, Martin. *Grafové algoritmy pro doporučování v Linked Data*. Praha, 2016. Diplomová práce. České vysoké učení technické v Praze, Fakulta informačních technologií. Vedoucí práce Ing. Milan Dojčinovski.

Abstrakt

Diplomová práce se zabývá grafovými algoritmy pro doporučování aplikovanými na oblast propojených dat. Autor se v práci zaměřuje především na popis technologií sémantického webu a principů propojených dat, dále na doporučovací systémy, jejich funkce a techniky doporučování, a zmiňuje též současná existující řešení z oblasti doporučování v propojených datech. Zaobírá se analýzou grafových algoritmů a představuje řešení své aplikace, kde jich využívá. V neposlední řadě též autor podrobuje aplikaci experimentům na reálných datech, diskutuje získané poznatky a konečně poskytuje čtenáři pohled do budoucnosti ve směru dalšího možného vývoje a rozšíření jak vlastní aplikace, tak této práce.

Klíčová slova Sémantický web, metadata, propojená data, zdroj

Abstract

The thesis deals with graph based recommendation algorithms for Linked Data. The author focuses on the description of the Semantic Web and Linked Data principles, recommender systems, their functions, recommendation techniques, and mentions the current existing solutions in the Linked Data recommendation. He deals with the analysis of graph algorithms and presents design and implementation of the application which uses them. Finally, the author makes experiments with the application on real data, discusses acquired knowledges and gives readers insight into the future development and expansion of this thesis and application.

Keywords Semantic web, matadata, Linked Data, resource

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Stejně tak, jako byl vznik počítačů podmíněn velkým množstvím dat, které nebylo v lidských silách zpracovat, zrodil se web a jeho technologie z potřeby jednoduše data sdílet, zpřístupňovat a prezentovat. Postupným vývojem se rozšířila nejen základna jeho uživatelů, ale také množství zařízení, jež využívají jeho dat a aplikací. Ruku v ruce s tímto trendem jde pochopitelně i webový obsah v současné době zahrnující nezměrné množství rozličných dat. Objem digitálních dat na webu roste nesmírnou rychlostí, a to každým okamžikem, z obecného pohledu nekontrolovatelně a nezadržitelně.

Třebaže je převážná část webových dat určena k prezentaci uživatelům, stále více zařízení, která s nimi pracují, vyžaduje dodatečné informace o jejich významu, ne-li přímo data v konkrétním strojově čitelném formátu, aby je mohla správně zpracovat. A jelikož data přístupná na webu nepodléhají žádným obecně závazným pravidlům pro publikaci, bylo nutné přistoupit k vytvoření doporučení, jak jednotlivé entity dat popisovat. Principy propojených dat, které tyto postupy shrnují, přinášejí též možnost mezi sebou popisované entity provazovat pomocí jejich identifikátorů.

Původní myšlenka přišla z oblasti knihovnictví, kde se pracuje například se seznamy autorů a publikací společně s přidanými metadaty o těchto položkách. Pro evidenci i veřejnou dostupnost takových informací byl tento přístup velmi vhodný. V současnosti ovšem propojená data obsahují četné informace z všemožných oblastí, ať se již jedná o věci úředního charakteru povinně publikované podle zákona, statistická data, nebo třeba informace o uživatelích na sociálních sítích.

Propojená data, tak jak jsou koncipována, už z podstaty reprezentuje graf vzájemně prolinkovaných prvků, jehož průchodem lze snadno objevovat nové informace, rozšiřovat doposud nabyté znalosti a zjišťovat konkrétní význam

dat. Primárně zajištěná strojová čitelnost pak umožňuje jejich snadné a efektivní zpracování. Nástroje, které s propojenými daty pracují, mají za úkol je především analyzovat, extrahovat z nich důležité informace, vyhledávat v jejich strukturách a vizualizovat je koncovým uživatelům.

Tato práce se na jeden takový nástroj, přesněji obecnou techniku, zaměřuje. Předmětem jejího zkoumání jsou doporučovací algoritmy aplikovatelné na grafové struktury propojených dat v oblastech, jakými jsou kupříkladu filmy nebo hudba; vždyť právě tyto oblasti mohou využít potenciálu propojených dat nejlépe. Uživatelé datům rozumějí, a tak informace není třeba před jejich prezentací nijak upravovat, významy vazeb mezi entitami jsou jim ovšem skryty. Aplikace, která je součástí této práce, má uživatelům napomoci ve výběru důležitých entit na základě jejich propojení.

Cíl práce

Úlohou diplomové práce v teoretické rovině je seznámit čtenáře se základními pojmy z oblasti propojených dat a uvést jej jednak do problematiky klasického webu a jeho vývojových stádií, jednak i do oblasti sémantického webu. Spolu s principy propojených dat má ozřejmit jejich podobu, způsoby publikace na webu a možnosti zpracování. Taktéž by se měla obecně dotknout doporučovacích systémů, jejich vlastností a charakteristik se zaměřením na různé doporučovací techniky a přístupy aplikovatelné na grafové struktury propojených dat.

V dalším by zde měl být představen návrh a implementace webové aplikace pro inteligentní procházení grafem propojených dat a doporučováním vyhledaných položek, k čemuž využije vhodných algoritmů vybraných z výstupů jejich předchozí analýzy. Její podstatou je postupné procházení výsledků, kdy v každém jednom kroku dovoluje uživateli personalizovat doporučený obsah podle jeho preferencí. Hodnocení uživatele se dále promítne do výsledků následujícího kroku.

Součástí práce nechť jsou také experimenty na existujících datových sadách, v rámci nichž budou sledovány jak výstupy závislé na počátečním nastavení aplikace, tak vliv různých hodnot parametrů na výsledky doporučovacích algoritmů. Ideálním datovým podkladem se jeví výběr dvou různých zavedených datových sad obsahujících metadata z jedné zvolené oblasti, na kterou se doporučování omezí.

Závěrem by též neměla scházet nabídka možných budoucích směrů, jakými se může ubírat rozšíření samotné práce i vytvořené aplikace.

Struktura práce

Obsah práce se skládá ze dvou zastřešujících částí. Do první části, rešeršní, spadá kapitola o sémantickém webu seznamující čtenáře s rozdíly oproti klasickému webu, jeho vývojem a technologiemi, ozřejmuje principy propojených dat a představuje rámec pro sémantický popis dat. Patří sem taktéž kapitola popisující funkce doporučovacích systémů, techniky doporučování a s nimi spojené nesnáze, prozkoumává též existující řešení v oblasti doporučovacích systémů založených na principech propojených dat.

Druhým okruhem je část realizační, kde je nejprve provedena analýza problému a algoritmů k jeho řešení. Další kapitoly jsou poté zaměřeny na návrh a implementaci aplikace pro procházení propojených dat a doporučování jejich entit. Poslední kapitola této části se soustředí na experimenty s aplikací a jejími algoritmy na reálných datech spolu s diskusí získaných výsledků a nabytých poznatků.

Povaha pramenů informací

Převážná část použité literatury v rešeršní části je povahy vědeckých článků, které se zabývají doporučovacími technikami a praktickými řešeními v oblasti systémů pracujících s propojenými daty. Obecné přístupy, technické a matematické popisy a definice jsou čerpány z renomovaných publikací. Informace o webových standardech vycházejí z doporučení spravovaných příslušnými autoritami, proto jsou do pramenů zahrnuty i veřejně přístupné online zdroje s popisy vydaných norem. Podobně jsou prezentovány popisy a manuály diskutovaných webových aplikací operujících nad propojenými daty.

Kapitola 1

Sémantický web

S pojmem *Web*, přesněji řečeno s termínem *World Wide Web*, se setkává každý uživatel Internetu, neboť se jedná o nejrozsáhlejší systém, síťovou službu, jejíž obsah její konzumenti nejen odebírají, ale též přímo vytvářejí. Od roku 1991, kdy Timothy John Berners-Lee, duchovní otec Webu, poprvé publikoval v rámci Evropské organizace pro jaderný výzkum (CERN) strukturovaný dokument přístupný přes tamější síť, uplynula již dlouhá doba a po zpřístupnění Webu široké veřejnosti se Internet začal prudce zaplňovat množstvím *hypertextových dokumentů*, totiž vzájemně provázaných textových dokumentů za použití odkazů [58]. Postupem času tak vznikla velká decentralizovaná síť obsahující dokumenty ve formátu HTML obohacené o multimediální složku: obrázky, zvuk, video [59].

1.1 Klasický web

Architektura klasického webu je založena na jednotném sdíleném prostoru dokumentů určených především pro uživatele – lidi. Často je označovaná jako *web dokumentů* a je vystavěna na několika základních principech, jimiž jsou používání

- HTML, tedy *Hypertext Markup Language*, značkovacího jazyka a přeneseně i formátu textových dokumentů, v němž jsou na webu publikovány;
- URL, čili *Uniform Resource Locator*, jednotné adresy zdroje, která jednoznačně identifikuje dokument v rámci globálního prostoru;
- HTTP, *Hypertext Transfer Protocol*, internetového protokolu pro určení umístění dokumentů s užitím URL a pro vlastní přístup k těmto;
- a odkazů, tedy hypertextových *linků*, propojujících dané dokumenty mezi sebou [31].

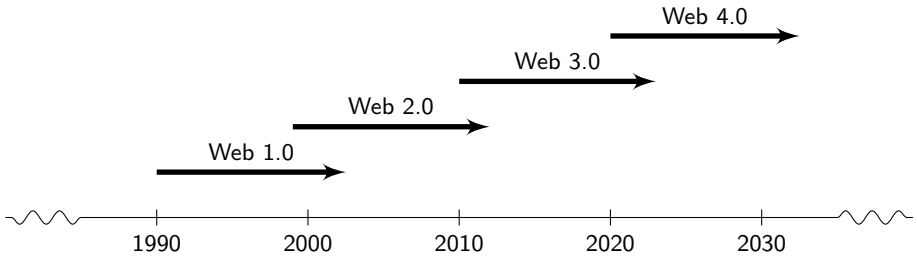
Nad takto organizovaným prostorem dokumentů pracují dva typy aplikací, a sice

- *webové prohlížeče*, které pomocí URL lokalizují dané hypertextové dokumenty a umožňují přecházet mezi provázanými dokumenty pomocí odkazů, a dále

- *webové vyhledávače*, které umožňují indexování a fulltextové vyhledávání hypertextových dokumentů [31].

Původní koncepce Webu, zaměřená na lidsky čitelné informace, neposkytuje vhodný základ pro strojovou čitelnost publikovaných informací. V případě, že chceme zveřejnit na klasickém webu data, ať již lidsky čitelná či nikoli, nebo více či méně vhodně strukturovaná, máme v zásadě dvě možnosti, jak to učinit: buďto poskytnout data v určitých formátech identifikovaná vlastními URL, nebo zvolit pokročilejší metodu, a tedy veřejně vystavit data skrytá za *rozhraními* – API (= *Application Programming Interface*) [31].

Vývoj Webu však jde stále kupředu a rozmach s ním spojených technologií a nových přístupů zapříčinil, že se dnes jeho jednotlivé vývojové etapy rozlišují označením *Web x.0*. Následující časová osa na obrázku 1.1 ilustruje důležité mezníky.



Obrázek 1.1: Vývojová stádia Webu x.0 v čase [69]

První etapou Webu, označovanou pořadovým číslem 1.0, se myslí web od jeho prvopočátků, takový, jak jej stvořil Timothy Berners-Lee, tedy jako jeden z kanálů pro sdílení informací, proto je též někdy nazýván *webem poznání* či *webem informací* [1]. Jednalo se o web statický, který uživatelům umožňoval pouze vyhledávat a zobrazovat informace [6].

S označením *Web 2.0*, jako nové éry Webu, přišli v roce 2004 Tim O'Reilly a Dale Dougherty [1]. Obecně je na tuto éru nahlíženo jako na web, kde je dosavadní prostor statických stránek a dokumentů nahrazován prostorem pro sdílení a společnou tvorbu obsahu, takzvaný *read-write web*, sdružující sku-

piny uživatelů se společnými zájmy za pomoci vzájemných sociálních interakcí, velmi často je možné se setkat s označením *sociální web* [1, 48].

Následný vývojový stupeň, *Web 3.0*, přináší strojovou čitelnost informací pro zařízení, která s ním pracují. Snaží se tak odlehčit uživatelům v komunikaci a interakci s jejich zařízeními, co do častých rozhodnutí, a přenechat část tohoto úkolu danému stroji. A protože přináší obohacení o informace s významem, kterému zařízení rozumějí, nazývá se toto stádium vývoje *webem sémantickým*. [1]

Nejbližším milníkem v evoluci Webu je pak *Web 4.0*, jehož definice dosud nemá jasné obrysy; někdy je s tímto pojmem spojováno označení *mobilní web*, jindy je pojmenováván jako *inteligentní web* či *web s umělou inteligencí* [69], ačkoli poslední dvě přízviska jsou mnohými již přisuzována následnému *Webu 5.0*, který má být *webem emočním* spojujícím veškeré dosavadní přínosy předchozích vývojových verzí Webu [1, 30].

1.2 Počátky sémantického webu

Myšlenku *sémantického webu* poprvé představil širší veřejnosti Timothy John Berners-Lee v roce 2001, když společně s Jamesem Hendlerem a Orou Lassilou publikovali článek, v němž s jistým vizionářským nádechem hledí do budoucnosti Webu [41]. Autoři v něm nastiňují, jak by v budoucnu mohl vypadat jistý časový úsek dne v životě běžného člověka, který bude vlastnit mobilní zařízení a připojovat se s ním k Internetu [41]. Zásadní je však skutečnost, že zde přesně vymezují pojem sémantického webu – jeho definici (přeloženo podle [41]):

„Sémantický web je rozšířením současného webu, v němž informace mají přidělen dobře definovaný význam lépe umožňující počítačům a lidem spolupracovat. Sémantický web představuje reprezentaci dat na WWW. Je založen na technologii Resource Description Framework (RDF), která integruje širokou škálu aplikací využívajících syntaktický zápis v XML a identifikátory URI pro pojmenovávání.“ [8, 71]

Důraz je kladen především na formulaci *„Sémantický web je rozšířením současného webu. . . “*, neboť Web 3.0 sestává z původního Webu 2.0 obohaceného o sémantiku informací. Sémantika, jakožto nauka o významu slov, o vztazích mezi výrazy a tím, co označují a případně i těmi, kdo je užívají [51]. Přispívá onou významovou složkou, kterou obohacuje data Webu 2.0, proto se nejedná o zcela nový web, nýbrž pouze o jeho rozšíření, kde data získávají přesný význam [58]. S označením Web 3.0 se také užívá pojmu *propojená data* (z anglického *Linked Data*) [53].

1.3 Hlavní aspekty

Vytváření sémantického webu spočívá v popisu jevů, vymezování pojmů a kategorií – takzvané *konceptualizaci* dat – dostupných v prostředí Internetu, přičemž se vytváří abstraktní model určité části skutečného světa, z něhož jsou vybírány a určovány jeho významné pojmy [41].

1.3.1 Ontologie

Klíčovým nástrojem pro konceptualizaci dat jsou *ontologie*. Jedná se o formální reprezentaci znalostí určených především k jejich znovupoužití a sdílení. Organizovány jsou hierarchicky, případně jsou provázány do sítí, a velmi často jsou *doménově orientované*, tedy zaměřují se na konkrétní oblast a poznatky z daného oboru [41]. Jejich obsahem jsou definice tříd, konceptů a vzájemných ontologických vazeb mezi nimi, popisující existence, kategorie nebo klasifikační systémy stran aplikační domény [41, 62]. V rámci ontologií se uplatňuje dědičnost mezi třídami, kdy podtřídy představují konkrétní koncepty a vymezují jim užší význam, dále jsou zde určeny jejich vlastnosti (označované jako *role* či *sloty*) a omezení v rámci těchto vlastností [10]. Pro ontologii společně s jednotlivými instancemi jejích tříd se užívá termínu *znalostní báze* [10].

Příkladem jednoduché ontologie může být model souhrnného popisu vín [10]:

Třídy: `Vinné_sklepy` a `Víno`,

podtřídy třídy `Víno`: `Červené`, `Bílé` a `Rosé`,

vlastnosti: `Výrobce` a `Tělo`,

instance třídy `Vinařství`: `Château Lafitte-Rothschild`,

instance třídy `Víno`: `Château Lafitte-Rothschild Pauillac`,

- **vlastnost `Výrobce`:** `Château Lafitte-Rothschild`,
- **vlastnost `Tělo`:** `plné`.

1.3.2 Jazyk pro reprezentaci

Ve světě sémantického webu je nutné zaznamenávat *metadata*, tedy data o datech zachycující obsah, kontext a strukturu dat, jež popisují, a to jazykem, kterému porozumí především výpočetní stroje [41]. Nutný je proto posun od přirozeného jazyka k *jazyku reprezentačnímu*, který dobře poslouží pro popis neomezeného skutečného světa ve světě webu, kde navíc zohlední i jeho specifika; jedním z nejdůležitějších je nutnost entity reálného světa v sémantickém webu jednoznačně pojmenovávat [21].

Na metadata je nutné nahlížet jako na data ve všech jejich aspektech, jelikož mohou být jako jakákoli jiná data vytvářena, měněna a uchovávána v nějakém zdroji, je tedy například možné, aby zdroj obsahoval informace nejen o zdrojích cizích, ale také o sobě samém [7].

Sledujeme tři způsoby existence a cest metadat [11]:

1. Údaje o dokumentu jsou obsažené v něm samotném,
2. údaje o dokumentu doprovázejí komunikaci typu klient-server a je možné je získat po jeho přenosu,
3. údaje o dokumentu je možné získat z jiného dokumentu, jehož jsou součástí.

1.3.3 Zpracování informací

Za účelem automatizovaného zpracovávání informací zakódovaných v určitých reprezentačních jazycích je třeba počítat s přístupem *softwarových agentů* k metadatům. Pojem *agent* v tomto smyslu představuje autonomní inteligentní programovou komponentu, která realizuje určitou úlohu pro jejího vlastníka, příkazce. Mezi takové komponenty můžeme zařadit zejména webové vyhledávače a prohlížeče či inteligentní webové klienty schopné komunikovat i navzájem mezi sebou, schopné informace vyhledávat a sdílet nebo též realizovat transakce. [41, 21]

1.4 Technologie sémantického webu

Jedním z cílů pracovní skupiny W3C Semantic Web Activity je standardizovat klíčové technologie, které umožní decentralizovaný vývoj sémantického webu, ale zároveň zajistí, že všechny části a výsledky tohoto vývoje do sebe zapadnou, proto definuje následující principy, které jsou základními stavebními kameny sémantického webu (převzato a upraveno z [32] a obohaceno o [59]).

1. **Vše je možno jednoznačně ztotožnit s URI.**
   Veškeré objekty skutečného světa kolem nás mohou být jednoznačně identifikovány pomocí URI, tedy nejen klasické webové stránky a dokumenty, ale i konkrétní lidé, události, veškeré věci i abstraktní objekty, díky čemuž se lze na tyto snadno a odkudkoli odkazovat.
2. **Zdroje a odkazy mezi nimi mohou být typované.**
   Lze přidávat informaci nejen k samotným zdrojům, ale i k odkazům mezi jednotlivými zdroji a konkrétně tak popsat a odlišit vztahy mezi nimi i určit druh těchto vztahů.

3. **Neúplné informace jsou tolerované.**
   Jelikož nelze zcela zabezpečit konzistentnost mezi odkazy na zdroje a skutečnou existencí odkazovaných zdrojů, musí být zajištěna tolerance k těmto anomáliím a zaručena zpětná vazba s informací o tomto stavu, přičemž takové skutečnosti nesmějí znemožnit přístup ke zbylým částem informace.

4. **Netřeba záruka pravdivosti.**
   Protože není možné zaručit pravdivost veškerých informací vyskytujících se na webu, je odpovědnost za ověření jejich důvěryhodnosti přesunuta na stranu zpracovávající tyto informace, a je proto zcela na jejím rozhodnutí, jakým způsobem tak učiní a kterým informacím bude důvěřovat.

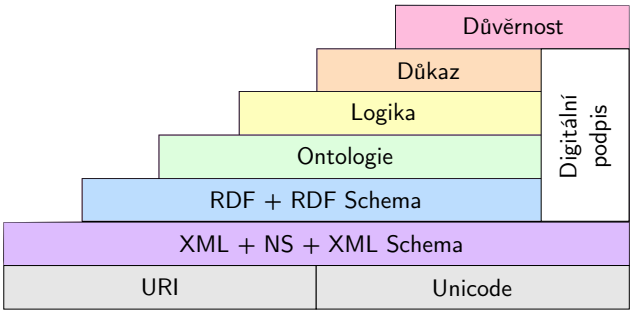
5. **Budoucí vývoj je podporován.**
   Nedá se vyhnout různým popisům a definicím podobných dat na různých místech webu, z tohoto důvodu musí být zachována možnost dané části metadat kombinovat s jinými, znovu používat a obohacovat, navíc musí být možné navázat na předchozí metadata tak, aby nevznikaly nejasnosti při jejich interpretaci, a to bez nutnosti měnit původní data či metadata.

6. **Minimalistický návrh.**
   Jednoduché věci je vhodné činit jednoduchými, složité možnými. Cílem standardizace je normovat pouze tak, jak je nezbytně nutné.

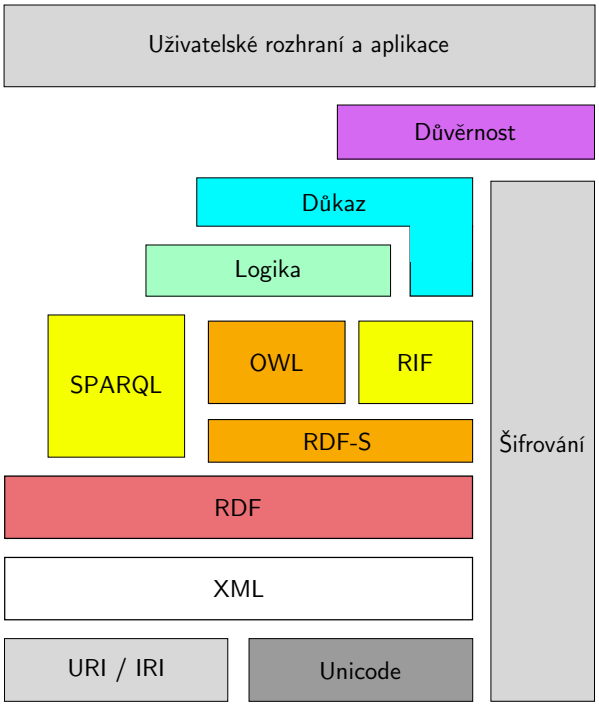
1.4.1 Vrstvený model sémantického webu

Původní model publikovaný skupinou W3C Semantic Web Activity je představen na obrázku 1.2. Obsahuje definice standardů pro obecný přístup k sémantickému webu. V nejnižší vrstvě je zajištěn přístup přes `URI` a určena znaková sada `Unicode` pro kódování dat zapsaných v transkripcích přirozených jazyků zahrnující prakticky veškerá užívaná písma [59]. Vrstva `XML + NS + XML Schema` poskytuje standard jazyka pro konkrétní zápis strukturovaných dat v různých formátech a zajišťuje vysokou interoperabilitu a stupeň integrace [4]. Vrstva implementující model metadat, `RDF + RDF Schema`, poskytuje nástroje pro jejich návrh a zápis [25]. Vrstva `Ontologie` pak zajišťuje především jazykovou podporu pro vytváření vyšších stupňů metainformací: slovníků a ontologií; přidává další funkcionalitu. Vrstva `Logika` určuje univerzální jazyk pro monotónní logiky [4], avšak společně s vrstvami `Důkaz` a `Důvěrnost` dosud nejsou standardizovány a přenechávají řešení na koncové uživatele [59]. `Digitální podpis` pak zabezpečuje ověření integrity dokumentu a důvěryhodnosti zdroje, jakož i autentičnosti jejího autora [59].

Postupným vývojem přibyly do vrstveného modelu sémantického webu další vrstvy či se transformovaly nebo vydělily z vrstev původních. Novější verzi vrstveného modelu demonstruje obrázek 1.3.



Obrázek 1.2: Původní vrstvený model sémantického webu [32, 25]



Obrázek 1.3: Novější vrstvený model sémantického webu [53]

Ve spodních vrstvách se objevuje nový standard z roku 2005, IRI (Internationalized Resource Identifier), který je rozšířením schématu dosavadních URI o znaky sady Universal Character Set (Unicode/ISO10646) [52]. Vyšší vrstvy jsou obohaceny o dotazovací jazyky, v tomto případě označeno jako `SPARQL`, a standard RIF (Rule Interchange Format) pro výměnu pravidel především mezi nástroji pracujícími s pravidly v rámci webů. Celý soubor vrstev pak zastřešuje vrstva pro přístup externích aplikací a napojení na uživatelská rozhraní.

1.5 Propojená data

Termínu *propojená data*, který zavedl Timothy Berners-Lee, se užívá pro pojmenování souboru osvědčených postupů při tvorbě, publikování a vzájemném propojování dat na webu [9]. Následující principy, známé též jako *principy propojených dat*, jsou úzce spjaty s technologiemi sémantického webu, respektive přímo z nich vycházejí (převzato z [5]):

1. Pro pojmenovávání věcí se používají URI.
2. Používají se HTTP URI, aby bylo možné odkazované entity vyhledat.
3. Vyhledává-li kdosi pomocí URI, je vhodné poskytnou mu informace v rámci definovaných standardů (RDF*, SPARQL).
4. Zahrnutím URI odkazů na jiné entity do vlastních dat umožníte objevování dalších věcí.

Myšlenka propojených dat tímto plně koresponduje s architekturou World Wide Webu, která je aplikována na sdílení dat v globální prostoru. Je proto nutné zabezpečit unikátnost daných URI pro jednoznačnou identifikaci, použít HTTP jako mechanismus pro univerzální přístup k dokumentům a umožnit provázání dat odkazy.

1.5.1 Principy

1.5.1.1 Pojmenovávání věcí URI

Konkrétní objekt zájmu, který je pojmenován pomocí URI, nese označení *zdroj* (anglicky *resource*), přeneseně jde o zdroj metadat, které jej popisují. HTTP URI pak poskytují snadný způsob, jak vytvářet jedinečné identifikátory v globálním webovém prostoru, a to decentralizovaným způsobem, ba co více, každý oprávněný uživatel doménového jména může vytvářet své vlastní nové identifikátory. [9]

1.5.1.2 Vytváření dereferencovatelných URI

Veškerá HTTP URI by mělo být možné *dereferencovat*, což znamená, že přistupuje-li klient přes HTTP ke zdroji identifikovanému pomocí URI, má možnost získat jeho popis. Právě z důvodu obecnosti URI nemusí identifikátor odkazovat na webový dokument a data, která se za ním skrývají, mohou být různých druhů a formátů. Typ získané datové reprezentace zdroje závisí na klientu, jenž si data žádá. Pomocí HTTP získá kýženou reprezentaci takzvanou *dohodou o obsahu* (anglicky *content negotiation*), a to nastavením hlavičky `Accept` v odesílaném požadavku. [9]

V praxi je možné se setkat se dvěma strategiemi, jak tvořit dereferencovatelné identifikátory [9].

303 URI
Pokud klient odešle na server požadavek (ideálně metodou HEAD), namísto odpovědi s daty (a stavovým kódem `200` s informací o úspěchu dané operace) je mu zaslán zpět kód `303 See other` s informací o možnosti přesměrování na konkrétní datovou reprezentaci. Seznam všech dostupných reprezentací ve formě posloupnosti URI odkazů se vrátí v rámci odpovědi v hlavičce `Location`; k vybrané reprezentaci je možné následně přistoupit HTTP metodou GET.

Hash URI
Předchozí přístup vyžaduje odeslání dvou HTTP požadavků a přijetí dvou na ně navazujících odpovědí. Pakliže se chceme vyhnout této potenciálně nadbytečné komunikaci, můžeme zvolit způsob dereferencování URI odkazu za pomoci takzvaného *fragmentu adresy* (anglicky *fragment identifier*), což je nepovinný řetězec připojovaný na konec URI odkazu pomocí symbolu `#` (mřížka, hash). Protokol HTTP před samotným požadavkem vyžaduje, aby byl fragment adresy odstraněn, takové URI tedy není možné získat přímo, a proto může být využito k obecné identifikaci.

1.5.2 Životní cyklus propojených dat

Propojená data je třeba udržovat již od jejich vzniku. Tento proces sestává z několika dílčích fází (obrázek 1.4), které ovšem není vhodné řešit odděleně; jsou to (převzato z [3]):

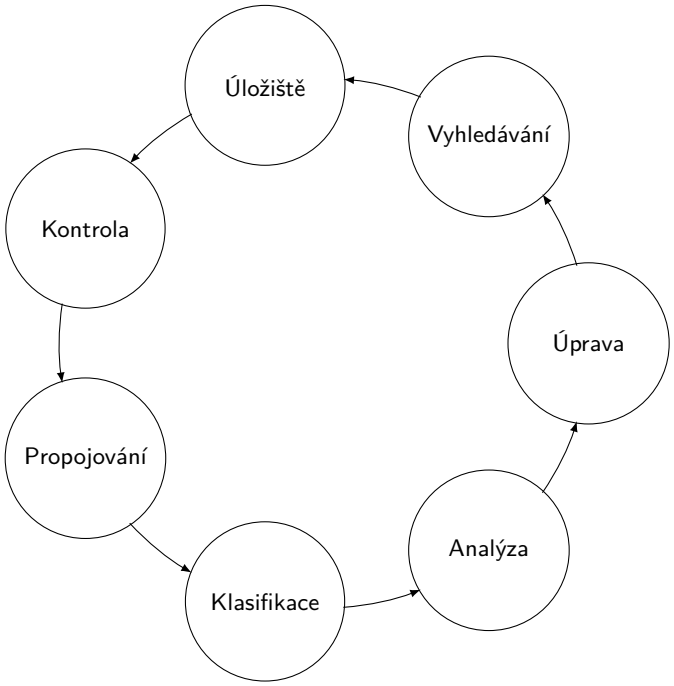
Úložiště a jeho správa je náročnější než je tomu u relačních dat, proto je třeba dbát na užití vhodných technologií pro ukládání dat, zpracovávání grafových struktur a optimalizaci dotazů.

Kontrola znalostních bází založená na paradigmatu editace strukturovaných dokumentů WYSIWYM (z anglického What You See is What You Mean), tedy volně přeloženo: *to, co vidíte je to, co máte na mysli.*

Propojování dat, vytváření a udržování vazeb mezi daty více či méně automatizovaným způsobem za účelem zajištění jejich souvislosti a možnosti další integrace.

Klasifikace surových dat a jejich instancí pro možnost budoucího propojování, kombinování, vyhledávání a další integraci v rámci ontologií vyšších úrovní.

Analýza kvality obsahu dat na webu na základě charakteristik, jakými jsou například jejich původ, kontext, pokrytí nebo struktura.



Obrázek 1.4: Životní cyklus propojených dat [3]

Úprava a vývoj dat, znalostních bází, slovníků a ontologií při zachování jejich celkové stability, a to z důvodu dynamičnosti dat na webu.

Vyhledávání a průzkum dat spojený s jejich zviditelněním běžným uživatelům konkrétními vizualizačními technikami.

1.6 Otevřená propojená data

Způsob, jakým je možné data na webu publikovat, zahrnuje jak formát a přístupnost, tak i úroveň provázanosti s jinými daty. Splňují-li data jisté požadavky, lze je nazývat *otevřenými propojenými daty* (LOD, z anglického *Linked Open Data*). Podmínky, založené na principech propojených dat, kladené na samotná data při jejich „otevírání“ je možné charakterizovat úrovněmi naplnění jejich premis, a data ohodnotit stupni otevřenosti, jež se označují hvězdičkami [31] (hodnocení převzato z [5] a doplněno o [31]).

☆ **Data jsou zpřístupněna pod otevřenou licencí.**
Snadné zpřístupnění dat; uživatelé je mohou procházet, zobrazovat, ukládat, rozmnožovat a upravovat.

☆☆ **Data jsou dostupná ve strukturované podobě.**
Stále snadno publikovatelná; uživatelé je mohou zpracovávat pomocí specializovaných softwarových nástrojů.

☆☆☆ **Data jsou vystavena v neproprietárním formátu.**
Snadná manipulace s daty, bez omezení ze strany softwaru, nepříliš snadný převod do proprietárního formátu.

☆☆☆☆ **K identifikaci entit jsou použita URI.**
Možnost provázání dat, nutnost porozumění struktuře dat, možnost kombinovat různé části dat.

☆☆☆☆☆ **Data jsou propojena s dalšími daty pomocí odkazů.**
Možnost objevovat další informace, nutnost vypořádat se s nefunkčními odkazy, data jsou navíc snadno přístupná.

1.7 Datové modely webu

Datovým modelem je myšlen soubor pravidel a způsobů popisu datových struktur a operací nad kolekcí dat (zvanou databáze) určité domény [47] s následujícími vlastnostmi [23]:

1. Popisují určitou vlastnost reálného světa či specifické znaky oblasti působnosti dané aplikace;

2. měly by být logicky souvislé, měly by nést společné znaky (domény) a měly by mít společný účel;

3. zaměřují se na zvolenou skupinu uživatelů a obvykle vycházejí vstříc již vytvořeným aplikacím.

Dva poslední body určují jasný rozdíl mezi daty, která je možné nalézt v klasických aplikacích pro správu dat a daty na webu [23]. Na klasickém webu dokumentů se setkáme v první řadě s HTML reprezentací dat, jež je určena zejména pro vizualizaci dokumentů [23]. Nejjednodušším způsobem pro publikaci dat, strukturovaných alespoň pro snadné čtení, se tedy nabízelo formátování do tabulek [23, 56].

1.7.1 Běžně užívané modely

1.7.1.1 Tabulková data

S tabulkovými daty, ať již v HTML tabulce či v přiloženém souboru ve formátu daného tabulkového procesoru, je možné lehce manipulovat, z hlediska významu, který datům přísluší, je však možné se opřít pouze o názvy sloupců [56]. Zpracování takových dat je o to snazší, jsou-li v samostatném souboru, v textovém formátu, s oddělovači sloupců.

1.7.1.2 Serializovaný objekt

Objekt převedený do určitého druhu serializace přináší o něco konkrétnější význam pro data, jichž je nositelem. Pro přístup k hodnotám se používají vlastnosti (properties) objektu, které svými jmény mohou naznačovat druh informace, datový typ či formátování držených hodnot. Identifikována jsou konkrétním názvem objektu či názvem souboru, do něhož je serializace ukládána. Zde je již možné s daty manipulovat za pomoci dotazovacích konstruktů [70].

1.7.1.3 Relační model dat

Běžné systémy pro řízení báze dat na relační úrovni umožňují ukládání dat do tříd entit (tabulek), identifikovat je pomocí primárních klíčů, odkazovat se do jiných tabulek cizími klíči a definovat integritní omezení. Především doménová integritní omezení zaručují, že se v daných sloupcích budou vyskytovat konzistentní data. Přístup do databází nebývá veřejný, pohled na určité výseky dat zprostředkovává webová prezentace s privátní přístupovou logikou. Model však uživateli stále neposkytuje konkrétní informace o významu uložených dat, na rozdíl od předchozích ale obsahuje silný nástroj pro dotazování [70].

1.7.1.4 Hierarchická data

Data uložená v XML formátu poskytují informace o obsahu v rámci značek (tagů) a atributů s danými hodnotami. Přidanou hodnotu zde představují metadata pro definici struktury dokumentu: XSD (XML Schema Definition) a DTD (Document Type Definition). Stále však scházejí konkrétní informace o významu dat a jejich propojení [70].

1.7.1.5 Grafový model dat

Grafový model konečně poskytuje vhodné řešení pro modelování dat sémantického webu. Ta jsou reprezentována v textových serializacích. Pro jednoznačnou identifikaci dat jsou užita URI, pomocí nichž je možné je provazovat s dalšími daty, význam jim dodávají ontologie či slovníky, navíc je možné nad nimi realizovat dotazy podobně jako v relačních databázích [70].

1.7.2 Pokusy o formalizaci modelů

S růstem velikosti webu dokumentů se vynořily i první pokusy modelovat jej jako obrovský datový systém a formalizovat jej jako celek. Následující dva modely patří mezi ty nejzajímavější [70].

1.7.2.1 Abiteboulův–Vianuův model

V roce 2000 představili Serge Abiteboul a Victor Vianu model webu, který je sofistikovanější nežli prostý grafový model. Vychází z předpokladů, že informace v něm obsažené jsou volně strukturované, globálního charakteru, a nahlíží na web jako na nekonečný prostor částečně strukturovaných objektů nad relačním schematem definovaným jako

$$\{Obj(oid); Ref(source, label, destination); Val(oid, value)\}, \tag{1.1}$$

kde *Obj* je specifikovaný objekt, *oid* jeho identifikátor (URI), *Ref* určuje konečnou množinu pojmenovaných hran a Val je hodnotou objektu. Již nyní je patrné, že se jedná o model pracující se schématem obsahujícím trojice prvků. V tomto případě si za objekty můžeme představit webové stránky, za jejich hodnotami obsah jednotlivých stránek a jako hrany lze uvažovat odkazy mezi nimi.

Definováno je též dotazování nad modelem vycházející z formalizace zobecněné teorie vyčíslitelnosti. Abiteboul s Vianuem zavádějí pojem *webový stroj*, ekvivalent Turingova stroje s nekonečnými vstupy a výstupy, na němž je pojem dotazu formulován.

1.7.2.2 Mendelzonův–Miloův model

Podobný model vycházející z předpokladu, že se web nechová jako databáze (a sice z důvodu nedostatečné možnosti kontroly souběžného přístupu a omezeným možnostem přístupu k datům vůbec) je i Mendelzonův–Miloův. V tomto modelu je web sice nekonečným prostorem, avšak konečným v každém jednom okamžiku, nekonečnost prostoru totiž stírá rozdíl mezi nesnadno řešitelnými úlohami a úlohami nemožnými. Zatímco tedy v Abitebouově–Vianuově modelu je zjištění seznamu všech stránek dostupných z aktuální stránky problémem nevypočitatelným, v Mendelzonově–Miloově modelu je „pouze“ nesnadno řešitelným.

Model je vystavěn na Turingově stroji s orákulem, které má za úkol simulovat přístup z množiny odkazů (URI) do grafu webu. S jeho pomocí byl vytvořen dotazovací jazyk WebSQL umožňující vyhledávat data pomocí jejich obsahu, struktury a topologie. Na druhé straně jsou ale omezení modelu, která spočívají v práci se statickým obsahem (aktualizace se zcela ignorují), neřeší ani

různorodost dat, nedostatečnou strukturovanost ani úroveň nezávislosti mezi uživateli.

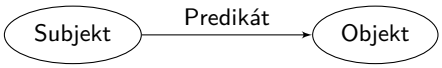
1.8 Resource Description Framework

Hlavním pilířem a technologickým základem sémantického webu je *rámec pro popis zdrojů* (RDF, z anglického Resource Description Framework) [11], případně dalším českým názvem *obecný rámec pro popis, výměnu a znovupoužití metadat* [41]. Opět se jedná o doporučení z dílny W3C konsorcia, které uvádí zúženou definici RDF, tedy, že jde o *jazyk pro reprezentaci informací o zdrojích* [35], a definici aktualizovanou, že jde o *rámec pro vyjádření popisu zdrojů* [57]. Původní verze 1.0 byla představena 10. února 2004, verze novější 1.1 pak 25. února 2014 [31], zásadní změny oproti verzi 1.0 jsou především v používání IRI jakožto jednoznačného identifikátoru zdroje, navíc přibývají nové formáty textové serializace a datové typy [68]. Dále se v práci bude operovat s verzí RDF 1.1.

1.8.1 Schéma

Jádrem frameworku je strukturovaný zápis informací tvořící obecný orientovaný hranově ohodnocený multigraf, jehož základní stavební jednotkou je uspořádaná *trojice* (S; P; O), kde S je *subject*, P *predicate* a O *object* [31, 53]. Velmi zřídka se můžeme sektat i s českým označením *podmět – vztah – předmět* nebo *zdroj – vlastnost – hodnota vlastnosti* [41], častěji však s přejatým označením *subjekt – predikát – objekt* [31, 53].

Množina trojic pak tvoří RDF graf, jehož uzly jsou tvořeny subjekty nebo objekty a orientované hrany od subjektu směrem k objektu predikáty.



Obrázek 1.5: Schéma RDF grafu se dvěma uzly [16]

1.8.1.1 Trojice

V oficiální terminologii se trojice nazývají *tvrzení*, v rámci daného tvrzení je zdroj subjektem, vlastnost predikátem a hodnota vlastnosti objektem [41]. Je tedy možné subjekt ztotožňovat s podmětem věty, predikát s přísudkem a objekt s předmětem, význam každé trojice je pak snadno interpretovatelný v přirozeném jazyce s významem, který tímto vytváří.

Subjekt je IRI, neboli zdroj, případně prázdný uzel.

Predikát je IRI a vytváří binární relaci mezi subjektem a objektem.

Objekt může být IRI, literál nebo prázdný uzel.

Podle definice [16] musejí být IRI absolutní a mohou obsahovat fragment. Relativní IRI jsou vyhodnocována proti předem určenému základnímu IRI, aby s nimi bylo možné pracovat jako s absolutními. Jejich formát se odvíjí od doporučení předepsaných v RFC3987.

Literály jsou takové objekty, které vystupují ve formě hodnot – textových řetězců, čísel anebo kalendářních dat. Sestávají ze třech částí [16]:

- lexikálního vyjádření hodnoty textovým řetězcem znaků v Unicode kódování,
- datového typu ve formátu IRI a
- v případě, že je daná hodnota datového typu textový řetězec, mělo by být definováno neprázdnou *jazykovou značkou* (*language tag*), v jakém jazyce je text zapsán.

Samostatnými vrcholy RDF grafu jsou i *prázdné uzly* (z anglického *blank nodes*), které svým charakterem neodpovídají ani IRI ani literálům, neboť se jedná o uzly s lokální působností s proměnnými identifikátory a syntaxí jejich zápisu plně závislé na konkrétní implementaci serializace [16].

1.8.1.2 Jazyky zápisu, serializace

Původní model RDF vycházel ze zápisu trojic v XML, která již dostatečně naplňovala potřebu přiřadit danému zdroji určité vlastnosti a umožňovala i popsání vztahů mezi zdroji navzájem [41]. Hierarchická stromová struktura (uspořádaný značený neohodnocený kořenový strom) přispívá především k efektivnímu zpracování softwarovými nástroji. Přestože se jedná o součást doporučení W3C konsorcia, zásadní nevýhodou zápisu v XML je horší čitelnost pro člověka [29]. Následující stručný přehled vyvinutých a běžně používaných formátů je vyňat z [57].

Zcela prostou a přímočarou serializací RDF grafu je přímý zápis všech trojic, jak jsou definovány. Tento způsob reprezentace využívá formát N-Triples uchovávající trojice v prostém textovém souboru. Jeho nevýhodou je však vysoká náročnost z hlediska objemu ukládaných dat, protože obsahuje redundantní informace – každá trojice je zapisována celá, byť by všechny obsahovaly stejný subjekt. Rozšířená serializace N-Quads umožňuje zápis trojic do pojmenovaných grafů (vznikají tak čtveřice, avšak s další úrovní nadbytečných dat).

Rozšířením N-Triples je formát Turtle, který již svou syntaxí umožňuje redukovat nadbytečná opakující se data. Zápis je přehledný a čitelný i pro běžného uživatele, zobrazuje-li původní soubor v klasickém textovém editoru. Nadstavbou serializace Turtle je pak formát TriG, jehož syntaxe dovoluje zápis pojmenovaných grafů.

Pro RESTové služby a javascriptové zpracování je možné využít formátu JSON-LD, ježto představuje zápis pro okamžitou deserializaci na objekt. A konečně zápis v RDFa, s jehož pomocí je lze zapsat sémantická data přímo do HTML nebo XML dokumentů.

1.8.2 Cíle

Koncepce RDF si klade tři hlavní cíle (převzato z [34]):

- Nezávislost — mělo by být možné definovat svá vlastní schémata a opětovně je používat určeným způsobem.
- Přenositelnost — výměna dat v RDF, jakož i jejich uchovávání, by mělo být přirozeně jednoduché.
- Škálovatelnost — i pro obrovská množství dat by nemělo být náročné manipulovat s nimi a zpracovávat je.

Kapitola 2

Doporučovací systémy

Množství požadavků kladených na uživatele a přemíra informací, které jsou jim předkládány a z nichž značná část může být zcela nerelevantních, mohou mít negativní dopad na jejich komfort a orientaci v daném prostředí. Příznačné pro tuto situaci je například webové vyhledávání, kdy je uživateli na položený dotaz poskytnuta množina výsledků, které je nucen následně samostatně analyzovat a vybrat takové, jejichž informační hodnota je pro něho významná. Procházení množinou nabídnutých dat, na jejichž způsobu předkládání se nikterak neodrážejí detailnější preference a očekávání těch, kdo si jich vyžádali, může na uživatele neblaze působit enormní zátěží, a tím ovlivňovat jejich trpělivost, potažmo dobu setrvání v systému, stejně tak jako kvalitu získaných informací [54].

Východiskem pro zvýšení efektivity práce se systémem a především pro nejvyšší možné zjednodušení manipulace s daty servírovanými uživatelům se stávají *doporučovací systémy* (RS, z anglického Recommender System) [2]. Zpravidla se jedná o softwarové nástroje a techniky, které poskytují návrhy na doporučení určitých položek z dané oblasti, na níž jsou aplikovány, a to takovým způsobem, aby předkládané údaje byly pro uživatele co možná nejužitečnější, zajímavé a pohotově jim přinášely relevantní informace; pro návrhy využívají rozličných rozhodovacích procesů [55, 60].

Uplatnění doporučovacích systémů v oblasti webových aplikací pro internetové obchody, v oblasti sociálních sítí a reklamy je nabíledni. V drtivé většině případů jsou tyto zaměřeny na získávání zpětné vazby od uživatelů pro (převzato a upraveno z [55])

- zvýšení počtu prodaných položek,
- prodej většího množství různých položek,

- navýšení spokojenosti uživatelů,
- zajištění věrnosti uživatelů,
- hlubší porozumění potřebám uživatelů.

2.1 Funkce doporučovacích systémů

Prvotní funkcí, v níž tkví podstata doporučovacích systémů v prostoru Webu, je nabídnout uživatelům takové dokumenty (obecně tedy položky), které jsou relevantní z hlediska informační potřeby uživatele. Taktéž ale mohou být využity pro přiřazení důležitosti webových stránek ve výsledku vyhledávacího dotazu, případně i pro objevování skrytých závislostí mezi jednotlivými dokumenty vyvstávajících z jejich textových obsahů – používání podobných slov, frekvence jejich užití a další charakteristiky [55]. Doporučování položek se pak rozebíhá mnoha možnými směry (převzato a upraveno z [55, 24]).

Nalezení nějakých vhodných položek nabízí žebříček položek ohodnocených na předem definované stupnici předpovědí, jak zajímavé mohou pro uživatele být.

Nalezení všech vhodných položek doporučí všechny takové, které uspokojí určité potřeby uživatelů.

Kontextová anotace položek pro zaznamenání těsnějších vztahů či v závislosti na dlouhodobých preferencích uživatelů (například pořady v televizním programu).

Doporučení posloupnosti položek, které spolu logicky souvisejí a jako celek zvyšují svou přidanou hodnotu pro uživatele (kupříkladu kniha, na její motivy natočený film a rozhlasová hra).

Doporučení balíku položek, jejichž charakteristiky jsou si blízké či se vzájemně doplňují (příkladem může být námět na výlet spolu s atrakcemi v daném místě, ubytováním, restauracemi a jinými službami).

Prosté procházení umožňuje nabídnout uživateli takové položky, které spadají do aktuálně procházené oblasti, aniž by byl jakkoli omezován v možnostech přecházení mezi nimi.

Volba vhodného doporučovače vychází vstříc uživatelům, kteří mají zájem vyzkoušet nabízené doporučovací nástroje a jejich funkce.

Profilování předkládá uživatelům aktivní možnost, jak předat zpětnou vazbu systému přímo, a sice volbami, které systém nabízí, a docílit tak doporučování šité na míru této konkrétní osobě.

Možnost vyjádření nabízí uživatelům příležitost vyjádřit svůj názor k dané situaci přímo.

Pomoc ostatním skýtá potenciál ve sběru informací od zkušených uživatelů pro usnadnění rozhodování uživatelů nových.

Ovlivňování uživatelů za účelem manipulace s jistými položkami (týká se zejména nákupu zboží).

2.1.1 Získávání znalostí

Pro kvalitní doporučování je nutné získat povědomí o chování všech třech složek, které systém utvářejí. První složkou jsou již dříve zmiňované *položky*, což jsou objekty, jichž se doporučování týká a je možné je charakterizovat složitostí, hodnotou nebo třeba užitečností [55].

Jednotliví *uživatelé* systému pracující s položkami vykazují jisté vzory v chování, pomocí kterých je možné vyhodnocovat jejich zájmy či predikovat budoucí kroky. V oblasti webových doporučovacích systémů se může jednat kupříkladu o způsob procházení webových stránek [63].

Poslední, třetí, složkou jsou *transakce*, čili interakce mezi uživateli a doporučovacím systémem, které uchovávají důležitá data společně s jejich kontextem. Obvykle jsou informace o transakcích uchovávány jako číselné hodnoty, prvky výčtu (ordinální) či binární a unární hodnoty; další metodou je označení položek štítky (tagy), jež suplují konkrétní významové informace. [55]

2.2 Techniky doporučování

Herlocker [24], a potažmo i Ricci [55], rozeznávají následující čtyři přístupy k doporučování.

2.2.1 Kolaborativní doporučování

Též *kolaborativní filtrování*, z anglického *collaborative filtering*, je technikou z oblasti kolektivní inteligence. Spočívá v předpovězení budoucího vkusu skupin uživatelů, vychází přitom z předchozích zkušeností – historie provedených transakcí s položkami. Zaměření čistě na uživatele nevyžaduje žádných znalostí o doporučovaných položkách, vychází pouze z porovnávání chování mezi jednotlivci [28]. Metody pracující se sousedností položek či uživatelů patří mezi nejčastěji nasazované. V dalším lze aplikovat grafové techniky řešící zejména tranzitivitu vztahů a také způsoby redukce rozměrnosti charakteristických příznaků [55]. V praxi se setkáme s tímto typem doporučování téměř ve všech internetových obchodech [28].

2.2.1.1 Aktivní přístup

Přenesením fungování doporučování z reálného světa získáme *aktivní kolaborativní doporučování*, jehož přístup věrně kopíruje způsoby, jakými si lidé mezi sebou běžně předávají informace. Peer-to-peer přístup je vhodný v situacích, kdy je třeba propagovat informaci rychle a efektivně mezi všechny zúčastněné. Nutný je však aktivní postoj uživatele, neboť systém vyžaduje aktivní spolupráci ode všech stejně a nerozdílně [55, 20].

2.2.1.2 Pasivní přístup

Skryté získávání důležitých charakteristik bez požadavku na spolupráci uživatelů je technikou *pasivního kolaborativního doporučování*. Skrytým sběrem dat je myšleno implicitní získávání převážně statistických údajů o uživatelových činnostech v systému; může jít o informace týkající se počtu a druhu prohlížených položek, počtu dotazů na dané položky, čas strávený nad popisem položky a podobně [20].

2.2.2 Doporučování založené na obsahu

Je-li možné analyzovat obsah jednotlivých položek pro konkrétní uživatele bez ohledu, jak jsou organizováni, a s pominutím jejich vzájemných vztahů, lze nasadit metodu *doporučování založeného na obsahu*, anglicky *content-based recommendation* [20]. Systémy tohoto typu pracují na základě získávání informací (techniky *information retrieval*), kdy je porovnáván vlastní obsah položek, a sice pomocí podobností, respektive vzdáleností [28]. Ke spojení s uživatelským profilem dochází přes atributy položek: většinou jde o klíčová slova, v případě sémantického indexování pak o pojmy [55].

2.2.3 Doporučování založené na znalostech

Doporučování založené na znalostech (anglicky *knowledge-based recommendation*) vychází ze specifických sfér znalostí o potřebách a preferencích uživatelů a o míře, s jakou se shodují s charakteristickými vlastnostmi a příznaky položek. Není tudíž vyžadována žádná informace, typu zpětné vazby od uživatele, ve formě hodnocení položek [55]. Za zmínku z této oblasti stojí dva přístupy, které spojuje způsob komunikace s uživatelem, jehož požadavky jsou vstupem do systému jakožto popis problému a výstupy (čili doporučení) jsou řešeními tohoto problému.

2.2.3.1 Doporučování na základě omezení

Zpřesňování výsledků odpovídající uživatelem definovaným omezujícím podmínkám je náplní přístupu *doporučování na základě omezení* (anglicky *constraint-based recommendation*) [55]. Jde o způsob filtrování položek, které splňují definovaná kritéria – vstupy metody jsou [28]:

Požadavky uživatele na parametry hledaných položek;

vlastnosti položky popisující charakteristiky a příznaky položky;

sada omezení a podmínky, které musí každá položka splňovat;

filtrovací kritéria pro výběr položek definují vztahy mezi požadavky uživatele a vlastnostmi položky;

omezení položky z hlediska dostupnosti či oprávnění.

2.2.3.2 Doporučování na základě požadavků

Přestože se prakticky jedná o způsob doporučování založeného na obsahu, je *doporučování na základě požadavků* (anglicky *case-based recommendtaion*) zahrnuto do doporučování vycházejícího ze znalostí. Využívá podobností pro popis, do jaké míry splňují vlastnosti kýžené položky podmínky kladené uživatelem [55]. Avšak zahrnuje navíc pro uživatele možnost určit vlastnosti položky, které mají být minimalizovány (případ zvaný *čím méně, tím lépe*) či maximalizovány (varianta *čím více, tím lépe*) [28].

2.2.4 Smíšené doporučovací techniky

Za účelem zvýšení efektivity systému a odstranění některých nevýhod výše zmíněných doporučovacích technik je možné tyto kombinovat do *smíšených doporučovacích technik* (anglicky *hybrid recommendation techniques*). Hybridní systémy je možné rozdělit do sedmi kategorií podle způsobu, jakým se jednotlivé metody a jejich výsledky kombinují (převzato z [12]).

Vážení neboli číselné kombinování výsledků (hodnocení) jednotlivých součástí systému.

Volba (přepínání) jedné z doporučovacích komponent systému.

Mísení výsledků doporučování a společné zobrazení jejich hodnot.

Kombinace příznaků pocházejících z různých zdrojů znalostí zavedených na vstup jednoho algoritmu.

Obohacování příznaků získaných určitou doporučovací technikou a zavedených (všech či určité podmnožiny) na vstup jiné doporučovací techniky.

Kaskáda doporučovacích komponent systému s přiřazenou prioritou.

Mimoúrovňová kombinace spočívající v postoupení získaného modelu (ten je výsledkem předchozí techniky) další technice, pro kterou je vstupem.

2.2.5 Další techniky doporučování

Množství a jakost dat jsou hlavními činiteli určujícími úspěch dříve předestřených přístupů. Pro správnou práci kolaborativních technik doporučování je třeba mít aktuální data o uživatelích a jejich preferencích. Získat tato data z obrovských databází ovšem nemusí být vůbec snadným úkolem. Totéž v případě doporučování na základě obsahu, kde může být výpočet podobností na nesmírném množství rozmanitých a velmi komplexních dat extrémně náročný, a to především časově [19].

Máme-li však k dispozici organizovaná data v podobě grafu, která svou strukturou mezi sebou jasně definují vzájemné vazby a vztahy, je nasnadě využít těchto vlastností a aplikovat metody, které nejenže nevyžadují rozsáhlé databáze uživatelských hodnocení, ale ani podobnosti mezi konkrétními položkami. Užívá se algoritmů pro průchod grafem dat, přičemž uživatel určuje položkám své preference postupně v jednotlivých krocích průchodu, čímž objevuje další vhodné položky datového modelu, třeba z několika počátečních uzlů zároveň [19]. Vzájemnými vazbami mezi položkami nyní ohodnocenými a nově nalezenými lze těmto vyhledaným entitám určit důležitost, jakou jim pravděpodobně uživatel přikládá.

S úspěchem je možno techniku relevancí grafových dat rozšířit o oba zmiňované přístupy: kolaborativní i datově obsahový. V takovém smíšeném systému se pak získaná důležitost může využít jako váha pro upravení podobnosti či pro zpřesnění klasifikace nebo jako hodnota při řešení problému studeného startu.

2.3 Možné problémy doporučovacích systémů

Většina problémů doporučovacích systémů vyvstává ze způsobu hodnocení zpracovávaných položek. Nejen konkrétní technika doporučování však přináší jistá úskalí do světa automatizovaných, nýbrž i její koncoví uživatelé. Nejčastěji se vyskytují nesnáze spadající do kategorie *problémů studeného startu* (anglicky *cold-start problem*), a to především v čerstvě nasazených systémech, kde se nenacházejí žádné položky ani uživatelé, anebo po jejich doplnění chybí informace o předchozích hodnoceních, tudíž se systém nachází v počátečním stavu, „nemá tak z čeho vycházet“ [28].

2.3.1 Problém nového uživatele

Nově příchozí uživatel do systému nemá ve své historii žádná předchozí hodnocení, začíná tak v bodě „nula“. To je typický případ problému studeného startu, který se vyskytuje především u metody kolaborativního filtrování [33]. Problém lze redukovat nasazením doplňkové podpůrné techniky, čímž

se de facto systém stane hybridním, případně po novém uživateli vyžadovat informace, které mohou napomoci vymanit se z tohoto druhu problému.

2.3.2 Problém nové položky

Obdobou problému nového uživatele (taktéž verze problému studeného startu) je absence potřebných informací o hodnocení u nově přidaných položek. Řešením může být využití doporučování na základě obsahu, neboť zde se dá pracovat s charakteristickými vlastnostmi položek a jejich podobnostmi [42].

2.3.3 Problém přílišného zaujetí

Problém příznačný pro přeučené systémy doporučující na základě obsahu. Do nabídky se neprobojují žádné položky, které jsou odlišné od současných majících vysoké skóre pro doporučení. Ačkoli uživatel může mít snahu o objevování jiných, pro něho nových, a tedy odlišných položek, systém mu žádné takové nepředloží. Východiskem je doporučení celé škály možností, nikoli jen položek s nejlepší shodou [42].

2.3.4 Problém omezené obsahové analýzy

Další z řady problémů technik doporučování založených na obsahu je *problém omezené obsahové analýzy*, kdy se při reprezentaci dvou položek stejnou množinou popisných atributů může přihodit, že tyto není možné od sebe dostatečně rozlišit [33].

2.3.5 Problém řídkosti dat

Reprezentant problému studeného startu v systémech kolaborativního doporučování, kde při velkém objemu dat může docházet ke zkreslování výsledků, ba dokonce i k nemožnosti jejich výpočtu. Úzce to souvisí s matematickými problémy řídkých matic. Pomoci v tomto případě mohou opět hybridní techniky doporučování – například zacílení na uživatele z hlediska demografie a geografické polohy [42].

2.3.6 Nestandardní chování uživatele

Problém přirovnávaný ke stádu ovcí, kdy se jednotlivci ve skupině (možno i mezi skupinami) uživatelů chovají značně odlišně, podle míry odlišnosti buďto jako *šedé ovce* anebo jako *černé ovce*. Hodnocení těchto uživatelů může nabývat různých extrémů, tudíž není snadné jim vyhovět takřka žádným doporučením. Výskyt problému se zcela zřejmě dotýká kolaborativních systémů a dá se řešit pomocí hybridní techniky přepínání (voleb) doporučovacích technik (komponent systému) [13].

2.3.7 Záměrné podvádění a útoky

Ze strany uživatelů může v neposlední řadě docházet také ke vědomé manipulaci s hodnocením položek. V kolaborativním přístupu se lze setkat s takzvaným *útokem pomocí injektáže profilů* – útočník založí v systému množství nepravých uživatelských profilů, díky nimž je pak schopen manipulovat s celkovými výsledky doporučování jen tím, že vhodně ohodnotí položky každou instancí z množiny podvržených profilů [13].

Obdobně i *šilinkový útok* manipuluje s prestiží položek, je ovšem zaměřený na zisk útočníka. Nejenže podvodník vytváří falešné uživatelské profily, ale v zásadě může (dovoluje-li to systém) vkládat i vlastní položky, které se snaží upřednostňovat na úkor ostatních; cizím položkám hodnocení snižuje, svým vlastním navyšuje. Účinnou obranou je detekce nepravých profilů pomocí statistických metod či analýzy shlukování [22].

2.4 Doporučovací systémy založené na principech propojených dat v praxi

Přehled následujících nástrojů založených na principech propojených dat uvádí aplikace orientované na vyhledávání položek, usnadnění práce s výsledky a jejich přívětivou prezentaci uživatelům. V jejich pozadí pracují různé algoritmy pro procházení dat, doporučování či stanovování relevance. Uveden je nástin nejznámějších projektů spadajících do této oblasti.

V rešerši systémů a aplikací pracujících s propojenými daty se objevuje pojem *vyhledávání průzkumem*, případně *průzkumné vyhledávání*, zkratkovitě *objevování*. Termín vychází z anglického originálu *exploratory search*, který White a Roth v [67] definují (přestože připouštějí, že definice není zcela jednoznačná) následovně.

Vyhledávání průzkumem je strategie prohledávání obsahu, která umožňuje objevovat nové asociace a druhy poznání, dle nichž je možné činit přesnější rozhodnutí. Motivace vychází z potřeby řešit komplexní problémy se slabou terminologií a chudým prostorem informací s nutností získávat doplňující informace.

Jako příklad uvádějí osobu, která chce strávit prázdniny v odlehlé vesnici v pronajatém obydlí. Ta za pomoci vyhledávače zjišťuje informace nejen o cenách ubytování, nýbrž i o blízkých městech, kulturním vyžití v okolí apod. Jak postupně nabývá nových znalostí, zpřesňuje tím požadavky ve vyhledávání, tudíž dostává stále relevantnější informace pro své rozhodování, a stále se tak vylepšuje proces doporučování a kvalita výsledků. Kruh se uzavírá.

2.4.1 Discovery Hub

Nejmladším zástupcem ze všech probíraných je webová aplikace Discovery Hub (`http://discoveryhub.co/`) autorů Nicolase Marieho a Damiena Legranda vydaná roku 2013 ve verzi 1.0, v roce 2015 pak povýšila na verzi 2.0 a obdržela přízvisko Beta [38].

Dle popisu na webu projektu [39] a v příspěvku [38] je Discovery Hub prezentován jako nástroj pro vyhledávání průzkumem založený na datech webové encyklopedie Wikipedia a sémantických datech projektu DBpedia. Svým uživatelům nabízí přehledné rozhraní pro vyhledávání informací pomocí klíčových slov, pro zobrazování výsledků dotazů a jejich vzájemných vazeb a umožňuje jim filtrovat je a procházet. Dále také nabízí uživateli možnost zvolit míru důležitosti u určitých oblastí (osoby narozené v daném roce, oblast filmů, herci, zpěváci, ...), o něž má vyšší/nižší zájem, nebo kterým přisuzuje méně či více významu.

Z DBpedie získává Discovery Hub znalosti pomocí SPARQL koncového bodu. Ty následně využívá pro sestavení seznamu výsledných položek, u nichž zobrazuje především významné charakteristické vlastnosti, jako například název, popis a obrázky a názvy témat ve formě tagů. Aby byly uživatelé schopni lépe porozumět prezentovaným výsledkům, obohacuje jednotlivé položky o křížové odkazy na články ve Wikipedii, navíc umožňuje zobrazit přímé i nepřímé vztahy mezi dotazem a položkami výsledku ve formě grafu (uvádí [38] pro verzi 1.0).

Stěžejním algoritmem pro získávání relevantních výsledků je *semantic spreading activation* kombinovaný s fází vzorkování, což umožňuje zpracovávat propojená data za běhu, v danou chvíli, kdy je to třeba. Nasazen je na platformě Corese / KGRAM (Knowledge Graph Abstract Machine) [15], která představuje HTTP server s interpretem SPARQL verze 1.1 pro práci s RDF metadaty v grafové podobě s dalšími rozšířeními (kupříkladu o funkce SQL). Celý back end je napsán a běží v Javě, s uživatelským rozhraním (webová stránka) si vyměňuje data ve formátu JSON [38].

2.4.2 Aemoo

Aemoo, jak je uvedeno na projektových stránkách `http://wit.istc.cnr.it/aemoo/` [72], je jednoduchý vyhledávač pro průzkum webu vyvinutý v roce 2012 skupinou z Laboratoře sémantických technologií STLAB (Semantic Technology Laboratory) při Boloňské univerzitě ve složení: Alberto Musetti a Andrea Nuzzolese (hlavní vývojáři), Francesco Draicchio, Aldo Gangemi a Valentina Presutti.

Vyhledávat lze pomocí klíčových slov v propojených datech. Znalosti se získávají a agregují především ze zdrojů Wikipedia, Twitter a Google News, přičemž z Wikipedie se využívá veškerých existujících odkazů. Vyhledávač ale hlavně stojí na DBpedii, neboť využívá *encyklopedických vzorů znalostí – encyclopedic knowledge patterns* (EKP) [37].

Pokud existuje propojení mezi dvěma entitami (dle vyhledávání), Aemoo zobrazí uživateli popis vyhledávaného prvku a uzly v nejbližším okolí, které se k němu vážou v daném kontextu. Výsledek definuje EKP jako nejdůležitější (nejvýznamnější) věci (entity), které lidé běžně používají pro popis této konkrétní vyhledávané položky.

Aemoo pracuje se selekcí a filtrováním možných výsledků podle získaných poznatků. Data z Twitteru či Google News se zpracovávají pomocí *rozpoznání pojmenovaných entit* (NER; Named Entity Recognition), z oblasti extrakce informací [37].

2.4.3 MORE

Zkratka MORE je prostou složeninou z prvních dvou písmen slov „movie recommendaion“, která zcela stroze vystihuje podstatu tohoto projektu. Jedná se o doporučovací systém založený na datech z DBpedie, který spatřil světlo světa v roce 2010 a jeho autory jsou Roberto Mirizzi, Tommaso Di Noia, Azzurra Ragone, Vito Claudio Ostuni a Eugenio Di Sciascio, italská výzkumná skupina z Politecnico di Bari (Polytechnická univerzita v Bari) [43].

MORE je facebooková aplikace určená pro sémantické doporučování filmů vycházející ze znalostí získaných z propojených dat na DBpedii a LinkedMDB – sémantické verze Internet Movie Database (IMDB) – a z profilu přihlášeného uživatele Facebooku. Úspěšně kombinuje všechny tři techniky doporučování: kolaborativní, založené na obsahu i na znalostech. Pomocí dat z facebookových uživatelských profilů jednoduše odstraňuje problém studeného startu, mnoho uživatelů již totiž vyjádřilo své preference k mnoha filmům.

V případě, že uživatel povolí aplikaci MORE přístup ke svému profilu, může díky ní filmy vyhledávat skrze pole s chytrým napovídáním. K profilům uživatelů jako takovým je přistupováno přes Facebook API, sémantická data se získávají z příslušných SPARQL endpointů. Konkrétními dotazy se shromažďují metadata (charakteristiky) o filmech, žánrech, hercích či členech filmového štábu. Zpětnou vazbu ve formě hodnocení dané položky dodává uživatel systému přes nabídku s hodnotovými posuvníky.

Pro získávání mezivýsledků chytré nápovědy při vyhledávání se užívá algoritmu PageRank, který sbírá data z DBpede a doplňuje je o informace z Wi-

kipedie. Po vyhledání filmu, či jeho výběru z nabídnutých možností našeptávačem, je tento umístěn do uživatelova seznamu oblíbených položek a doporučeno prvních 40 filmů k němu příbuzných. Doporučuje se technikami klasického získávání informací s využitím vektorového modelu. Na každý film je nahlíženo jako na vektor určitých částí s jeho popisem, tudíž jejich podobnost lze počítat pomocí TF-IDF (term frequency – inverse document frequency) s kosinovou metrikou.

Mobilní varianta zvaná Cinemappy vytvořená pro systém Android vychází z principů MORE, přidává navíc možnost detekce polohy. Doporučování tak může vylepšit o informace získané z oblasti výskytu mobilního zařízení, v jeho okolí se totiž mohou nacházet kina, která mají ve veřejně dostupném programu opět filmové tituly vhodné k doporučení [37].

2.4.4 Lookup Explore Discover

Ačkoli zkratka LED spíše evokuje moderní zdroje osvětlení, v tomto případě jde o nástroj Lookup Explore Discover (`http://sisinflab.poliba.it/led/`) pro vyhledávání a objevování znalostí v prostředí webu. Jeho počátky sahají do roku 2010, kdy jej ve svém příspěvku představují již jednou zmiňovaní zástupci Polytechnické univerzity v Bari Roberto Mirizzi, Azzurra Ragone, Tommaso Di Noia a Eugenio Di Sciascio [45].

Webová aplikace slouží pro návrhy vhodných výsledků vyplývajících ze zadaného dotazu, které jsou prezentovány ve formě hesel, tagů [37]. Autoři se v zásadě nebrání označením, že jde o anotační nebo tagovací systém, který obohacuje běžné webové vyhledávání [45].

Lookup Explore Discover je vystavěn nad DBpedií. Na zadaný dotaz jsou uživateli nabídnuty tagy sémanticky podobné (příbuzné), a to nejen zadaným klíčovým slovům, ale i mezi sebou navzájem. Oblak výsledných tagů je prezentován jako seznam odkazů, takže uživatel má možnost procházet jednotlivá hesla a objevovat nové informace. Kromě množiny slov nabízí LED na stránce s výsledky také pohled do rezultátů třech nejužívanějších klasických webových vyhledávačů: Google, Yahoo! a Bing.

Proces používání LED sestává z následujících kroků [45]:

1. Lookup (vyhledání), kdy uživatel zadá do vyhledávacího pole klíčová slova. Během této fáze mu aplikace napovídá podobnými termíny získanými z DBpedie.

2. Exploratory browsing (objevování průchodem), kdy si uživatel vybírá z výsledků předchozí fáze, tak, že prochází nabídnutými hesly, čímž se aktualizuje vyhledávací pole, a tedy i výsledná množina tagů.

Hodnocení uzlů propojených dat je prováděno algoritmem DBpediaRanker. Počítají se podobnosti mezi dvojicemi uzlů – pro každou dvojici zdrojů z grafu DBpedie je za pomoci z dotazů na externí zdroje provedena analýza textu a odkazů. Dotaz na externí zdroje vyhledá počet takových stránek, které obsahují názvy (labels) jednotlivých uzlů zvlášť a poté oba dohromady. V rámci Wikipedie je lze zaměřit se na abstrakty a interní linky [45].

Funkčnost nabízená aplikací přes webové rozhraní je dostupná i přes veřejné RESTful API, touto cestou se dají využít veškeré algoritmy externími aplikacemi. Přenos dat je realizován ve formátu JSON. Webová služba LED nese název Not Only Tag (NOT) [45].

2.4.5 Yovisto

V roce 2010 byl Jörgem Waitelonisem a Haraldem Sackem představen systém Yovisto (`http://www.yovisto.com/index.jsp`) [37, 65], webová aplikace zaměřená na vyhledávání videomateriálů z akademického prostředí – přednášek a konferencí. Výhodou oproti jiným vyhledávačům videí je použití *časově závislého indexu*, díky němuž se dá vyhledávat v obsahu videa. Index je tvořen množstvím s časem spjatých metadat vytvářených automatizovanými způsoby získávání charakteristických vlastností z obsahu videa: rozpoznáváním textů a detekcí scén.

Jednotliví uživatelé také mohou přidávat k videím hodnocení, štítky a komentáře, a to prakticky do kteréhokoli místa jejich obsahu. Data spojená s uživateli se následně využívají v kolaborativním přístupu k doporučování.

Metadata videí Yovista jsou uchovávána ve standardizovaném přenositelném formátu MPEG-7. Pro snadný přístup jsou také všechna publikována v RDF, na webových stránkách pak v RDFa, a dle principů propojených dat namapována do LOD cloudu.

Postup pro vyhledávání a objevování dalších informací procházením:

1. Vyhledají se nejlepší shody mezi zadanými klíčovými slovy a indexovanými vlastnostmi uložených videí,
   - zobrazí se nalezená videa;
2. pro každé klíčové slovo je nalezena jedna či více odpovídajících entit na DBpedii za použití takzvaného *gazetteeru*, indexovaného slovníku, který

obsahuje názvy (termíny, klíčová slova) spjaté s konkrétním URI a počet jejich výskytů,

3. pro každé nalezené mapování entit se provede ověření vůči úložišti Yovista a

4. pro získané výsledky jsou stanoveny jim odpovídající zdroje (odkazy).

Vyhledávání a doporučení příbuzných entit spočívá hodnocení odkazů heuristikami pro skórování jejich vlastností, frekvenčními heuristikami analyzujícími četnosti instancí RDF tříd nebo hledání tříd stejných typů. Veškerá data jsou čerpána ze SPARQL endpointů.

2.4.6 Semantic Wonder Cloud

Dalším z řady nástrojů pro vyhledávání a objevování znalostí na DBpedii je Semantic Wonder Cloud (SWOC; `http://sisinflab.poliba.it/semantic-wonder-cloud/index/`), projekt z roku 2010, který představuje článek z italských univerzit Politecnico di Bari (Polytechnická univerzita v Bari) a Università degli Studi di Trento (Tridentská univerzita), jehož autory jsou Roberto Mirizzi, Azzurra Ragone, Tommaso Di Noia a Eugenio Di Sciascio [44].

Záměr původně vycházející z inspirace aplikací Google Wonder Wheel, který na rozdíl od ostatních nástrojů k prozkoumávání DBpdie umožňuje procházení nejen pomocí přímých (URI) odkazů v RDF metadatech, ale navíc integruje výpočty vztahů mezi jednotlivými uzly DBpedie. Díky těmto přidaným znalostem získaným z webových vyhledávačů a tagovacích systémů je objevování dalších znalostí a jejich sdružování mnohem komplexnější.

Aplikace jako taková je určena pro vyhledávání pomocí klíčových slov s chytrým napovídáním a doplňováním. Výsledky vizualizuje jako graf vztahů mezi nalezenými entitami. Zdroje z DBpedie hodnotí skórem po dvojicích, k čemuž využívá hybridního přístupu: sémantického, vycházejícího ze struktury RDF grafu, a textového, kde užívá extrakce informací z externích zdrojů pro výpočet popularity daných entit.

Uživatel se pohybuje prostorem informací, ve výsledném grafu, přecházením po nabídnutých položkách ze startovního uzlu, který zadá jako klíčové slovo. V aktuálním procházeném uzlu má k dispozici i jeho popis, uzly se vztahem k aktuálnímu jsou vykresleny v různých velikostech podle toho, jaká míra relevance jim přísluší. Každý uzel může zastupovat buď konkrétní instanci entity, nebo celou kategorii seskupující zdroje. Přejde-li uživatel na uzel reprezentující skupinu entit, zobrazí se mu nejpopulárnější instance společně s jejich nejrelevantnějšími kategoriemi.

Na pozadí aplikace pracuje algoritmus DBpediaRanker, jenž realizuje dotazy nad RDF grafem a během jeho průchodu počítá podobnosti mezi každými dvěma objevenými zdroji. Provádí se textová analýza a analýza odkazů v DBpedii, zjišťuje se počet samostatných výskytů popisků na webových stránkách z každého zdroje zvlášť a následně s oběma popisky dohromady. Sledují se abstrakty a odkazy na Wikipedii. Back end tedy sestává z bloků: Graph Explorer (pro SPARQL dotazy nad DBpedií), Context Analyzer (pro omezení domény a kontextu vyhledávání), Ranker (pro výpočet podobností zdrojů) a Storage (pro ukládání předem vypočítaných podobností mezi zdroji a popularit jednotlivých uzlů, aby bylo možné efektivně a opětovně používat tyto informace k dalšímu vyhledávání).

2.4.7 Seevl

Doporučovací systém Seevl (`http://play.seevl.fm/`) zaměřený na oblast hudby a vytěžování znalostí z propojených dat na webu o skladbách, skupinách, žánrech a interpretech pochází z dílny Seevl Ltd při Digital Enterprise Research Institute, National University of Ireland v Galway. Jeho autorem je Alexandre Passant, který jej představil ve svém článku z roku 2011 [49].

Seevl přináší možnost, jak vyhledávat a objevovat oblíbenou hudbu, a to nejen pro uživatele, ale též pomocí dalších aplikací. Jako doplňkovou službu nabízí doporučování a agregaci výsledků. Zdroji dat mu jsou především MusicBrainz, Wikipedia, Freebase, BBC a NY Times, z nichž čerpá RDF metadata. Nejsou-li k dispozici, pak si je do RDF překládá během procesu extrakce, přičemž veškerá mapuje na stejný model Music Ontology. Shromážděná data ukládá v lokální databázi běžící na OpenLink Virtuoso hostované na EC2 škálovatelném klastru.

Jako webová aplikace slouží uživatelům ke správě jejich oblíbených interpretů, vyhledávání a doporučování nových položek, o které by mohli mít v budoucnu zájem. Architektura kompletně založená na technologiích sémantického webu získává RDF data z koncových bodů pro dotazování SPARQL dotazy, k jejichž reprezentaci užívá formátu JSON-LD. Spíše než na trojice se zaměřuje na konkrétní entity v *entity-centric modelu*. Nabízí ovšem také zásuvné moduly pro integraci do plaforem Deezer a YouTube. Dále jsou deklarovány vlastnosti jako

- *škálovatelnost* – využití sémantických dat v celém rozsahu,
- *přísné vyhodnocování výsledků* – doporučování algoritmem dbrec,
- *široká uživatelská základna a komerční využití* – napojením na YouTube,

- *velká škála využití multimediálních dokumentů* – objevování nových informací.

2.4.8 Linked Jazz

Komunitou spjatou s jazzem a jeho historií se zabývá projekt Linked Jazz (`https://linkedjazz.org/`) představený v roce 2011 a následně specifikovaný roku 2013 výzkumnou skupinou z Pratt Institute, School of Information and Library Science ze Spojených států amerických, jejímiž prvotními členy byli M. Cristina Pattuelli, Matt Miller, Leanora Lange, Sean Fitzell a Carolyn Li-Madeo [50].

Výukový projekt je zaměřen na vyhledávání a objevování informací týkajících se oblasti jazzu, k čemuž využívá otevřených propojených dat digitálních archivů, knihoven a muzeí. Sleduje zejména profesionální a sociální vztahy mezi jazzovými hudebníky.

Mapují se jména jednotlivých umělců na přidružená URI, mezi kterými jsou vytvářeny vazby relacemi *osoba zná osobu*. Tato datová sada obsahuje již více než 9 000 jmen umělců. Vytváří se tak sociální síť, která zobrazena ve vizualizačním nástroji ukazuje množství vztahů mezi interprety, a tím nabízí možnost objevování dalších umělců.

Výsledky vyhledávání jsou roztříděny do jedné ze šesti kategorií podle míry shody s dotazem (perfektní, vysoká, střední, nízká, velmi nízká, žádná), uživatel má ale stále možnost je pročišťovat ručně. Tuto funkcionalitu zprostředkovává nástroj pro správu výsledků. Získat propojení do Linked Jazzu pomocí jmen, jakožto znakových řetězců, je možné s použitím vnitřního nástroje Transcription Analyzer, který je schopen identifikovat je v prostém textu (pomocí metod zpracování přirozeného jazyka) – obvykle se v tomto směru užívá přepisů rozhovorů s umělci, kteří se ve svých výrocích odkazují na jiné jim známé umělce, tudíž mezi nimi lze spatřovat jisté vazby.

Na stávající projekt bylo dále napojeno rozšíření Linked Jazz 52nd Street, crowdsourcingový nástroj určený pro spolupráci uživatelů na obohacování databáze Linked Jazz. Za pomoci lidských sil je možno doplňovat dosud neznámá fakta o vztazích mezi umělci, a přispívat tak k propracovanější síti znalostí. Linked Jazz poskytuje pro přístup k datům, do takzvaného Linked Jazz Directory, kromě webového uživatelského rozhraní také veřejné rozhraní RESTové.

2.4.9 inWalk

Interaktivní webová aplikace inWalk (`http://islab.di.unimi.it/inwalk/`) představená v Itálii roku 2014 autorským týmem Silvana Castano, Alfio Fe-

rrara, Stefano Montanelli na Università degli Studi di Milano [14] slouží k procházení propojených dat. Je založena na koncepci inCloud, totiž vysokoúrovňového grafu tematicky vzájemně propojených vrcholů, které zastupují shluky spolu souvisejících propojených dat a kde hrany reprezentují blízkost a vzájemné vztahy mezi těmito celky.

Tento graf je konstruován algoritmem HCf+, který utváří pohled na data z perspektivy témat, a sice pomocí metod podobnostních kombinací. Dále nabízí sadu nástrojů pro intuitivní vyhledávání založené na klíčových slovech. Využít se dá i dotazovacích jazyků svou konstrukcí podobných SQL či MQL, back end aplikace totiž pracuje z IQL, tedy inCloud Query Language, jazykem šitým na míru platformě, na které je projekt vystavěn.

Systém je charakterizován třemi následujícími vlastnostmi:

Abstrakce pomocí agregace podobných provázaných dat přese své vlastnosti, každý z takových shluků je popsán hodnotou důležitosti, s jakou vystupuje v rámci celého cloudu; mezi uzly shluků figuruje hodnota blízkosti (podobnosti) společně se stupněm blízkosti.

Objevování průchodem má dvě možné varianty:

1. *vnitřní procházka* po obsahu shluku za účelem prozkoumání jeho vlastností do hloubky a
2. *tematická procházka*, která zastupuje průchod po povrchu, čili dívá se na shluk jako na celek spolu s jeho přilehlým okolím s určenými hodnotami vzdáleností.

Filtrování pomocí dotazů nebo klíčových slov.

Pozadí aplikace tvoří extraktor informací například z DBpedie či Freebase, jenž propojená data zároveň přizpůsobují pro zpracování v „procházkách“. Klasifikátory pak člení data dle podobností po dvojicích nástrojem HMatch do hierarchických klastrů. Abstrakční manažer pak konečně zkonstruuje inCloud graf a připraví data pro jejich vizualizaci.

Webové uživatelské rozhraní v HTML5 s javascriptovými doplňky pro vizualizaci prezentuje vytvořený graf, jehož uzly jsou tvořeny jmény z dvojic název-typ a jejichž velikosti odpovídají ohodnocení číslem důležitosti (kvality) shluku. Pro zvolené uzly se zobrazují relevantní informace s popisem, jaký zdroj zastupuje. Skrze formulář na webové stránce se též provádí vyhledávání klíčovými slovy nebo v editoru dotazů lze psát jazykem podobným SQL.

2.5 Résumé diskutovaných systémů

Ucelený pohled na diskutované systémy předkládá tabulka 2.1. Jak patrno, všechny systémy, až na inWalk, čerpají data z DBpedie, ať již jako z hlavního zdroje, případně jako součást několika zdrojů, či formou zdroje doplňkového. Společným jmenovatelem všech je ovšem nějaká forma vyhledávání a procházení obsahu, struktury propojených dat, s výběrem, respektive doporučováním, relevantního obsahu. Většina systémů umožňuje objevování nových informací a vztahů mezi daty postupným traversováním jejich struktur a též obohacuje následné kroky (spolu se zobrazováním výsledků) o zpětnou vazbu ze strany uživatele, který tak svou činností (výběrem, hodnocením) zvyšuje přesnost a relevanci předkládaného obsahu v dalších krocích.

Systémy udržují i svůj vnitřní stav – historii – paměť prováděných operací a získaných výsledků, čímž optimalizují množství úkonů, které se mnohdy opakují, takto snižují zátěž serverů, z nichž si opatřují data. Informace tohoto charakteru se uchovávají buď během jednoho každého sezení (session), nebo dlouhodobě odděleně pro každého uživatele zvlášť v rámci jeho osobního profilu (vyžadována registrace do systému).

Způsob, jakým systémy zobrazují uživatelům výsledky své činnosti, se u každého různí, neboť každý má jiný účel. Vždy se ale dbá na takovou interpretaci, aby byli uživatelé schopni porozumět předávaným informacím a aby byli obohaceni o nové znalosti. Někde se jedná o prosté textové zobrazení ve formě seznamu relevantních odkazů s popisy, jindy jde o množinu tagů nebo graf s vazbami mezi entitami. Vyjádření důležitosti daných prvků výsledku se promítá do jejich grafické reprezentace v podobě řezů a stupňů písma, velikosti či sytosti barev vrcholů grafu a tak podobně.

Všechna rozhraní zmíněných systémů jsou webová založená na HTML s několika prvky i celými frameworky vybudovanými na JavaScriptu, Semantic Wonder Cloud pak obsahuje i technologie Flash. Vylepšení o fasetovou navigaci přinášejí konkrétně Yovisto, Seevl a Discovery Hub. Ta nabízí možnost lepšího výběru a selekce z nabídnutých výsledků aplikací filtrů na tyto položky.

Závěrem se sluší zmínit i velké projekty, které taktéž pracují s propojenými daty a využívají jich pro doporučování. Například Google Knowledge Panel (2012) běžící na platformě Google Pregel pro zpracovávání velkých grafů nebo Bing Snapshot (2013) používající Microsoft Research's Trinity graph engine. Dále přichází vyhledávač Yahoo! s doporučovacím systémem SPARK operující s daty Twitteru a Flickeru. Jejich algoritmy jsou však drženy v tajnosti, jelikož jde o know-how, obecně se ale dá říci, že pracují s metrikami v grafech, rozhodovacími stromy, hodnocením prokliků a dalšími statistickými informacemi [37].

Tabulka 2.1: Přehled diskutovaných systémů [27]

Název	Yovisto	SWOC	LED	Aemoo	Seevl	Discovery Hub	Linked Jazz	inWalk
Rok	2009	2010	2010	2012	2012	2013	2013	2014
Zdroj dat	DBpedia	Dbpedia	Dbpedia	DBpedia	Dbpedia, Freebase, MusicBrainz	DBpedia	Dbpedia, Linked Jazz DB	Freebase, Twitter
Doplňkový zdroj dat	ne	vyhledávače, tagovací systémy	vyhledávače, tagovací systémy	externí služby	ne	ne	ne	ne
Dotazovací model	klíčová slova	Dbpedia lookup	klíčová slova	vyhledávání	vyhledávání	Dbpedia Lookup	ruční výběr	vyhledávání, výběr
Vyhledávání	řetězcové shody	přímá shoda	řetězcové shody	přímá shoda	přímá shoda	přímá shoda	výběr	přímá shoda
Oblast	videa	informační technologie	informační a komunikační technologie	obecná	hudba	obecná	jazzoví hudebníci	atleti, novinky na Twitteru
Typ úložiště	Freebase Parallax	SWOC Storage	LED Storage	Knowledge Pattern Repository Manager	OpenLink Virtuoso	Virtuoso, MySQL	Linked Jazz Name Directory	inWalk repository
Účel úložiště	mapování dotazů na entity	popularita a podobnost dvojic zdrojů	výsledky hodnocení	indexování	škálovatelnost	uživatelé, průchody	trojice literálů	vysokoúrovňový pohled na data
Prezentace výsledků	návrhy dotazů	graf	množina tagů	graf	seznam	seznam	graf	graf
Popis výsledků	ne	ne	ne	užívající Wikipedii	sdílené vlastnosti	text, graf	ano	ne
Stav	session, registrace	session	session	session	session, registrace	session, registrace	session	session
Algoritmy	množina heuristik	DBpediaRanker	DBpediaRanker	pohled na vzory znalostí	LDSD, DBrec algorithm	Semantic spreading activation	Mapping, Curator Tool, Transcript Analyzer	HCf+ clustering algorithm
Skórování, hodnocení	ano	velikost grafu	ano	ne	ano	ano	ne	ne
Offline zpracování	ano	podobnost dvojic	ano	ano	ano	ne	ano	ano
API	RDF triple-store	ne	RESTful	RESTful	dohadování obsahu JSON-LD	ne	JSON, RDF, GEXF files	ne
Fasetová navigace	ano	ne	ne	ne	ano	ano	ne	ne
Uživatelské rozhraní	HTML	Flash	HTML	HTML	HTML, Ajax	HTML	HTML5 + jQuery	HTML5 + JS

Kapitola 3

Analýza

Problematika algoritmů pracujících s propojenými daty na úrovni jejich grafové struktury obsahuje dva okruhy, kterým je třeba věnovat pozornost a analyzovat je pro další postup. Prvním hlediskem je jejich skutečná podoba: to, jak jsou formátována, organizována, způsob, jakým jsou na webu zpřístupněna, a v neposlední řadě také jak jsou rozsáhlá a co je jejich obsahem. Druhou oblastí jsou vlastní algoritmy pro průchod grafovými strukturami, jejich druhy, modifikace, způsoby ohodnocování vrcholů, možnosti souběžného spouštění na tomtéž grafu a postupy při shromažďování výsledků.

3.1 Data

Otevřená propojená data sémantického webu, tedy publikační model stanovující způsoby, pravidla a doporučení, kterak zveřejňovat strukturovaná data v prostředí webu, splňující kritéria označená pěti hvězdičkami zaručují automatizovanou zpracovatelnost strojovým způsobem. V tomto směru je třeba se zaměřit na jejich určitá specifika, která přinášejí, neboť od nich se musejí odvíjet způsoby přístupu k nim samotným a k jejich zpracování.

3.1.1 Rozsáhlost

Velmi podobně jako roste rozlehlost webu dokumentů, roste i množství metadat sémantického webu, tudíž není možné obsáhnout je všechna naráz. Vždy je však možné začít z jediného URI a postupně procházet grafem propojených dat. Pro zajištění mantinelů, v rámci nichž se chceme grafem pohybovat, je možné využít například omezení v podobě domény, z jejíhož názvu jsou dané odkazy složeny, případně se také můžeme zaměřit na určité typy zdrojů, a sice dle jejich RDF popisu s využitím tříd, vlastností či hodnot, jež je popisují.

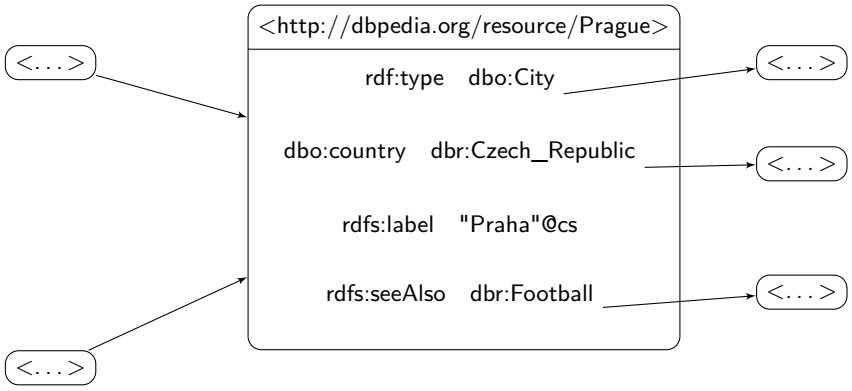
3.1.2 Přístupnost

Data by měla být zveřejněna pod otevřenou licencí – například Open Data Commons: Public Domain Dedication and License (PDDL), Attribution License (ODC-By), Open Database License (ODC-ODbL) či Creative Commons.

Přístup k fyzickým datům pak může být realizován dvěma způsoby. Známe-li konkrétní URI, jeho dereferencí můžeme získat odkaz na soubor s popisem tohoto zdroje v některém z formátů serializace RDF. Druhou možností je využití SPARQL enpointu. Výhoda veřejného koncového bodu pro dotazování se nad RDF daty spočívá v možnosti efektivně pracovat s určitými částmi dat, a tím se vypořádat s jejich přílišnou rozsáhlostí.

3.1.3 Organizovanost

Struktura otevřených propojených dat je z principu grafová. Pohlížet je na ni možné jako na graf o dvou pomyslných úrovních (viz obrázek 3.1). Ve svrchní vrstvě rozeznáváme jednotlivé identifikátory zdrojů (vrcholy grafu), druhá vrstva je následný RDF graf s popisem onoho zdroje. Získaná metadata o zdroji poskytují, mimo jiné, další URI, čímž definují přechody (hrany grafu) mezi zdroji.



Obrázek 3.1: Struktura propojených dat

3.1.4 Obsah

Popis zdroje v jisté RDF serializaci neobsahuje výhradně trojice s URI na místě objektu, vyskytují se zde také literály (hodnoty) či prázdné uzly. Kromě metadat o příslušném zdroji se dá natrefit na popis, který se týká jiných zdrojů, i na redundantní trojice popisující tutéž skutečnost vícekrát.

3.1.5 Rizika

Během celého životního cyklu propojených dat se mohou vyskytnout situace, se kterými se koncový uživatel při jejich získávání a zpracovávání musí vypořádat. Data totiž nemusejí být nepřetržitě dostupná. Náhlé výpadky serverů anebo pravidelné odstávky je mohou dočasně znepřístupnit. Dále může jít o data zastaralá, neaktuální, mohou obsahovat chyby, ať již obsahové, tak i syntaktické (nedovolené znaky), neúplné informace nebo data nesprávně formátovaná či kódovaná.

Soubory s popisy zdrojů mohou mít v závislosti na zvolené serializaci různě velkou datovou objemnost, stejně i výsledek SPARQL dotazu. Navíc endpointy pro dotazování mívají velmi často nastaven časový limit pro běh dotazu, po jehož uplynutí obdrží uživatel chybové hlášení namísto kýžených dat.

3.2 Grafové algoritmy

Dříve, než přistoupíme k samotným algoritmům, je třeba určit struktury, nad kterými by měly pracovat; těmi jsou již zmiňované grafy. V propojených datech se přesněji jedná o hranově ohodnocený (značený) orientovaný multigraf.

Hranově ohodnocený orientovaný multigraf je uspořádaná trojice $G = (V, E, l)$, kde V je neprázdnou množinou vrcholů, $E \subseteq V \times V$ je multimnožina uspořádaných dvojic prvků z množiny vrcholů a $l : M \mapsto E$ je zobrazení ze značkovací množiny do množiny hran přiřazující každé hraně příslušné označení [40].

V grafovém popisu propojených dat se vyskytuje i lehce upravená definice hranově značeného orientovaného multigrafu, a sice, že jde o uspořádanou trojici $G = (V, E, L)$, přičemž V je neprázdná množina vrcholů, L je značkovací množinou a $E \subseteq V \times L \times V$ je množinou hran sestávající z uspořádaných trojic (*počáteční vrchol, označení hrany, koncový vrchol*). Tato definice lépe reflektuje skutečnou reprezentaci dat v RDF.

Interpretace matematické definice grafu propojených dat je tedy zcela přímočará: vrcholy (množina V) odpovídají jednotlivým prvkům dané domény (subjekty, objekty), které se účastní popisu; prvky značkovací množiny L jsou binární relace určující vztahy mezi dvěma prvky a konečně množina trojic E vyjadřuje skutečnosti, známá fakta, o daných prvcích.

3.2.1 Procházení grafem

Prohledávání grafu, traverzování či procházení grafem, je základní činností prováděnou nad grafovými strukturami. Vychází ze dvou základních metod,

jejichž hlavní výhoda spočívá v tom, že není třeba znát celkovou strukturu grafu ani jej mít kompletně k dispozici, stačí pouze, aby byl v každém kroku algoritmu navštíven jeden vrchol (výchozím počínaje), u něhož jsme s to identifikovat s ním incidující hrany a objevit pomocí nich jeho sousední uzly. Předpokladem je tedy souvislost grafu, neboť prohledávání se uskutečňuje pouze v jeho souvislé komponentě, pomineme-li zcela nahodilé procházení množiny vrcholů ignorující hrany [26].

Pseudokód 1 Prohledávání do šířky/hloubky

Require: $v_{init} \in V(G)$
1: $Q \leftarrow v_{init}$ ▷ $Q \Rightarrow$ FIFO/LIFO
2: **while** $Q \neq \emptyset$ **do**
3: $v \leftarrow v_1 \in Q$
4: **for all** $w \in \{x \in V(G) \mid \{v, x\} \in E(G)\}$ **do**
5: $Q \leftarrow w$
6: **end for**
7: **end while**

Podmínkou pro zastavení výše uvedeného algoritmu je pouze prázdná struktura uchovávající uzly, které dosud nebyly zpracovány. Obecně však nejsou řešeny případy, kdy se v grafu vyskytují násobné hrany (multigraf), které by způsobovaly neustálé plnění fronty/zásobníku a algoritmus by tak nemohl být ukončen. Řešením je například udržování seznamu již navštívených vrcholů, které do fronty či zásobníku dále znovu neumisťujeme, případně se algoritmus dá ukončit při dosažení určitého počtu zpracovaných vrcholů, nebo dosáhne-li předem stanovené hloubky (patra) uzlů v grafu.

3.2.1.1 Prohledávání do šířky

První variantou procházení grafem je algoritmus *prohledávání do šířky*, anglickým názvem *breadth-first search*, zkráceně BFS. Ten vychází z postupného procházení sousedních vrcholů jednoho aktuálně zpracovávaného, a to v pořadí, ve kterém byli tito sousedé zaznamenáni. Za účelem poznačení dosud nezpracovaných vrcholů se užívá *fronty* – datové struktury na bázi FIFO, v ukázkovém kódu 1 se jedná o prvek Q.

3.2.1.2 Prohledávání do hloubky

Druhým typem je *prohledávání do hloubky*, anglicky *depth-first search* (DFS) pracující na stejném principu vyhledávání sousedních vrcholů jako BFS, avšak tyto se ukládají do zásobníku (LIFO struktura, v pseudokódu 1 opět Q), tudíž naposledy vložený sousední vrchol je zpracováván jako první, čímž se celý algoritmus neustále zanořuje hlouběji do grafu.

3.2.2 Union colours algorithm

Česky *algoritmus slučování barev*. Principiálně se jedná o algoritmus prohledávání do šířky navíc obohacený o barvení jednotlivých vrcholů dle následujícího schématu:

1. Všechny počáteční vrcholy označ nějakou barvou;

2. z každého počátečního uzlu spusť prohledávání do šířky tak, že

   a) vlož počáteční vrchol do fronty,

   b) vyjmi z fronty první vrchol v řadě,

   c) zpracuj současný uzel tak, že

      i. nemá-li dosud přiřazenu žádnou barvu, obarvi jej barvou, která přísluší počátečnímu vrcholu, nebo

      ii. pakliže je již obarven, sluč současnou barvu s barvou počátečního uzlu;

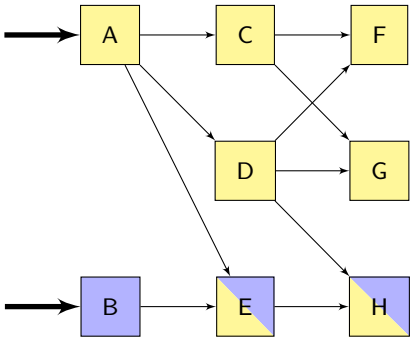
   d) najdi všechny jeho sousední vrcholy a vlož je do fronty,

   e) pokračuj znovu bodem b), dokud není splněna nějaká zastavující podmínka.

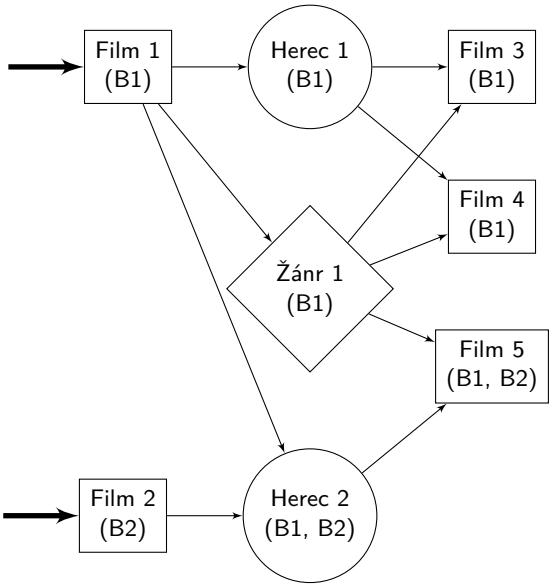
Konstrukce algoritmu umožňuje současný běh z několika počátečních uzlů zároveň. Samotné slučování barev je pak možné zajistit v konstantním čase užitím speciální datové struktury pracující s disjunktními množinami. Ve skutečnosti se ale „barva“ jako taková nahrazuje prostým číslem, a tedy operace sloučení je převedena do reálné aritmetiky.

Obrázek 3.2 ilustruje výsledek běhu algoritmu nad uvedeným grafem. Počáteční uzly A a B (označené šipkami větší tloušťky) jsou ve výchozím stavu obarveny různými barvami. Hrany grafu jsou orientované pro snazší čitelnost, jakým směrem postupovalo barvení. Pro vlastní algoritmus však není orientace nezbytně nutná.

Aplikací algoritmu slučování barev na graf propojených dat v doméně filmů je možné získat ohodnocení uzlů, jaké představuje obrázek 3.3. Nejvyšší relevance dosáhne takový film, jemuž je přisouzeno nejvíce různých barev. Pakliže se rozhodneme pro všechny barvy uvažovat pouze jediné jednotkové číslo a operaci sloučení stanovíme jako prostý součet, registrujeme u jednotlivých uzlů počet barev, kterými jsou obarveny. V případě, že použijeme čísla z intervalu $\langle 0; 1\rangle$, normalizované váhy, pak se algoritmus spíše přiblíží *algoritmu mísení barev*, který se od slučování liší jenom v interpretaci barev a operacemi s nimi.



Obrázek 3.2: Algoritmus slučování barev



Obrázek 3.3: Algoritmus slučování barev aplikovaný na propojená data – oblast filmů [19]

Nejvyšší důležitost ze všech filmů získal na obrázku 3.3 vrchol s názvem „Film 5“, který je obarven dvěma barvami, ostatní („Film 3“ a „Film 4“) jsou na stejné úrovni relevance.

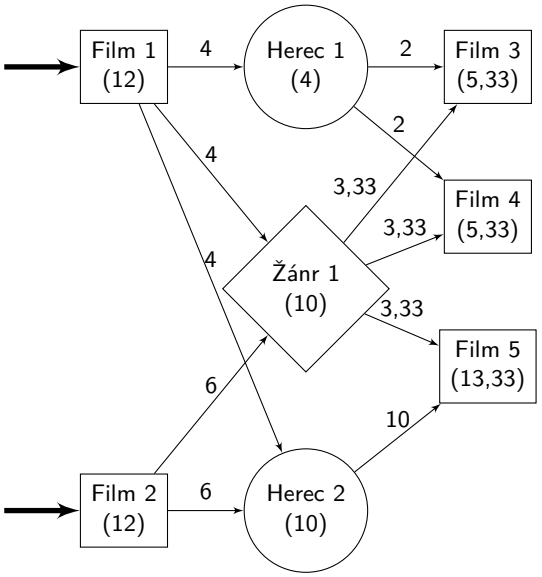
3.2.3 Energy spreading algorithm

Dalším zástupcem z řady algoritmů vycházejících z prohledávání do šířky je *energy spreading algorithm* (volně přeloženo jako *algoritmus rozprostírání energie*). S úspěchem se využívá k prohledávání v sémantických (asociativních) sítích a sítích neuronových [19].

Podstata algoritmu je již obsažena v jeho názvu. Na počátku je vybraným vrcholům grafu předána iniciální energie určité velikosti, která se následně rozlévá do jejich sousedních uzlů a ve společných uzlech se opět stéká dohromady. Celá logika rozprostírání energie sestává z následujícího sledu kroků:

1. Nastav vybraným počátečním uzlům pevné množství energie;
2. všechny dosud nenavštívené vrcholy nechť mají hodnotu energie rovnu nule;
3. ze startovních vrcholů spusť prohledávání do šířky tak, že
   a) energii E současného uzlu rozděl mezi jeho n sousedů,
   b) každému sousednímu vrcholu zvyš energii právě o hodnotu $\frac{E}{n}$.

Energie každého vrcholu se může zvyšovat libovolně i několikrát za sebou, k jejímu rozprostření ale dochází pouze při prvním zpracování uzlu. Z obrázku 3.4 je patrné, jak vypadají energetické toky v jednotlivých větvích grafové struktury. Čísla uvedená u hran neudávají jejich pojmenování či váhu, nýbrž ozřejmují podíl energie předávaný sousednímu uzlu s touto hranou incidujícímu. Výchozí vrcholy jsou i zde určeny výraznou šipkou. Aplikováním na oblast filmů je vybrán takový uzel, jemuž po doběhnutí algoritmu přísluší nejvyšší hodnota energie, tedy přeneseně důležitosti (uzel s názvem „Film 5“).



Obrázek 3.4: Energy spreading – oblast filmů [19]

Při zahájení běhu rozprostírání energie se algoritmu předají všechny počáteční uzly, které se za sebou umístí do fronty dosud nezpracovaných vrcholů. Záleží na charakteru aplikace a na programátorovi, jak se vypořádá s eventuálními zpětnými smyčkami, které by v určitých uzlech grafu mohly způsobit kumulování velkého množství energie. Přístupů, jak takové situaci předejít je několik:

- Kružnice jsou akceptovány, omezena je pouze hloubka prohledávání grafové struktury.
- Smyčky jsou eliminovány pomocí záznamů o již zpracovaných vrcholech.
- Energie se v cyklech předává, avšak z daných uzlů se šíří pouze při jejich prvním zpracování.

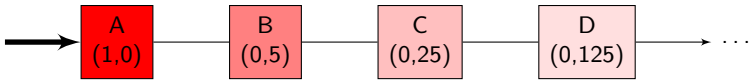
3.2.4 Spreading activation

Do rodiny „spreading“ algoritmů patří i *spreading activation* čili *šířené (postupné) vybuzování (aktivace)*, nicméně jeho konstrukce vychází z prohledávání do hloubky. V praxi se používá jako nástroj pro vytěžování mnohodimenzionálních socio-sémantických sítí, a to především za účelem získávání informací o podobnostech uživatelů, relevanci zdrojů (odkazů, linků) nebo též pro určování metrik centrality uzlů u metod zaměřených na shlukování [18].

Počátečním vrcholům je nastavena budicí energie, která se postupně předává sousedním uzlům. Algoritmus ovšem navíc obsahuje parametr pro tlumení účinku aktivační energie, pročež s narůstající hloubkou zanoření (a počtem uzlů v cestě) efekt budicí síly klesá. Zastavující podmínkou pro běh algoritmu, krom klasických, jakými jsou hloubka zanoření, počet zpracovaných uzlů nebo probrání se celým grafem, je zde i práh stanovující množství energie, s níž má ještě smysl daný uzel zpracovávat.

Každému uzlu anebo hraně vedoucí k tomuto uzlu může také náležet váha, se kterou je schopen tento vrchol přijmout budicí hodnotu (někdy se jedná o jednotnou hodnotu pro všechny uzly). Tlumicí činitel i váhové koeficienty se nejčastěji volí v hodnotách z intervalu $\langle 0; 1\rangle$ reálných čísel.

1. Přiřaď startovním uzlům jejich počáteční aktivační hodnotu (> 0);
2. ostatním uzlům nastav budicí hodnoty rovny nule (uzly jsou dosud nevybuzené);
3. stanov prázdný nezávislý seznam pro definitivní výsledky dvojic: uzel se svou aktivační hodnotou;
4. z každého výchozího uzlu spusť prohledávání do hloubky tak, že



Obrázek 3.5: Spreading activation na jednoduchém grafu

a) vlož počáteční vrchol do zásobníku,

b) vyzvedni první uzel svrchu zásobníku,

c) nalezni všechny jeho sousední uzly,

d) každému sousedovi vypočti aktivační hodnotu dle vztahu

$$a'_j = a_j + a_i \cdot w \cdot d, \tag{3.1}$$

kde a_i je budicí hodnota současně zpracovávaného uzlu, a_j je aktivační hodnota j-tého potomka současného uzlu, w je váhový koeficient a d je činitel tlumení – výsledek se po dosazení prohlásí za novou aktivační hodnotu a'_j j-tého potomka současného uzlu,

e) ulož postupně všechny sousední vrcholy do zásobníku,

f) pokračuj znovu bodem b), dokud není splněna nějaká z ukončovacích podmínek;

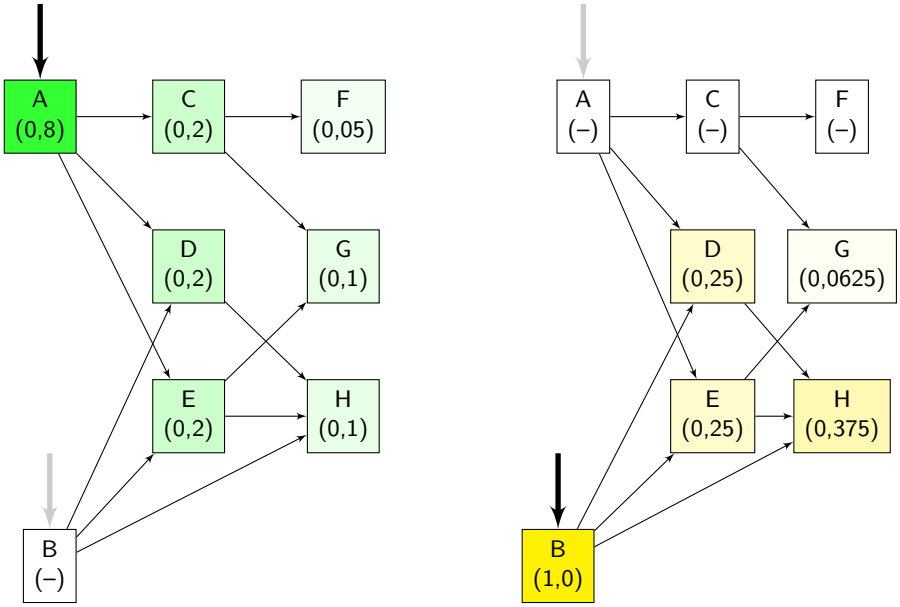
5. pro každý ohodnocený vrchol proveď:

a) není-li dosud přítomen ve výsledném seznamu, vlož jej tam i s jeho aktivační hodnotou,

b) pokud již v seznamu je, přičti jeho současnou aktivační hodnotu k hodnotě ve výsledném seznamu.

Poslední položka (číslo 5) seznamu o odstavec výše dovoluje souběžné spuštění algoritmu z několika počátečních uzlů zároveň. Výsledky se umisťují do globálního seznamu, v němž se kumulují aktivační hodnoty daných uzlů.

Na jednoduchém grafu se dá znázornit průběh postupného vybuzování nejen ohodnocením vrcholů, ale také intenzitou jejich podbarvení, jak vidno z obrázku 3.5. Zde je nastavena globální váha na hodnotu 1,0 a činitel tlumení na 0,5. Naproti tomu na obrázku 3.6, kde je již složitější graf, v němž je možné se dostat do některých uzlů i více než jednou cestou, nabývá globální váha hodnoty i činitel útlumu hodnoty 0,5. Algoritmus v tomto případě prochází vrcholy předtím už jednou navštívené, a to i několikrát po sobě – tak, jak mu to graf umožňuje. Graficky jsou znázorněny obě fáze průchodu z každého jednoho počátečního vrcholu zvlášť. Výsledný seznam z kroků ilustrovaných na obrázcích 3.6(a) a 3.6(b) vypadá takto: `[A=0,8; B=1,0; C=0,2; D=0,45; E=0,45; F=0,05; G=0,1625; H=0,475]`.



(a) Buzení z bodu A (b) Buzení z bodu B

Obrázek 3.6: Postupné vybuzování z každého počátečního vrcholu

3.2.5 Upravený Dijkstrův algoritmus

Klasický *Dijkstrův algoritmus* se užívá k vyhledání nejkratší cesty v nezáporně hranově ohodnoceném grafu [40]. I zde je podstatou prohledávání do šířky, necháme-li jej tedy projít celým grafem, získáme optimální cesty do všech vrcholů z jistého počátečního. Následující popis prostého Dijkstrova algoritmu zahrnuje návod pro jeho realizaci:

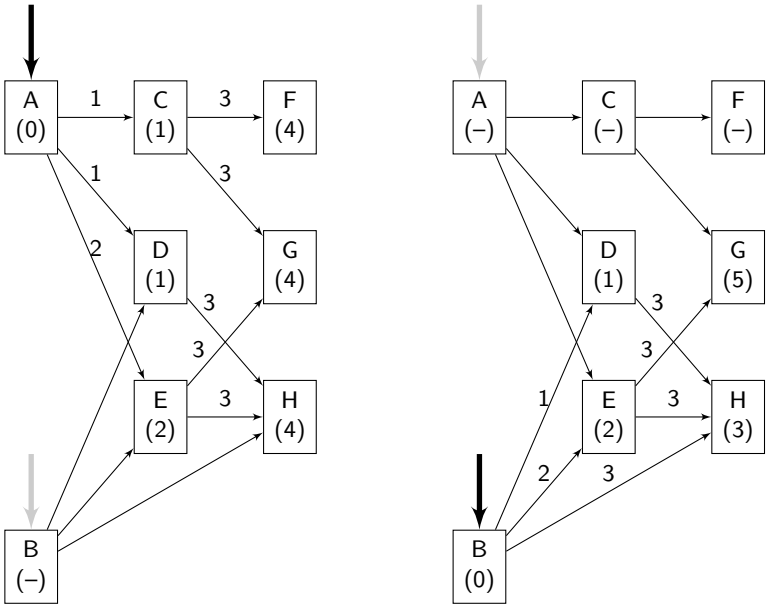
1. Počátečnímu vrcholu nastav vzdálenost rovnu nule, všem ostatním neprozkoumaným vrcholům nastav vzdálenost od počátečního uzlu na hodnotu $+\infty$;
2. vlož počáteční uzel do fronty nezpracovaných vrcholů;
3. spusť prohledávání do šířky tak, že
   a) vyjmi první vrchol z fronty (právě zpracovávaný);
   b) nalezni všechny jeho sousedy a pro jednoho každého proveď:
      i. vlož jej do fronty nezpracovaných uzlů,
      ii. zjisti vzdálenost od nyní zpracovávaného uzlu k tomuto sousedovi a přičti ji k hodnotě právě zpracovávaného uzlu:

$$d_a = d(u) + w(\{u, v\}), \tag{3.2}$$

kde $u, v \in V(G)$ jsou vrcholy grafu, u je aktuálně zpracovávaný, v je sousedním vrcholem u, tudíž $\exists\{u, v\} \in E(G)$ (existuje hrana mezi u a v), $d(x)$ vrací dosud nalezenou optimální vzdálenost z počátečního uzlu do vrcholu $x \in V(G)$ a $w(e)$ vrací hodnotu hrany $e \in E(G)$;

iii. jestliže je vypočtená vzdálenost d_a lepší, než je hodnota optimální cesty do souseda v, tedy $d_a < d(v)$, aktualizuj hodnotu $d(v)$ sousedního uzlu v na hodnotu d_a;

c) pokračuj bodem a), dokud není splněna nějaká z ukončovacích podmínek



(a) Z bodu A (b) Z bodu B

Obrázek 3.7: Upravený Dijkstrův algoritmus z každého počátečního vrcholu

Modifikace algoritmu spočívá v možnosti spustit jej z několika počátečních uzlů současně. K tomu je ovšem zapotřebí výsledného seznamu, do něhož se zapíší výsledky běhů z oněch počátečních vrcholů, a proto v upravené verzi následuje za bodem 3 ještě:

4. Po dokončení části prohledávání do šířky s ohodnocováním vrcholů zapiš výsledek, totiž všechny zpracované vrcholy s jejich hodnotami optimální cesty, do výsledného seznamu; pakliže už je v seznamu ohodnocený vrchol přítomen, přičti k jeho hodnotě tuto nyní zapisovanou.

Volitelně je ještě možné aplikovat bod 5:

5. Vyber do výsledného seznamu pouze takové vrcholy, které byly navštíveny ze všech počátečních uzlů.

Situace na obrázku 3.7 představuje stav ohodnocení vrcholů grafu nejprve po spuštění algoritmu z uzlu A, následně pak z uzlu B (oba s počátečními hodnotami 0). Výsledný seznam ohodnocených vrcholů by měl v tomto případě podobu: `[A=0, B=0, C=1, D=2, E=4, F=4, G=9, H=7]`. Po aplikaci pravidla, že se ve výsledku smí objevit jen uzly navštívené ze všech startovních vrcholů, by seznam vypadal takto: `[D=2, E=4, G=9, H=7]`.

Protože upravený Dijkstrův algoritmus pro svou činnost vyžaduje nezáporně ohodnocené hrany grafu, je do původně neohodnoceného grafu třeba váhy zavést uměle. První možností je nastavit jejich hodnoty globálně na jedinou pevnou hodnotu, druhou variantou, poněkud sofistikovanější, je určení různých hodnot hranám mezi významnými uzly a ostatními vrcholy. Pro doménu filmů v propojených datech se naskýtá možnost zvýhodnit například všechny hrany ukazující na URI identifikující konkrétní film, zatímco hrany vedoucí k osobám (hercům, režisérům apod.) nebo napojující žánr, do něhož filmy spadají, není třeba nabízet s tak vysokou relevancí.

Úloha nejkratší cesty v grafu s sebou přináší i otázku, zda je možné nalézt cestu nejdelší. Na obecném grafu se jedná o problém NP-těžký, přesto je možné použít Dijkstrův algoritmus, avšak pouze na grafu se stromovou strukturou (minimální souvislý graf bez kružnic). S upravenou podmínkou pro výběr optimální vzdálenosti do následujícího vrcholu a počátečním nastavením vzdáleností do nenavštívených uzlů na hodnotu $-\infty$ se zachováním nezápornosti vah hran je možné aplikovat Dijkstrův algoritmus i na takto definovanou úlohu [64].

3.2.6 Bellmanův–Fordův–Mooreův algoritmus

Algoritmus běžně v literatuře uváděný jako *Bellmanův–Fordův* spadá do ranku algoritmů pro výpočet nejkratší cesty v hranově ohodnoceném grafu. Přestože je schopen pracovat s jakýmkoli ohodnocením, tedy i se zápornými váhami hran, a proto má širší pole působnosti, na stejném problému je pomalejší nežli Dijkstrův algoritmus [17].

1. Počátečnímu vrcholu nastav vzdálenost rovnu nule, všem ostatním neprozkoumaným vrcholům nastav vzdálenost od počátečního uzlu na hodnotu $+\infty$;

2. do následujícího vrcholu v je možné se dostat za cenu hodnoty vzdálenosti současného vrcholu u zvýšené o váhu $w(u, v)$ hrany mezi těmito

dvěma uzly. Hodnota $d(v)$ následujícího uzlu v se pak stanoví takzvanou *relaxací*

$$d(v) = \min\left\{d(v);\ d(u) + w(u,v)\right\}, \tag{3.3}$$

a to v tolika iteracích, kolik je v grafu vrcholů sníženo o jedna, tedy $(|V|-1)$-krát, a zároveň v každé iteraci pro všechny hrany v grafu;

3. ověř, že se v grafu nevyskytují záporné cykly. Pokud ano, přeruš provádění chybou.

Záporný cyklus v grafu je taková (orientovaná) uzavřená cesta, tj. kružnice, ve které je součet ohodnocení hran záporný [26].

Nebezpečí záporného cyklu spočívá ve vytvoření nekonečné smyčky, v níž dochází k neustálému opakovanému zlepšování délky cesty vzhledem k optimalizačnímu kritériu.

Aby bylo možné algoritmus nasadit, vyžaduje úplný seznam vrcholů i hran s jejich ohodnocením, pracuje totiž na kompletním známém grafu. Tato skutečnost jej činí nevhodným pro použití na grafu propojených dat bez předchozího prohledání a objevení alespoň výseku dané struktury, podgrafu, s nímž je již možné pracovat. Ohodnocení hran se dá přenést z vlastností uzlů, se kterými inciduje – tak, jak je tomu i u Dijkstrova algoritmu.

Algoritmus je i v tomto případě aplikovatelný na problém hledání nejdelší cesty, a sice v orientovaném grafu bez kružnic. Všechny váhy hran je třeba nejprve znegovat (násobením hodnotou -1), dále pak upravit nerovnost v relaxační podmínce a inicializovat všechny vzdálenosti do dosud nenavštívených vrcholů hodnotou $-\infty$. Také je třeba ověřovat existenci kladného cyklu [66].

3.2.7 Floydův–Warshallův algoritmus

V rodině algoritmů pro hledání nejkratších cest v grafu setrváváme i nadále. Jejím dalším členem je *Floydův–Warshallův algoritmus*, který je určen pro výpočet nejkratších cest mezi každými dvěma vrcholy. Postupně se v jednotlivých krocích vylepšují dosud vypočtené vzdálenosti mezi všemi dvojicemi vrcholů najednou. Graf, nad nímž algoritmus operuje, může obsahovat záporně ohodnocené hrany, ale nesmí se v něm vyskytovat, stejně jako v případě Bellmanova–Fordova–Mooreova algoritmu, záporný cyklus [26].

Vstupem pro algoritmus je matice sousednosti $\mathbf{A}$: čtvercová matice $n \times n$, kde $n = |V|$ je počet vrcholů a její prvky $a_{i,j}$ mají hodnoty

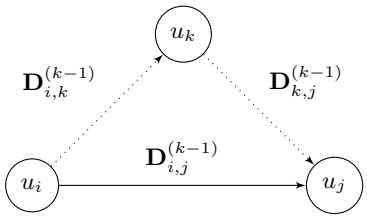
$$a_{i,j} = \begin{cases} 1, & \text{pokud } (\exists e \in E(G))(e = \{i,j\} \wedge i,j \in V(G)), \\ 0, & \text{jinak.} \end{cases} \tag{3.4}$$

Z matice sousednosti se vypočítá matice vzdáleností (případně může být tato přímo vstupem algoritmu) $\mathbf{D}$ též rozměru $n \times n$. Na počátku je inicializována v nulté iteraci $\mathbf{D}^{(0)}$ dle struktury grafu (matice sousednosti) tak, že

$$d_{i,j}^{(0)} = \begin{cases} 0, & \text{pokud } i = j \wedge \{i,j\} \notin E(G), \\ w(i,j), & \text{pakliže } \{i,j\} \in E(G), \\ +\infty & \text{v ostatních případech,} \end{cases} \qquad \text{pro } i,j \in V(G), \tag{3.5}$$

přičemž $w(i,j)$ je váha hrany mezi vrcholy i a j. Matice tak v každé iteraci k vyjadřuje vzdálenosti do všech uzlů, do nichž vede cesta s k mezilehlými vrcholy. Transformace matice $\mathbf{D}$ je vyjádřitelná rekurentním vztahem 3.6.

$$\mathbf{D}_{i,j}^{(k)} = \min\left\{\mathbf{D}_{i,j}^{(k-1)};\ \mathbf{D}_{i,k}^{(k-1)} + \mathbf{D}_{k,j}^{(k-1)}\right\}, \quad i,j,k = 1,\ldots,n. \tag{3.6}$$



Obrázek 3.8: Rekurentní vztah 3.6 aplikovaný na strukturu grafu [17]

Výstupem programu je matice $\mathbf{D}^{(n)}$, pseudokód 2 představuje kroky algoritmu. A protože obsahuje tři vnořené cykly, jedná se o pomalejší algoritmus než Bellmanův–Fordův–Mooreův [17].

Pseudokód 2 Floydův–Warshallův algoritmus

```
Require: D
for k ← 1,...,n do
    for i ← 1,...,n do
        for j ← 1,...,n do
            D(i,j) = min(D(i,j), D(i,k) + D(k,j))
        end for
    end for
end for
```

Taktéž zde je nevýhodou pro nasazení na předem neznámý graf, že pro běh algoritmu je třeba znát kompletní seznam uzlů a hran alespoň určitého podgrafu. Pro zpracování grafu propojených dat by bylo nutné nejprve prohledat danou část jejich struktury, teprve poté aplikovat Floydův–Warshallův algoritmus.

3.2.8 A* algoritmus

Příbuzným algoritmem k Dijkstrovu, ze kterého přímo vychází, je *A* (A star)*, algoritmus určený pro vyhledávání optimálních cest v grafech, tentokráte však z předem definovaného počátečního vrcholu do určeného cílového vrcholu. Jeho jádrem je dijkstrovské prohledávání do šířky, navíc ale přidává obohacení o heuristickou funkci $h(x)$. Skóre uzlu se pak vypočítá funkcí $d(x)$, která zastupuje délku dosud nejlepší cesty. Výběrová funkce $f(x) = d(x) + h(x)$ určuje hodnotu pro volbu dalšího vrcholu určeného ke zpracování. Veškeré funkce jsou definovány pro všechny vrcholy $x \in V(G)$ [17].

1. Urči prázdnou množinu uzavřených uzlů a množinu otevřených uzlů, do množiny otevřených uzlů na počátku vlož startovní vrchol;

2. dokud není množina otevřených uzlů prázdná, prováděj prohledávání do šířky tak, že

   a) z množiny otevřených uzlů vyjmi takový vrchol u, který má nejmenší hodnotu $f(u) = d(u) + h(u)$, a vlož jej do množiny uzavřených uzlů;

   b) je-li u cílovým uzlem, přejdi na bod 3;

   c) pro všechny sousední uzly v právě zpracovávaného uzlu u, které nejsou v množině uzavřených uzlů,

      - vlož v do množiny otevřených uzlů a
      - zjisti váhy hran $w(u, v)$ s nimi incidujících, a jestliže $d(v) + w(u, v) < d(u)$, tj. nová cesta je lepší než dosud nalezená, pak aktualizuj $d(u)$ na hodnotu $d(v) + w(u, v)$;

   d) pokračuj bodem a);

3. vyhodnoť úspěch nalezení cílového vrcholu a cesty k němu.

Heuristická funkce $h(x)$ určuje vzdálenost od uzlu x do nejbližšího koncového uzlu. Její tvar se odvíjí od problému, na který je algoritmus aplikován. Hledáme-li například nejkratší vzdálenost mezi místy v mapě (dvourozměrný prostor), bude $h(x)$ zastupovat vzdálenost vzdušnou čarou, třeba definovanou pomocí eukleidovské metriky [36].

Bohužel užití algoritmu na problém procházení struktury propojených dat není vhodné, neboť vyžaduje kromě počátečního vrcholu i konkrétní vrchol koncový. Ba co více, je třeba také přesně definovat heuristickou funkci pro určování vzdáleností k cíli, což v takto specifickém stavovém prostoru nemusí být triviální záležitostí.

3.2.9 Náhodné procházení

Skutečným protipólem systematického procházení či prohledávání grafových struktur je zcela nahodilé procházení vrcholů (*random walks*), počínaje startovním uzlem, náhodným výběrem hran spojujících aktuální uzel s jeho sousedy. Mějme tedy vrchol $u \in V(G)$ neorientovaného grafu G, který má n různých sousedních vrcholů v_i, tedy existují různé hrany $e = \{u, v_i\} \in E(G)$, incidující s u a v_i pro všechna $i = 1, \ldots, n$. Potom stupeň $deg_G(u)$ vrcholu u v grafu G bude nabývat hodnoty $deg_G(u) = n$.

Náhodná procházka neorientovaným grafem G z počátečního vrcholu u do jednoho ze sousedních vrcholů v_i, je přesun po hraně $e = \{u, v_i\}$ z uzlu u do v_i s pravděpodobností $p_{u \to v_i} = \frac{1}{deg_G(u)}$. Toto lze aplikovat na každý uzel v grafu, to znamená, že v každém kroku k je možné se dostat z uzlu u_k do uzlu u_{k+1} s pravděpodobností $p_{u_k \to u_{k+1}} = \frac{1}{deg(u_k)}$. Pravděpodobnost přechodu do následujícího vrcholu tudíž závisí pouze na stupni aktuálního vrcholu.

Náhodná procházka je stochastický proces vyjádřitelný Markovovým řetězcem, kde pravděpodobnosti přechodů z vrcholu i do uzlu j jsou [46]

$$p_{i,j} = P(X_{k+1} = j \mid X_k = i) \tag{3.7}$$

a kde $X_0, X_1, \ldots$ jsou diskrétní náhodné veličiny nabývající hodnot ze spočetné množiny stavů, zde vrcholů grafu. Matice přechodů $\mathbf{P}_{i,j}$ pak pro každou dvojici vrcholů i a j určuje pravděpodobnost přechodu $p_{i,j}$ mezi i a j

$$\mathbf{P}_{i,j} = p_{i,j} = \begin{cases} \frac{1}{deg_G(i)}, & \text{pokud } \{i,j\} \in E(G) \wedge i, j \in V(G), \\ 0, & \text{jinak.} \end{cases} \tag{3.8}$$

Matice $\mathbf{P}^{(n \times n)}$ je stochastická. Pro každý její řádkový vektor platí, že součet hodnot jeho prvků je roven jedné: $\sum_j p_{i,j} = 1;\ i, j = 1, \ldots, n$.

Označíme-li vektor pravděpodobností pro každý vrchol, ze kterého bude náhodná procházka startovat, $\vec{\mathbf{s}}_0$ a $\vec{\mathbf{s}}_k$ jako vektor rozložení pravděpodobnosti, s níž bude náhodná procházka procházet danými vrcholy po k krocích z počátků $\vec{\mathbf{s}}_0$, pak se dá pravidlo přechodu vyjádřit takto [55]:

$$\vec{\mathbf{s}}_{k+1} = \mathbf{P}^T \vec{\mathbf{s}}_k, \qquad \text{respektive} \qquad \vec{\mathbf{s}}_k = \left(\mathbf{P}^T\right)^k \vec{\mathbf{s}}_0. \tag{3.9}$$

Různými výpočty, v nichž figurují maticové a vektorové operace, se dá stanovit například střední doba návštěvy konkrétního uzlu, střední doba přechodu z daného uzlu, průměrný počet návštěv průchozího vrcholu a podobně. Kvůli přítomnosti matice přechodů je třeba znát kompletní graf – všechny jeho uzly

a hrany. Navíc takový graf musí být souvislý neorientovaný, s jednou komponentou souvislosti, neboť je třeba, aby jemu odpovídající Markovův řetězec byl *nerozložitelný* (též *ireducibilní* či *regulární*). Znamená to, že každý jeho stav (vrchol) je dosažitelný z kteréhokoliv jiného stavu (vrcholu) [46].

Podmínku ireducibility není možné zajistit u orientovaných grafů ani u stromů. Dá se však v případě orientovaného stromu, jak je tomu u propojených dat, nasadit algoritmus náhodné procházky s opakovanými restarty ze stejného počátečního vrcholu. Nicméně, takovýto nesystematický přístup procházení/prohledávání grafové struktury je v tomto případě nevyužitelný. Vzhledem k předpokládané velké rozsáhlosti grafu propojených dat bychom zbytečně mnohokrát navštěvovali uzly blízké počátečnímu a naopak od vzdálených vrcholů, které nebyly navštíveny vůbec, by se nám nedostalo informace žádné.

3.2.10 Shrnutí a diskuse možností uplatnění algoritmů

Ne každý výše analyzovaný algoritmus je vhodný pro použití na grafu propojených dat k procházení jeho struktury a hodnocení (doporučování) vrcholů. U všech algoritmů, které nevyhovují požadavkům nebo je jejich princip fungování neaplikovatelný na tento problém, je uveden důvod, proč tomu tak je. Přijatelnost následného použití v implementované aplikaci, která přímo z těchto důvodů vychází, se v kostce dá zhodnotit porovnáním analyzovaných algoritmů. Srovnání s vysvětlením je provedeno slovním ohodnocením vhodnosti algoritmu: VHODNÝ a NEVHODNÝ.

Union colour algorithm Jednoduchý algoritmus prohledávání do šířky s hodnocením procházených uzlů hodnotou počátečního vrcholu. Je snadno aplikovatelný na graf propojených dat, jehož celková struktura není známa a jehož vrcholy jsou objevovány postupně. VHODNÝ.

Energy spreading algorithm Algoritmus taktéž principiálně využívající prohledávání do šířky s jemnější metodou ohodnocování vrcholů. I v tomto případě je lehce nasaditelný na graf propojených dat s postupným objevováním uzlů jeho struktury. VHODNÝ.

Spreading activation Zástupce algoritmu postaveného na prohledávání do hloubky s postupným úbytkem hodnoty relevance s možností určení spodní meze na tuto hodnotu. Jde o hojně využívaný algoritmus v sémantickém vyhledávání, tudíž na grafu propojených dat ideálně upotřebitelný. VHODNÝ.

Upravený Dijkstrův algoritmus Optimalizační algoritmus pro hledání cest v grafech založený na prohledávání do šířky. Reflektuje vztahy mezi vrcholy tím, že pracuje s ohodnocením hran mezi nimi. Je možno jej nasadit na neznámý graf s postupným odhalováním vrcholů jeho struktury. VHODNÝ.

Bellmanův–Fordův–Mooreův algoritmus Algoritmus pro hledání optimálních cest, který ke svému běhu vyžaduje kompletní seznam vrcholů i hran (s jejich ohodnocením) grafu. Na neznámém grafu by bylo nutné spustit nejprve nějaký prohledávací algoritmus. Z tohoto důvodu narůstá složitost, a tedy je pro tento problém NEVHODNÝ.

Floydův–Warshallův algoritmus Opět zástupce algoritmů pro hledání optimální cesty v grafu. Vyžaduje taktéž kompletní graf s informacemi o vzdálenostech. A protože sám o sobě nestojí na prohledávání, je nutné tuto úlohu provést samostatně předem. NEVHODNÝ.

A* algoritmus Vylepšený Dijkstrův algoritmus o heuristiku, který ovšem vyhledá optimální cestu mezi zvoleným počátečním a koncovým vrcholem v grafu. Jelikož na předem neznámém grafu nelze identifikovat koncový uzel, nedá se tento přístup vhodně aplikovat na graf propojených dat. NEVHODNÝ.

Náhodné procházení Využívá náhody pro volbu uzlů pro následující krok průchodu. Nejedná se tudíž o systematickou strategii procházení grafové struktury, která je pro graf propojených dat vyžadována, aby nedocházelo ke ztrátě informace o sousedních uzlech. Je zde však možnost upravit tento algoritmus pro procházení stromové struktury s náhodnými restarty ze stejného počátečního vrcholu. A jelikož graf nemusí být znám celý (stochasticky se prohledává), je tento přístup na graf propojených dat spíše NEVHODNÝ.

3.3 Doplňkový algoritmus kolaborativního doporučování

I když je obsah této práce zaměřen výhradně na grafové algoritmy pro doporučování, není od věci dotknout se i klasických doporučovacích technik, které s přístupy k hodnocení relevance založené na struktuře (vzájemných vztazích) položek přímo pracují, a využívají je zejména při řešení problému studeného startu. Nejlépe se dá tato symbióza mezi oběma metodami ukázat na běžně užívaném kolaborativním doporučování.

V běžných aplikacích a systémech s uživateli jde o nejpoužívanější přístup vůbec. Jeho úkolem je predikovat hodnocení položky u aktivního uživatele, který ji dosud neohodnotil, pomocí ostatních uživatelů, kteří již k této položce hodnocení poskytli a kteří vykazují velmi podobné zaujetí společně hodnocenými položkami. Technika předpovědi hodnocení vyžaduje databázi uživatelů s hodnotami jejich preferencí, jenž přiřadili příslušným položkám, proto se o ní hovoří jako o technice *založené na paměti* (*memory-based technique*).

Doporučovací algoritmus se skládá z několika vzájemně se využívajících částí (převzato a upraveno z [61]):

1. Výpočtu váhy mezi dvěma uživateli, která odráží jejich vzájemnou podobnost.

2. Určení možného ohodnocení na základě predikce podle podobností (vah) uživatelů.

3. Výběru nejlepších doporučení.

3.3.0.1 Výpočet podobnosti

Budeme-li uvažovat dva různé uživatele u a v, každého s množinou položek, které ohodnotil, tedy I_u a I_v, pak pro tyto uživatele chceme zjistit podobnostní váhu $w_{u,v}$ podle hodnocení společných položek $I = I_u \cap I_v$ [61].

Kosinová podobnost Základní vektorovou podobnost nám může poskytnout kosinová míra, která udává velikost úhlu, jenž mezi sebou svírají dva vektory. Nechť tedy $\vec{\mathbf{r}}_u$ je vektor hodnocených položek z množiny I uživatele u a $\vec{\mathbf{r}}_v$ vektor hodnocených položek z množiny I uživatele v, potom se váha $w_{u,v}$ vypočte dle vztahu

$$w_{u,v} = \cos(\varphi_{\vec{\mathbf{r}}_u,\vec{\mathbf{r}}_v}) = \frac{\vec{\mathbf{r}}_u \cdot \vec{\mathbf{r}}_v}{\|\vec{\mathbf{r}}_u\| \cdot \|\vec{\mathbf{r}}_v\|}, \tag{3.10}$$

kde $\|\cdot\|$ značí eukleidovskou normu vektoru: $\|\vec{\mathbf{x}}\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$.

Pearsonova korelace Vztah mezi dvěma uživateli jde vyjádřit i statistickou mírou, kterou nám poskytuje Pearsonova korelace. Nechť u a v jsou dva uživatelé, $r_{x,i}$ je ohodnocení položky $i \in I$ uživatele x a $\bar{r}_x$ je aritmetický průměr hodnot všech společných ohodnocených položek z I uživatele x, potom se váha $w_{u,v}$ vypočte dle vztahu

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}}. \tag{3.11}$$

3.3.0.2 Predikce ohodnocení položky

Úkolem předpovědi hodnocení položky i aktivního uživatele a je stanovit hodnotu $r'_{a,i}$ [61].

Vážený aritmetický průměr odchylek Zde se využívá výpočtu váženého průměru odchylek hodnocení od aritmetického průměru hodnocení všech uživatelů z množiny N, která může obsahovat všechny uživatele z množiny U vyjma aktivního a, tedy $N = U \setminus \{a\}$, nebo je složena z prvních n uživatelů s nejvyššími podobnostními váhami $w_{a,u}$:

$$r'_{a,i} = \bar{r}_a + \frac{\sum\limits_{u \in N} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}}{\sum\limits_{u \in N} |w_{a,u}|}. \tag{3.12}$$

Kapitola 4

Návrh a implementace aplikace

Výchozím bodem pro vznik diplomové práce na téma grafových algoritmů pro doporučování v propojených datech se stala myšlenka vytvoření aplikace, která by uživateli poskytla možnost procházet propojená data určité oblasti jeho zájmu (například filmy, hudbu, ...) z předem určeného výchozího bodu. Postupným průchodem daty mu budou nabízeny výsledky v pořadí dle relevance vycházející z předchozího ohodnocení důležitosti dílčích startovních uzlů v jednotlivých krocích postupu strukturou grafu.

4.1 Použité nástroje, pomůcky a technologie

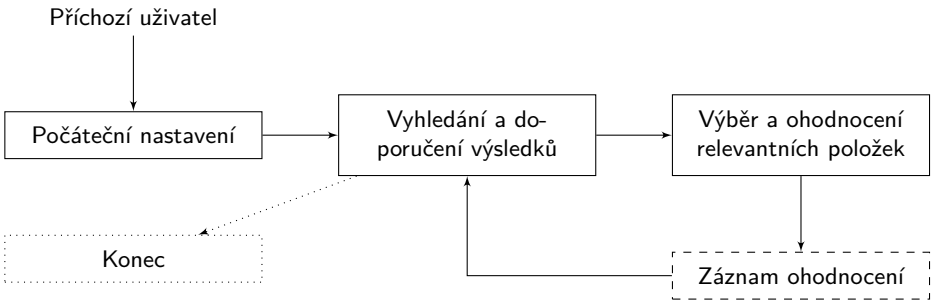
Prostředí sémantického webu a propojených dat zcela jednoznačně implikuje, že webová aplikace pro procházení online dat v reálném čase je tou správnou volbou. Dokladem nechť jsou i diskutované doporučovací systémy v kapitole 2.4, všechny koncipované jako webové aplikace.

Přehled použitých nástrojů včetně vývojových prostředí a doplňků:

- Webová aplikace postavená na
  - jazyce Java 8 u91 (build 1.8.0_91-b14),
  - frameworku Spring MVC 4.2.3,
- integrované vývojové prostředí Netbeans 8.0.2,
- nasazeno na webový server Glassfish 4.1.1,
- další závislosti Maven Project Object Model 4.0.0,
  - Apache Jena 3.0,
  - SQLite 3.8.11.2.

4.2 Návrh

Scénář pohybu uživatele jednotlivými kroky aplikace a odpovídající reakce na jím učiněné akce by měl zároveň obsahovat určitý návod, jehož body přehledně demonstruje diagram 4.1.



Obrázek 4.1: Možné kroky uživatele a odezva na jeho akce v aplikaci

1. Nejprve příchozí uživatel nastaví počáteční parametry pro vyhledávání a doporučování, a sice
   - druh algoritmu pro procházení grafu a doporučování,
   - počáteční uzel, tedy zdroj, a to pomocí přímého IRI,
   - omezení IRI odkazů v rámci zvolené domény,
   - omezení typu vyhledávaných položek (například dle ontologie) taktéž pomocí IRI,
   - metodu přístupu k datům,

   navíc doplňkově
   - identifikátor uživatele pro záznam hodnocených entit.
2. Následně se provede vyhledání položek ve struktuře grafu a jejich ohodnocení dle zvoleného algoritmu a předchozího ohodnocení.
3. Uživatel má následně možnost výběru relevantních položek z výsledných IRI přiřazením hodnoty důležitosti k těmto.
4. O jeho hodnocení se provede záznam a pokračuje se bodem 2.

4.2.1 Procházení grafové struktury

Výchozí zdroj (IRI) zadaný uživatelem na začátku průchodu musí svým formátem odpovídat uvedenému omezení, které zabezpečí prohledávání grafu výhradně v jedné doméně. Z popisu zdroje je třeba vyselektovat pouze taková IRI, která vyhovují tomuto omezení. Jelikož se jedná o prostý text, je zde možnost porovnávat omezení s vybíranými IRI řetězcovými metodami (například,

obsahuje-li IRI definovaný podřetězec). Ostatní uzly RDF grafu, jako literály, prázdné uzly a zdroje s IRI neodpovídajícími omezení, nechť jsou ignorovány a nezahrnují se do dalšího zpracování.

Algoritmy pro procházení a doporučování by měly umožňovat zpracování grafu z několika počátečních vrcholů ohodnocených uživatelovou preferencí zároveň. Ačkoliv selekce zdrojových IRI dle omezení částečně zredukuje množství zpracovávaných uzlů, je i přesto třeba pamatovat na rozsáhlost propojených dat, a vyjít vstříc eventuálnímu souběžnému zpracování, kupříkladu pomocí vláken.

4.2.2 Zpracování a prezentace výsledků

Zpracováním části grafu propojených dat z jednoho či více počátečních vrcholů získáme množinu výsledných IRI, každý odkaz ohodnocený relevancí. Opět z důvodu velkého množství dat a položek, jejichž reprezentace neodpovídá oblasti, na níž je doporučování zaměřeno, je vhodné výsledky pročistit aplikací filtru, který vybere pouze taková IRI, jež spadají do třídy určené omezením typu (ontologie). V dalším lze redukovat počet výsledných prezentovaných položek výběrem prvních n dosahujících nejlepší hodnoty relevance.

4.3 Implementace

Webová aplikace s názvem *Graph Based Recommendation Algorithms for Linked Data*, zkráceně GBRAFLD, je složena z několika skupin tříd určených k řešení konkrétních úkolů. Jedná se především o

- oblast konfigurace, konstant, podpůrných a doplňkových tříd,
- aplikační logiku
  - realizace architektury MVC,
  - přístup k síti, komunikace,
  - zpracování RDF dat,
  - algoritmy pro procházení grafových struktur a doporučování,
- přidanou nadstavbovou funkcionalitu a
- testovací třídy.

4.3.1 Konfigurace

Třídy obsahující konfigurační logiku, konstanty a další nastavení jsou pro běh aplikace nezbytné. Nejenže jde většinou o neinstanciovatelné třídy se statickými daty, jako například definice adresářových cest, dovolené akce a jména používaných atributů, ale jsou zde také zaváděcí třídy frameworku.

4.3.2 Architektura Model-View-Controller

Jádro aplikace využívá frameworku Spring MVC, který přímo realizuje návrhový vzor Model-View-Controller za účelem vhodného oddělení aplikační logiky od části prezentační.

4.3.2.1 Model

Data obsažená v *modelu* se využívají pro prezentaci, pracuje se zde také s databází či se souborovým systémem. Třídy zahrnuté do skupiny „model“ obsahují data získávaná a předávaná uživateli jako vstup/výstup, názvy formulářových prvků zobrazovaných na webové stránce a jejich hodnoty. Jsou tu i třídy pro zapouzdření dat předávaných mezi objekty. Logicky sem patří i část s přístupem k RDF metadatům a jejich zpracováním, nicméně v aplikaci je vyčleněna jako samostatná součást.

4.3.2.2 View

Oblast *wiev* zahrnuje především šablony webových stránek umožňující zobrazování dynamických dat. O správný výběr a transformaci do HTML se stará vnitřní nástroj frameworku Spring MVC spolu s technologiemi JavaServer Pages (JSP) a značkovacím systémem JavaServer Pages Standard Tag Library (JSTL).

4.3.2.3 Controller

Skupina kontrolérů působí coby hlavní řídicí jednotka aplikace GBRAFLD. V ní se vyskytují tři základní kontroléry (neuvažujeme-li servlet pro mapování požadavků frameworku Spring MVC):

1. hlavní přístupový bod do aplikace zpracovávající vstupní požadavky uživatele,

2. řadič požadavků pro procházení propojených dat a

3. kontrolér pro chybová hlášení.

4.3.3 Přidaná funkcionalita

Grafové doporučování je navíc obohaceno o klasickou doporučovací techniku kolaborativního filtrování. Jde o nadstavbu výpočtů relevance pro zdrojová IRI v grafu propojených dat, která zahrnuje do procesu doporučování uživatele aplikace. Pro zajištění bezpečnosti původního kódu je tento doplněk umístěn v balíku testovacích tříd, jeho funkcionalita tudíž není dostupná klientovi přímo.

4.4 Klientská část

Uživatel se k aplikaci připojuje přes běžný webový prohlížeč. Je zapotřebí, aby byl klient schopen pracovat se standardními HTTP metodami a zobrazovat přijatá data v HTML5, navíc je také nutné, aby klient na své straně podporoval spouštění JavaScriptu. To z toho důvodu, že po odeslání požadavku může výpočet na serveru trvat delší dobu, a data tak nejsou k dispozici neprodleně.

V každém kroku, kdy uživatel učiní požadavek na výpočet (to jest žádá server o prohledání grafu propojených dat spolu s výpočtem relevance jednotlivých uzlů), je přesměrován na stránku s výsledky, kde se na pozadí spustí javascriptový kód pro *Ajax Polling* (využívá se frameworku jQuery 1.12.0). Klient se v pravidelných intervalech po 0,5 s dotazuje serveru HTTP metodou GET, jestli již výpočet skončil a zda už jsou pro něho k dispozici data. Server pak odpovídá buď kódem

202, tedy že požadavek dosud nebyl zpracován (Accepted), tudíž polling poběží i nadále,

200, tedy OK, s daty ve formátu JSON určenými pro zobrazení na webové stránce, nebo

500 v případě jakékoliv chyby vzniklé na serveru.

4.5 Serverová část

4.5.1 Kontroléry

Dříve již bylo zmíněno, že v aplikaci operuje trojice kontrolérů. Jejich třídy jsou anotovány značkou `@Controller` z frameworku Spring MVC.

4.5.1.1 Hlavní kontrolér

Třída `IndexController` má za úkol shromáždit od uživatele data týkající se počátečního nastavení aplikace:

- druh algoritmu,
- identifikátor uživatele,
- výchozí zdroj (IRI resource),
- SPARQL endpoint (má-li se použít),
- omezení na formát zpracovávaných IRI,
- omezení na typ zpracovávaných zdrojů (IRI ontologie).

Pakliže jsou data v pořádku a platná, poskytnou se pomocí session třídě `TraversalController`. Dojde-li během operací hlavního kontroléru k jakékoli chybě, hlášení o ní se skrze `ErrorController` dostane až k uživateli.

4.5.1.2 Kontrolér požadavků na procházení propojených dat

Tento řadič, třída `TraversalController`, má za úkol zpracovat data s nastavením od hlavního kontroléru, jakož i data, která obdrží coby zpětnou vazbu od uživatele během průchodu propojenými daty. Uživatel předává kontroléru v jednotlivých krocích informace o položkách, které ho zaujaly, tím, že jim přiřadí hodnotu důležitosti, jakou jim přisuzuje na škále od 0 do 1. Platné informace o preferencích uživatele se předávají dále třídě doporučovače ve formě mapy: IRI (řetězec) $\mapsto$ preference (reálné číslo). Zahrnuta jsou pouze taková mapování, kde je hodnota preference ostře větší než nula.

Druhá úloha řadiče `TraversalController` spočívá v implementaci HTTP metody GET pro RESTové rozhraní, pomocí něhož předává data s výsledky průchodu propojených dat ve formátu JSON klientovi do webové stránky (klient si o ně říká Ajax Pollingem).

4.5.1.3 Kontrolér chyb

Posledním z trojice řadičů je `ErrorController`, jehož jediným úkolem je předávání informací klientovi o nastalých chybách v aplikaci.

4.5.2 Doporučování

Celý proces, počínaje získáním RDF popisů zdrojů a výběrem odkazů na další zdroje z jejich struktury, přes výpočet hodnoty jejich relevance, a navrácením a filtrováním výsledných dat konče, zajišťuje a řídí třída `Recommender`.

Podle informací z nastavení běhu aplikace, které určuje uživatel a které se doporučovači předají z kontroléru požadavků na procházení propojených dat, se vybere algoritmus, jímž se bude průchod grafem a doporučování realizovat. Získané výsledky se následně redukují, vybírají se pouze takové, které odpovídají vstupním omezením na typ, řadí se dle relevance, případně se výstup omezuje pouze na n prvních nejlépe hodnocených.

4.5.2.1 Doporučovací algoritmy

Každá třída zastupující jeden grafový algoritmus implementuje společnou strategii průchodu (návrhový vzor *Strategy*). Důležitým parametrem je seznam výchozích uzlů s hodnotami důležitosti (> 0), z nichž prohledávání započne, a omezení na tvar IRI vybíraných z RDF popisů procházených zdrojů.

Uživatel si v aplikaci před začátkem průchodu zvolí jeden algoritmus (názvy algoritmů odpovídají názvům tříd v aplikaci):

- `Simple link collector` je testovací algoritmus, který zobrazí pouze bezprostřední sousedy zvoleného výchozího vrcholu v grafu propojených dat. Neaplikuje během procházení žádný výpočet relevance ani žádné hodnoty od uživatele nepřijímá, všechny výsledné odkazy na zdroje mají relevanci rovnu nule.
- `Union colours`, `Energy spreading`, `Modified Dijkstra`, `Spreading activation`.

Jednotlivé třídy si samy řeší souběžné zpracování grafu z několika počátečních vrcholů zároveň, pokud to algoritmus podporuje. Souběh realizují vlákna z *thread poolu* řízená exekutory. Výsledky se pak vrací třídou `Future` dovolující dodat data klientovi ve chvíli, kdy jsou kompletně k dispozici. Hodnoty vypočtené relevance jsou přímé výstupy algoritmů, pro jejich další zpracování bude zřejmě třeba je vhodně normalizovat do intervalu $\langle 0; 1\rangle$.

Union colours Prohledává do šířky a hodnotí všechny sousedy relevancí počátečního uzlu. Dovoluje souběžné zpracovávání z více výchozích IRI. Respektuje smyčky v grafu. Omezujícím parametrem je maximální hloubka zanoření do grafu.

Energy spreading Rozprostírá energii rovnoměrně do všech sousedů (do šířky). Dovoluje postupné zpracovávání z více výchozích IRI. Respektuje smyčky v grafu. Omezujícím parametrem je maximální hloubka zanoření do grafu.

Modified Dijkstra Hledá nejdelší cestu ve stromové struktuře grafu průchodem do šířky. Všechny hrany mají hodnotu rovnu jedné. Dovoluje souběžné zpracovávání z více výchozích IRI. Respektuje smyčky v grafu. Omezujícím parametrem je maximální hloubka zanoření do grafu.

Spreading activation Prohledává do hloubky a hodnotí postupně uzly dle relevance předchozího uzlu násobené váhovým (hodnota 1) a tlumicím (hodnota 0,5) koeficientem. Dovoluje souběžné zpracovávání z více výchozích IRI. Ignoruje smyčky v grafu. Omezujícím parametrem je maximální hloubka zanoření, která se promítá do prahové funkce, jejíž hodnota řídí hloubkový průchod grafem. Prahová funkce f_T má tvar

$$f_T(r_S, h, w, d) = r_S \cdot (w \cdot d)^h \,, \tag{4.1}$$

kde $r_S \in (0; 1\rangle$ je relevance výchozího vrcholu, $h \in \mathbb{N}$ maximální hloubka zanoření, w je váhový a d tlumicí koeficient, oba z intervalu $\langle 0; 1\rangle$. Přípustná

pro zpracování je výhradně aktivační hodnota uzlu ostře větší než práh a $(w = d) \neq 1$. Zpracováním se rozumí vyhledání sousedních vrcholů.

4.5.2.2 Zpracování RDF dat

Průchod grafem propojených dat je v podstatě přecházením po IRI, odkazech na zdroje, vyňatých z jejich RDF popisů. Aplikace nabízí dvě možnosti, jak získat RDF popis konkrétního zdroje.

1. Klasickou dereferencí URI přes HTTP. Na zdroj je odeslán dotaz s hlavičkou `Accept`, kde je specifikována vyžadovaná serializace RDF metadat. V hlavičce odpovědi, parametru `Location`, následně obdržíme adresu umístění souboru s daty. Třebaže je deklarována podpora pro všechny typy formátů, preferovány jsou zejména
   - `application/rdf+xml`,
   - `application/trig`,
   - `application/n-quads` a
   - `text/turtle`.
2. Popisem zdroje pomocí SPARQL dotazu. Na existující koncový bod pro dotazování pomocí SPARQL je odeslán dotaz na popis zdroje, odpovědí jsou (v případě korektního zdroje) RDF data v určité serializaci. SPARQL dotaz má tvar `DESCRIBE <`*IRI_zdroje*`>`.

Surová RDF data zpracovává třída `Recommender` a každá třída implementující nějaký algoritmus pro průchod grafem a doporučování. Za tímto účelem se využívá frameworku Apache Jena pro efektivní manipulaci s RDF datovým modelem v operační paměti nezávislým na vstupní serializaci.

Dotazy je možno nad modelem volat programově; děje se tak především kvůli nutnosti selekce takových IRI, jejichž subjekt nebo objekt odpovídá současnému startovnímu IRI, a formát vyhovuje omezení tvaru IRI – procházíme-li například zdroje na DBpedii, chceme, aby všechna IRI (jakožto řetězce znaků) začínala `http://dbpedia.org/resource/`. K tomu slouží třídy, jež spadají do skupiny selektorů, tyto rozšiřují svou implementací třídy z modelu Apache Jena.

Filtrování je nutné provádět ve dvou nezávislých krocích. Selekce zdrojových linků z RDF dat probíhá již během zpracování souboru s RDF popisem daného zdroje, aby eventuální pohyb grafem zůstával v oblasti zdrojů definované omezením na formát IRI. Naopak výběr takových zdrojů, které odpovídají svým typem třídě určené typovým omezením, je nutno provádět až ve chvíli, kdy jsou získány všechny uzly procházené části grafu (pochopitelně redukované

omezením na formát IRI). Čili pouze tehdy, jestliže nasazený algoritmus již doběhl, vzhledem ke skutečnosti, jak je vystavěna kostra těchto algoritmů. To má bohužel velmi často neblahý dopad na dobu běhu, je-li dat příliš mnoho, neboť není možné zajistit dotazování se na každý zdroj právě jednou, ideálně pouze v době jeho objevení a prvotního zpracování.

4.6 Popis uživatelského rozhraní a ovládání

O prezentaci výsledných doporučených položek s hodnotami jejich relevance a o interakci uživatele s aplikací se stará jednoduché webové rozhraní. Klient se v průběhu jeho užívání pohybuje mezi třemi stránkami – třemi typy obrazovek:

1. hlavní stránkou,
2. stránkou pro průchod propojenými daty a
3. chybovou stránkou.

4.6.1 Hlavní strana

Na úvodní stránce generované z šablony `index.jsp` je umístěn vstupní formulář určený pro prvotní nastavení aplikace, kterého se bude využívat po celou dobu průchodu aktivního uživatele, dokud se nevrátí zpět na úvodní stránku, nebo dokud aplikaci neukončí (ukončení session).

Následující popis se týká obrázku 4.2 ilustrujícího úvodní obrazovku aplikace Graph Based Recommendation Algorithms for Linked Data.

1. Volba algoritmu.
2. Identifikace uživatele.
3. Počáteční uzel (IRI) grafu.
4. Volba SPARQL endpointu. Není-li zaškrtnuto, použije se HTTP dereference IRI.
5. Omezení kladená na
   a) tvar IRI (běžně podle názvu domény),
   b) typ IRI (definuje jej odkaz na ontologii).

4.6.2 Strana pro průchod propojenými daty

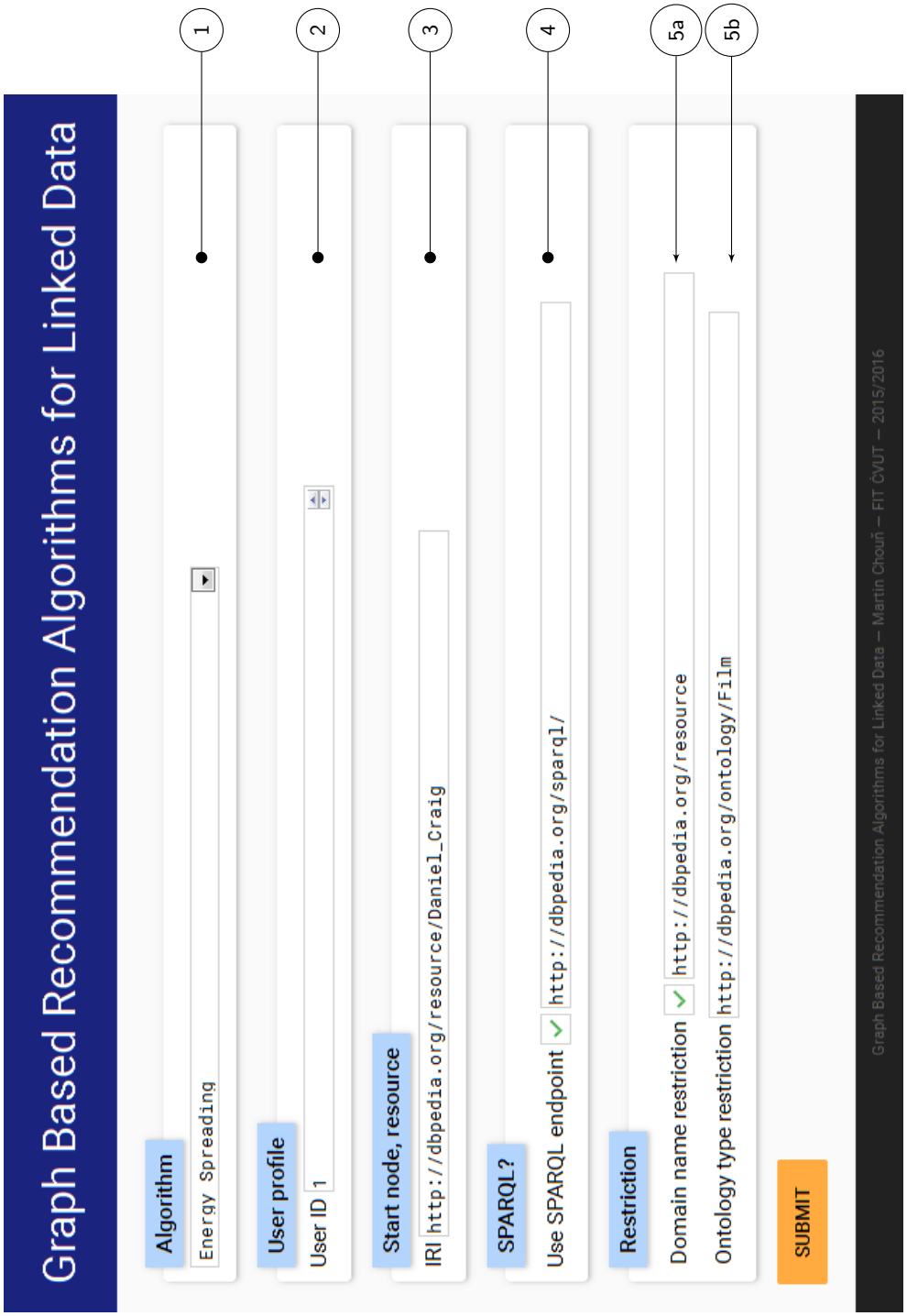
Výsledky jednotlivých kroků v postupu grafem propojených dat, totiž odkazy s vypočtenou hodnotou důležitosti s možností ohodnotit je dle uživatelova zájmu, předkládá strana transformovaná z šablony `traversal.jsp`.

Číslovaný seznam níže odpovídá popisu obrázku 4.3, kde je vyobrazena strana pro průchod propojenými daty. Tatáž strana je obsahem i obrázku 4.4, avšak zde je navíc ilustrován běh javascriptového Ajax Pollingu na pozadí v doplňku Firebug 2.0.16 webového prohlížeče Mozilla Firefox 45.0.2.

1. Identifikátor aktivního uživatele a odkaz na hlavní stranu.
2. Seznam právě procházených IRI.
3. Tlačítko pro vykonání následujícího kroku průchodu.
4. Nalezené sousední odkazy v grafové struktuře propojených dat.
5. Vypočítaná relevance, dle níž jsou nalezená IRI seřazena sestupně.
6. Vstupní pole pro zadání hodnoty, s jakou důležitostí uživatel přistupuje k výslednému odkazu.

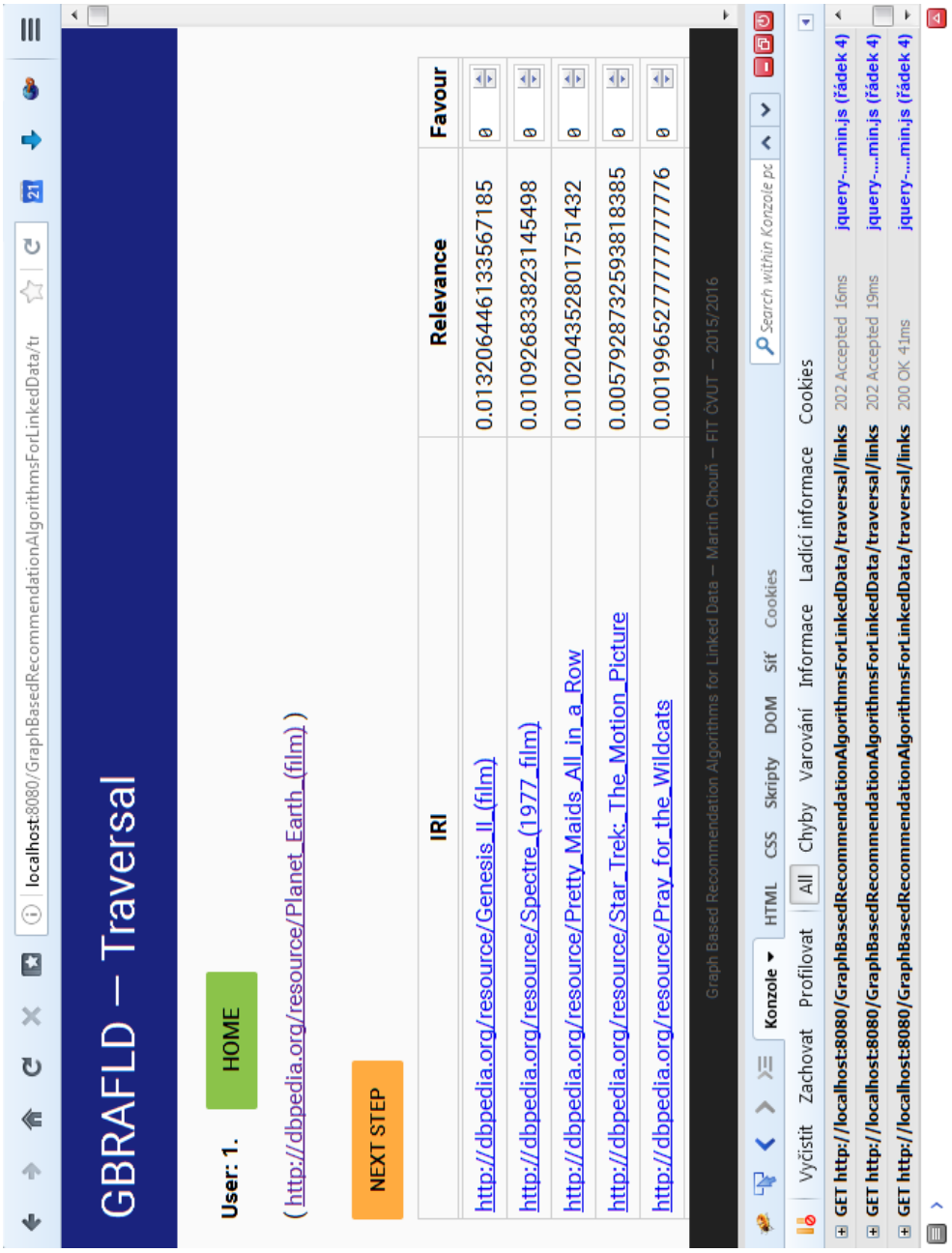
4.6.3 Chybová stránka

Chybová obrazovka pouze obsahuje hlášení, o jakou chybu se jedná a případně v jakém místě aplikace nastala. Stránku je možno opustit pouze přechodem na hlavní stranu.



Obrázek 4.2: Hlavní strana aplikace GBRAFLD

Obrázek 4.3: Strana pro průchod propojenými daty aplikace GBRAFLD



Obrázek 4.4: Strana pro průchod propojenými daty – Ajax Polling

Kapitola 5

Experimenty

Abychom získali přehled, jak ve skutečnosti vypadá struktura propojených dat na konkrétních datových sadách a jak se na takové struktuře chovají implementované algoritmy, je třeba přistoupit k pozorováním jejich běhu na různých datech a pokusům spočívajícím ve změnách parametrů těchto algoritmů. Veškeré vlastnosti a pozorování společně s naměřenými hodnotami jsou diskutovány v každé oblasti experimentů.

5.1 Cíle experimentů

Hlavní úlohou experimentů je prozkoumat rozlehlost grafu propojených dat z konkrétní oblasti (domény) a zmapovat chování jednotlivých algoritmů na tomto grafu, a sice z různých počátečních IRI na dvou zvolených datových sadách a v závislosti na nastavení jejich parametrů.

5.2 Zdroje dat

I když je aplikace koncipována jako systém pro doporučování obecných entit v propojených datech, zavedená omezení dovolují zúžit výběr na konkrétní sféry zájmu uživatelů. V dalším tedy budou uvažovány experimenty na datech spadajících do oblasti filmů, především původem z existujících datasetů, jimiž jsou DBpedie a LinkedMDB.

5.2.1 DBpedia

Projekt DBpedia (`http://dbpedia.org/`) zpřístupňuje na webu strukturovaná sémantická data získaná z Wikipedia a Wikidat, k nimž je možno přistupovat přímou dereferencí HTTP URI či přes SPARQL endpoint `http://dbpedia.org/sparql`.

5.2.2 LinkedMDB

LinkedMDB (`http://data.linkedmdb.org/`) publikuje otevřená data výhradně z oblasti filmů propojená do LOD cloudu. Bohužel jejich získávání se v současné době (duben 2016) uskutečňuje jen přes přímou dereferenci HTTP URI, byť na stránkách projektu je deklarována přístupnost také přes SPARQL koncový bod. Spolehlivost přístupu na server LinkedMDB je ovšem poznamenána jeho častými výpadky a nedostupností, jakož i nefunkčností některých odkazů.

5.3 Experiment: Struktura a rozsáhlost dat

Rozsáhlé databáze a datové sady propojených dat představují hustou spleť odkazů tvořících velmi rozlehlý graf. Průchodem jeho struktury z počátečního vrcholu po orientovaných hranách (IRI linky) je postupně odkrýván strom jedinečných uzlů (zanedbáme-li existující kružnice a zpětné hrany). V takovém stromě může s přibývajícími patry enormně narůstat počet objevených vrcholů.

Tabulky 5.1 a 5.2 odhalují rozsáhlost grafu z různých výchozích uzlů a na různých datových sadách. V každé jsou sledovány následující parametry:

Hloubka zanoření do grafu;

Selekce čili způsob výběru sousedních vrcholů:

A = všechny sousední vrcholy, jež jsou zdroji (IRI),

IR = všechny sousední vrcholy, které odpovídají omezení na tvar IRI,

ITR = všechny sousední vrcholy, které odpovídají omezení na tvar IRI a které zároveň splňují typové omezení;

n značí počet získaných jedinečných entit (vrcholů) prohledáváním do šířky;

t je doba běhu prohledávání do šířky.

5.3.1 Nastavení experimentu

Pro experimenty na datových sadách DBpedie a LinkedMDB byly zvoleny podobné typy počátečních IRI, z nichž startovalo prohledávání do šířky (bez jakéhokoliv hodnocení uzlů). Protože není možné triviálně vyhledat metadata, která popisují téže entity vyskytující se v obou datových sadách zároveň, je dbáno alespoň na zachování stejných typů popisovaných položek, jejichž IRI slouží jako výchozí uzly. Vybrána jsou vždy IRI označující herce, film a kategorii (žánr) filmů, od každého typu po čtyřech různých.

V prvním sloupci tabulky 5.1 s počátečními IRI se předpokládá u všech názvů prefix `dbr:`, který je zkratkou pro `http://dbpedia.org/resource/`. Skutečný

tvar IRI pro první řádek má tvar `http://dbpedia.org/resource/Daniel_Craig`. Obdobně i řádky zbývající. Omezení na tvar IRI odpovídá témuž řetězci, proto všechna IRI, na které je aplikováno, musejí obsahovat jako svůj podřetězec `http://dbpedia.org/resource/`. Typové omezení na oblast filmů má tvar `http://dbpedia.org/ontology/Film`. Data jsou získávána SPARQL dotazem z DBpedie a procházena obyčejným prohledáváním do šířky (v aplikaci algoritmus Simple link collector).

Sloupec s počátečními IRI v tabulce 5.2 je taktéž prefixován, a sice řetězcem `lmdbr:` zkracujícím `http://data.linkedmdb.org/resource/`. Skutečný odkaz tudíž vypadá takto: `http://data.linkedmdb.org/resource/actor/1`. Omezení na tvar IRI: `http://data.linkedmdb.org/resource` (podřetězec); typové omezení: `http://data.linkedmdb.org/resource/movie/film` (film). Prohledávání dat LinkedMDB získaných dereferencí URI je do šířky (v aplikaci algoritmus Simple link collector).

5.3.2 Výsledky experimentu

Experiment pro odhalení rozsáhlosti části grafu filmových dat na DBpedii, který shrnuje tabulka 5.1, ukazuje, že v první úrovni prohledávání byly u většiny počátečních IRI nalezeny nějaké sousední vrcholy se zdrojovými IRI. Omezením se výhradně na trojice, které popisují výchozí vrchol dalším IRI splňujícím podmínku jeho tvaru (doménové jméno apod.), se sníží počet vyhledaných použitelných vrcholů grafu přibližně na polovinu původního počtu. V případě filtrování pouze takových sousedních IRI, které popisují film, již získáme i množství nulových hodnot, tedy prázdné množiny sousedních uzlů. Doba běhu prohledávání do šířky se pohybuje řádově nejvýše v jednotkách sekund.

Prohledávání do druhé úrovně grafu již přináší větší množství uzlů vhodných pro zpracování, a to i po aplikaci zmiňovaných omezení. Počty uzlů ve většině případů překračují jednotky tisíců, v několika případech i stovky tisíců. Po výběru filmových vrcholů zbývají řádově stovky použitelných uzlů, zatímco rapidně narůstá doba běhu prohledávání do šířky. Přestože je k datům přistupováno přes SPARQL endpoint, může se čas zpracování takového množství vyšplhat až na desítky minut.

Množství informací získaných z metadat LinkedMDB (tabulka 5.2) se nedá srovnávat s předchozím měřením na DBpedii, neboť jde o menší datovou sadu. I tak je ale úbytek počtu sousedních uzlů s IRI v první úrovni po aplikaci omezení na tvar IRI podobně velký – přibližně 50%. Rozdíl je ale patrný v době zpracování. Z důvodu přístupu k datům pomocí dereference URI se doba běhu prohledávání do šířky v některých případech vyšplhala až k hodnotám jednotek hodin.

Tabulka 5.1: Rozsáhlost grafu na datech z DBpedie

Hloubka	**1**						**2**					
Selekce	**A**		**IR**		**ITR**		**A**		**IR**		**ITR**	
Počáteční IRI (dbr:)	**n (–)**	**t (s)**	**n (–)**	**t (s)**	**n (–)**	**t (s)**	**n (–)**	**t (s)**	**n (–)**	**t (s)**	**n (–)**	**t (s)**
Daniel_Craig	130	6,223	62	4,310	24	10,023	107202	103,946	9024	25,207	64	746,817
Natalie_Portman	212	0,225	123	0,218	32	10,672	130869	67,111	31029	27,673	91	1485,660
Karel_Roden	80	0,147	29	0,119	14	2,642	98193	25,935	1366	13,752	14	72,160
Tereza_Voříšková	59	0,165	21	0,240	2	1,675	106261	45,716	13033	11,842	2	603,915
Dr._No_(film)	94	0,264	46	0,174	0	3,567	15988	19,985	1902	7,618	1355	97,269
The_Shawshank_Redemption_(film)	4	0,215	1	0,259	1	0,174	106	0,547	45	0,216	1	1,901
Memento_(film)	92	0,210	34	0,176	0	2,741	19084	21,016	5277	22,378	4728	312,400
Planet_Earth_(film)	37	0,286	24	0,270	0	1,843	47932	13,063	1841	7,630	379	86,844
Category:Czech_films	0	0,210	0	0,271	0	0,158	0	0,177	0	0,179	0	0,135
Category:American_films	0	1,604	0	1,541	0	1,737	0	1,429	0	1,620	0	0,953
Category:Comedy	119	0,100	105	0,107	0	8,279	4481	11,916	3398	10,399	26	158,236
Category:Animated_film_series	0	0,204	0	0,220	0	0,110	0	0,087	0	0,077	0	0,146

Tabulka 5.2: Rozsáhlost grafu na datech z LinkedMDB

Hloubka	**1**						**2**					
Selekce	**A**		**IR**		**ITR**		**A**		**UR**		**ITR**	
IRI (lmdbr:)	**n (–)**	**t (s)**	**n (–)**	**t (s)**	**n (–)**	**t (s)**	**n (–)**	**t (s)**	**n (–)**	**t (s)**	**n (–)**	**t (s)**
actor/1	12	4,427	6	4,108	6	12,089	2557	22,213	2515	12,844	2515	3219,554
actor/2	6	0,395	3	0,381	3	3,962	2550	13,323	2517	4,166	2517	3248,314
actor/3	10	0,534	7	0,373	7	5,925	2642	12,704	2596	6,524	2596	3363,714
actor/4	6	0,494	3	0,405	3	3,602	2540	6,971	2508	3,818	2508	3239,687
film/1	32	0,488	24	0,433	24	20,938	2970	37,195	2793	35,461	2763	3927,210
film/2	42	0,613	37	0,434	37	46,169	3278	29,171	3198	29,047	2970	3907,661
film/3	35	0,639	30	0,399	30	23,426	4286	45,099	3937	43,19	3115	5258,998
film/4	46	0,434	38	0,429	38	37,242	3077	45,370	3007	44,828	2873	5747,729
film_genre/1	4	0,383	2	0,397	2	7,200	491	13,587	480	2,513	480	728,901
film_genre/2	8	0,435	6	0,384	6	5,306	632	13,646	593	4,611	593	758,318
film_genre/3	0	0,381	0	0,401	0	0,375	0	5,103	0	0,413	0	1,153
film_genre/4	2503	3,244	2501	1,961	2501	325,548	11717	3375,959	9607	3364,630	8816	16267,087

Experiment na LinkedMDB je bohužel poznamenán častými výkyvy dostupnosti serveru s metadaty, a proto musel být několikrát restartován. Navíc zde nebylo možné zaručit, že všechny uzly byly v době běhu pokusu dostupné a že server poskytoval relevantní a konzistentní data.

5.4 Experiment: Analýza parametrů algoritmů

V následujících tabulkách se vyskytují tyto parametry sledované u jednotlivých algoritmů.

Hloubka zanoření do grafu;

Selekce čili způsob výběru sousedních vrcholů:

IR = všechny sousední vrcholy, které odpovídají omezení na tvar IRI,

ITR = všechny sousední vrcholy, které odpovídají omezení na tvar IRI a které zároveň splňují typové omezení;

$\mathbf{I_n}$ počet výchozích IRI;

$\mathbf{r_S}$ ohodnocení výchozího IRI uživatelem (důležitost);

$\mathbf{f_V}$ četnost návštěvy vrcholu;

$\mathbf{n_V}$ počet navštívených vrcholů s četností f_V;

k koeficient průměrného nárůstu nově objevených vrcholů;

r ohodnocení vrcholu (hodnota relevance, doporučení);

$\mathbf{n_r}$ četnost vrcholů s ohodnocením r;

h prahový exponent (u spreading activation);

w aktivační váha (u spreading activation);

d tlumicí činitel (u spreading activation).

Četností f_V návštěvy vrcholu je míněna frekvence jeho výskytu v množinách sousedních uzlů při prohledávání. Koeficient k průměrného nárůstu nově objevených vrcholů je vypočten jako geometrický průměr relativních přírůstků počtu nově objevených uzlů. U algoritmu spreading activation se ještě vyskytuje prahový exponent h, jehož hodnota je totožná s hloubkou.

5.4.1 Nastavení experimentů

Pro všechny experimenty jsou voleny co možná nejpodobnější podmínky jejich běhu v testovacím režimu. Pokusy jsou prováděny na DBpedii, neboť je to stabilní datová sada se spolehlivým přístupem k metadatům zdrojů přes SPARQL endpoint. Všechna pozorování započínají buď v jediném vrcholu `http://dbpedia.org/resource/Planet_Earth_(film)` nebo ve dvou výchozích vrcholech: `http://dbpedia.org/resource/Planet_Earth_(film)` a `http://dbpedia.org/resource/I_Love_Lucy_(film)` s různými hodnotami důležitosti. Startovní IRI jsou voleny s ohledem na rozsáhlost grafů jejich RDF popisů, aby bylo možné pozorovat odchylky v naměřených hodnotách. Společným uzlem obou výchozích zdrojů je jejich režisér (`http://dbpedia.org/resource/Marc_Daniels`), čímž je zaručena provázanost obou grafů i mezi takto na první pohled odlišnými filmy.

5.4.2 Výsledky experimentů

Společným ukazatelem experimentů napříč všemi algoritmy je koeficient průměrného nárůstu nově objevených vrcholů, jehož hodnoty se pohybují přibližně od 0,048 do 0,079. Ostatní hodnoty, jakými jsou četnosti vrcholů s daným hodnocením, se odvíjejí od druhu algoritmu.

Pro algoritmus spreading activation jsou uvedeny navíc tabulky, které sledují vývoj četností uzlů s hodnotami jejich relevance v závislosti na parametrech, jež jsou pro tento algoritmus specifické.

5.4.2.1 Union colours algoritmus

Algoritmus slučování barev je prakticky pouze prohledávání do šířky s průběžným ohodnocováním vrcholů hodnotou počátečního. V posledních dvou sloupcích tabulky 5.3 je patrné toto sloučení pro dva startovní uzly; hodnoceno je přirozenými čísly. A jelikož se jedná o základní typ systematického prohledávání, ostatní algoritmy by měly mít části svých tabulek pro jeden počáteční vrchol velmi podobné této.

5.4.2.2 Energy spreading algoritmus

Zcela zřetelné je rozložení četností relevancí jednotlivých uzlů v obou tabulkách 5.4 a 5.5 (rozděleno z důvodu množství dat) oproti algoritmu slučování barev. Pracuje se pouze s hodnotami z intervalu $(0; 1\rangle$.

5.4.2.3 Upravený Dijkstrův algoritmus

Algoritmus pro hledání optimální cesty v grafu s hodnocením hran přirozenými čísly, jemuž přísluší tabulka 5.6, vykazuje velmi podobné chování jako

algoritmus slučování barev, nejvyšší hodnoty optimálních cest se promítají do hodnot relevance a četností v posledních dvou sloupcích.

5.4.2.4 Spreading activation algoritmus

Spreading activation kromě parametru na omezení hloubky obsahuje i parametr váhy a útlumu. Tabulka 5.7 sleduje hodnocení vrcholů z jednoho a dvou startovních uzlů při standardním nastavení parametrů běhu, zatímco v tabulkách 5.8 a 5.9 jsou výsledky ohodnocení uzlů po experimentech s činiteli váhy a útlumu.

Z důvodu úspory místa se v tabulce 5.9 o dvou počátečních IRI vyskytuje znak vlnovky „~“, který zastupuje již jednou zanesené hodnoty. Výsledky jsou totiž stejné pro symetrické ohodnocení součinitelů váhy a útlumu: například pro $w = 0,2$ a $d = 0,4$ i pro $w = 0,4$ a $d = 0,2$ jsou hodnoty r spolu s n_r totožné.

Tabulka 5.3: Chování union colours algoritmu

	Selekce		IR						ITR				
	Hloubka		1			2			1			2	
$\mathbf{I_n}$	IRI (dbr:)	$\mathbf{r_S}$ (–)	$\mathbf{f_V}$ (–)	$\mathbf{n_V}$ (–)	$\mathbf{k}$ (–)	$\mathbf{f_V}$ (–)	$\mathbf{n_V}$ (–)	$\mathbf{k}$ (–)	$\mathbf{r}$ (–)	$\mathbf{n_r}$ (–)	$\mathbf{r}$ (–)	$\mathbf{n_r}$ (–)
1	Planet_Earth_(film)	1,000	1	25	24,000	1	1709	0,078677	–	–	1,000	379
						2	208					
						3	66					
						4	8					
						5	5					
						8	8					
						9	9					
						22	22					
2	Planet_Earth_(film)	1,000	1	48	10,615	1	1749	0,078677	0,500	1	0,500	48
	I_Love_Lucy_(film)	0,500	2	2		2	288				1,000	377
						3	87				1,500	2
						4	28					
						5	5					
						6	6					
						8	8					
						9	9					
						23	23					

Tabulka 5.4: Chování energy spreading algoritmu (1/2)

Selekce			IR						ITR			
Hloubka			1			2			1		2	
I_n	IRI (dbr:)	r_S (–)	f_V (–)	n_V (–)	k (–)	f_V (–)	n_V (–)	k (–)	r (–)	n_r (–)	r (–)	n_r (–)
1	Planet_Earth_(film)	1,000	1	25	24,000	1	1709	0,078677	–	–	0,000056	17
						2	208				0,000218	157
						3	66				0,000226	13
						4	8				0,000282	1
						5	5				0,000366	2
						8	8				0,000463	22
						9	9				0,000468	2
						22	22				0,000496	12
											0,000514	52
											0,000585	1
											0,000672	10
											0,000681	3
											0,000694	28
											0,000722	10
											0,000890	1
											0,000913	1
											0,001016	4
											0,001096	26
											0,001185	1
											0,001190	1
											0,001302	7
											0,001315	3
											0,001997	1
											0,005793	1
											0,010204	1
											0,010927	1
											0,013206	1

Tabulka 5.5: Chování energy spreading algoritmu (2/2)

Selekce			**IR**						**ITR**			
Hloubka			**1**			**2**			**1**		**2**	
$\mathbf{I_n}$	**IRI (dbr:)**	$\mathbf{r_S}$ (–)	$\mathbf{f_V}$ (–)	$\mathbf{n_V}$ (–)	**k** (–)	$\mathbf{f_V}$ (–)	$\mathbf{n_V}$ (–)	**k** (–)	**r** (–)	$\mathbf{n_r}$ (–)	**r** (–)	$\mathbf{n_r}$ (–)
2	Planet_Earth_(film)	1,000	1	48	10,615	1	1796	0,079820	0,500	1	0,000056	17
	I_Love_Lucy_(film)	0,500	2	2		2	216				0,000218	157
						3	60				0,000226	13
						4	20				0,000282	1
						5	10				0,000366	2
						8	8				0,000463	63
						9	9				0,000468	2
						22	22				0,000496	12
											0,000514	52
											0,000585	1
											0,000595	7
											0,000672	8
											0,000681	3
											0,000694	28
											0,000722	10
											0,000890	1
											0,000913	1
											0,001008	1
											0,001016	4
											0,001096	26
											0,001185	1
											0,001190	1
											0,001302	7
											0,001315	3
											0,001997	1
											0,005793	1
											0,010204	1
											0,010927	1
											0,013206	1
											0,522900	1

Tabulka 5.6: Chování upraveného Dijkstrova algoritmu

		Selekce	**IR**						**ITR**			
		Hloubka	**1**			**2**			**1**		**2**	
$\mathbf{I_n}$	**IRI (dbr:)**	$\mathbf{r_S}$ (–)	$\mathbf{f_V}$ (–)	$\mathbf{n_V}$ (–)	**k** (–)	$\mathbf{f_V}$ (–)	$\mathbf{n_V}$ (–)	**k** (–)	**r** (–)	$\mathbf{n_r}$ (–)	**r** (–)	$\mathbf{n_r}$ (–)
1	Planet_Earth_(film)	1,000	1	25	24,000	1	1709	0,078677	–	–	3,000	372
						2	208				4,000	7
						3	66					
						4	8					
						5	5					
						8	8					
						9	9					
						22	22					
2	Planet_Earth_(film)	1,000	1	48	10,615	1	1749	0,070872	0,500	1	2,500	48
	I_Love_Lucy_(film)	0,500	2	2		2	288				3,000	370
						3	87				4,000	7
						4	28				5,500	2
						5	5					
						6	6					
						8	8					
						9	9					
						23	23					

Tabulka 5.7: Chování algoritmu spreading activation

	Selekce			IR						ITR			
	Hloubka			1			2			1		2	
I_n	IRI (dbr:)	r_S (–)	w (–)	d (–)	f_V (–)	n_V (–)	k (–)	f_V (–)	n_V (–)	k (–)	r (–)	n_r (–)	r (–)	n_r (–)
1	Planet_Earth_(film)	1,000	1,000	0,500	1	25	24,000	1	1654	0,051428	–	–	0,250	377
								2	196					
								3	45					
								4	8					
								5	5					
								8	8					
								9	9					
								21	21					
2	Planet_Earth_(film)	1,000	1,000	0,500	1	48	10,615	1	1695	0,048628	0,500	1	0,125	48
	I_Love_Lucy_(film)	0,500			2	2		2	272				0,250	375
								3	81				0,375	1
								4	16				0,750	1
								5	5					
								6	6					
								8	8					
								9	9					
								22	22					

Tabulka 5.8: Závislost relevance na parametrech spreading activation (1/2)

				Selekce		ITR	
I_n	IRI (dbr:)	r_S (–)	h (–)	w (–)	d (–)	r (–)	n_r (–)
1	Planet_Earth_(film)	1,000	2	0,2	0,2	0,0016	377
					0,4	0,0064	
					0,6	0,1600	
					0,8	0,0256	
					1,0	0,0400	
				0,4	0,2	0,0064	
					0,4	0,0256	
					0,6	0,0576	
					0,8	0,1024	
					1,0	0,1600	
				0,6	0,2	0,1600	
					0,4	0,0576	
					0,6	0,1296	
					0,8	0,2304	
					1,0	0,3600	
				0,8	0,2	0,0256	
					0,4	0,1024	
					0,6	0,2304	
					0,8	0,4096	
					1,0	0,6400	
				1,0	0,2	0,0400	
					0,4	0,1600	
					0,6	0,3600	
					0,8	0,6400	

Tabulka 5.9: Závislost relevance na parametrech spreading activation (2/2)

				Selekce		ITR	
I_n	IRI (dbr:)	r_S (–)	h (–)	w (–)	d (–)	r (–)	n_r (–)
2	Planet_Earth_(film)	1,000	2	0,2	0,2	0,0008	48
	I_Love_Lucy_(film)	0,500				0,0016	375
						0,0024	1
						0,5016	1
					0,4	0,0032	48
						0,0064	375
						0,0096	1
						0,5064	1
					0,6	0,0072	48
						0,0144	375
						0,0216	1
						0,5144	1
					0,8	0,0128	48
						0,0256	375
						0,0384	1
						0,5256	1
					1,0	0,0200	48
						0,0400	375
						0,0600	1
						0,5400	1
				0,4	0,2	~	~
					0,4	0,0128	48
						0,0256	375
						0,0384	1
						0,5256	1
					0,6	0,0288	48
						0,0576	375
						0,0864	1
						0,5576	1
					0,8	0,0512	48
						0,1024	375
						0,1536	1
						0,6024	1
					1,0	0,0800	48
						0,1600	375
						0,2400	1
						0,6600	1

Pokračování tabulky na další straně.

Dokončení tabulky 5.9 z předchozí strany

				Selekce		ITR	
$\mathbf{I_n}$	**IRI (dbr:)**	$\mathbf{r_S}$ (–)	**h** (–)	**w** (–)	**d** (–)	**r** (–)	$\mathbf{n_r}$ (–)
2	Planet_Earth_(film)	1,000	2	0,6	0,2	~	~
	I_Love_Lucy_(film)	0,500			0,4	~	~
					0,6	0,0648	48
						0,1296	375
						0,1944	1
						0,6296	1
					0,8	0,1152	48
						0,2304	375
						0,3456	1
						0,7304	1
					1,0	0,1800	48
						0,3600	375
						0,5400	1
						0,8600	1
				0,8	0,2	~	~
					0,4	~	~
					0,6	~	~
					0,8	0,2048	48
						0,4096	375
						0,6144	1
						0,9096	1
					1,0	0,3200	48
						0,6400	375
						0,9600	1
						1,1400	1
				1,0	0,2	~	~
					0,4	~	~
					0,6	~	~
					0,8	~	~

5.5 Případy užití

Na několika následujících ukázkách jsou nastíněny možné situace, do nichž může uživatel průchodem grafu, za použití daného algoritmu, dospět. Situace se odehrávají na dvou zvolených datasetech (DBpedia a LinekdMDB) jedné oblasti (doména filmů). V ukázkových tabulkách se vyskytují atributy jako

IRI zdroje, který byl nalezen;

R neboli hodnota relevance zdroje;

$\mathbf{R_C}$ čili zvolená důležitost uživatelem.

Do dalšího kroku jsou zahrnuty pouze takové zdroje, jejichž hodnota zvolené důležitosti je ostře větší než nula. Tabulky s ukázkovými výsledky jsou omezeny na prvních deset položek.

5.5.1 DBpedia

Přístup přes SPARQL koncový bod k popisům zdrojů. Prohledává se vždy nejvýše do hloubky zanoření rovné dvěma, počáteční vrchol je vždy hodnocen relevancí rovné jedné.

5.5.1.1 Union colours

Počátečním IRI je `http://dbpedia.org/resource/Planet_Earth_(film)`.
V prvním kroku průchodu jsou nabídnuty výsledky s relevancí rovné jedné, jelikož počáteční vrchol je ohodnocen nejvyšší relevancí, která se propaguje dále.

Tabulka 5.10: DBpedia: Union colours – výsledky po 1. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
Superman_(1978_film)	1	0,9
Goin'_Coconuts	1	0
How_to_Make_a_Monster_(2001_film)	1	0
Godzilla_vs._Gigan	1	0,6
Cutter's_Trail	1	0
Avatar_(2009_film)	1	0
Graceless_Go_I	1	0,2
Ghosts_of_Mars	1	0
Tully_(1974_film)	1	0
Science_Ninja_Team_Gatchaman:_The_Movie	1	0

Do druhého kroku postupují tři uzly (z tabulky 5.10), kde je již patrná změna v hodnocení, z čehož lze vyvozovat, že některé z předchozích vybraných zdrojů

měly společný nějaký vrchol v grafu. První dvě IRI z tabulky 5.11 pak postupují do posledního ukázkového kroku.

Tabulka 5.11: DBpedia: Union colours – výsledky po 2. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
Where_Have_All_The_People_Gone%3F	1,7	0,7
Damnation_Alley_(film)	1,5	0,2
The_Happiness_Cage	1,5	0
The_War_in_Space	1,5	0
THX_1138	1,5	0
Star_Trek:_The_God_Thing	1,5	0
Soylent_Green	1,5	0
A_Cold_Night's_Death	1,5	0
A_Cosmic_Christmas	1,5	0
Travelers:_Jigen_Keisatsu	1,5	0

Poslední, třetí, krok (tabulka 5.12) je poznamenán zcela opačným jevem než v předchozí tabulce. Bohužel žádný z nyní vybraných zdrojů neměl společný uzel, tudíž se hodnocení vrátilo zpět k rozložení patrnému na začátku průchodu.

Tabulka 5.12: DBpedia: Union colours – výsledky po 3. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
The_Happiness_Cage	0,9	0
The_War_in_Space	0,9	0
THX_1138	0,9	0
Star_Trek:_The_God_Thing	0,9	0
Soylent_Green	0,9	0
A_Cold_Night's_Death	0,9	0
A_Cosmic_Christmas	0,9	0
Science_Ninja_Team_Gatchaman:_The_Movie	0,9	0
Embryo_(1976_film)	0,9	0
Colossus:_The_Forbin_Project	0,9	0

Uživatel u tohoto algoritmu nemusí získávat příliš mnoho různých hodnot relevance. Neobdrží-li informaci o důležitosti nabídky, je nucen se rozhodnout dle jiných vodítek, takže další rozhodování mu nebylo ulehčeno.

5.5.1.2 Energy spreading

Počátečním IRI je `http://dbpedia.org/resource/I_Love_Lucy_(film)`. Již v prvém kroku je z tabulky 5.13 viditelná různost výsledků; nízké hodnoty relevance jsou způsobeny jejich rozprostíráním do šířky.

Tabulka 5.13: DBpedia: Energy spreading – výsledky po 1. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
Jake_Spanner,_Private_Eye	0,001190	0
Beyond_the_Valley_of_the_Dolls	0,001190	0
Tough_Enough_(film)	0,001190	0
The_Octagon_(film)	0,001190	0
Heaven_with_a_Gun	0,001190	0,4
Zero_to_Sixty	0,001190	0
Black_Noon	0,001190	0,2
The_Gentleman_from_America	0,000926	0
Maker_of_Men	0,000926	1,0
Ma_and_Pa_Kettle_Back_on_the_Farm	0,000926	0

Tři vybrané zdroje postupují dále. Výstupem druhého kroku je tabulka 5.14 níže s velmi jemnými rozdíly mezi značně nízkými hodnotami relevance.

Tabulka 5.14: DBpedia: Energy spreading – výsledky po 2. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
A_Different_Story	0,003774	0
Bachelor's_Affairs	0,003676	0
It's_Great_to_Be_Alive_(film)	0,003676	0
Secret_Service_in_Darkest_Africa	0,003676	0
The_Lady_Objects	0,003676	0
In_Too_Deep_(film)	0,002857	0,8
Morgan_Stewart's_Coming_Home	0,002857	0
Beneath_the_Valley_of_the_Ultra-Vixens	0,002547	0
Up!_(1976_film)	0,002482	0
I_Love_Lucy_(film)	0,002024	1,0

Poslední krok (přehledová tabulka 5.15) ukazuje stav po výběru dvou zdrojů, přičemž druhý je totožný se startovním. Do seznamu se tento probojoval až ve druhém kroku, protože zdroje totožné s výchozími se ze stávajících výsledků vždy odstraňují.

Tabulka 5.15: DBpedia: Energy spreading – výsledky po 3. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
The_Octagon_(film)	0,006524	0
Perfume_(2001_film)	0,006401	0
Angel_Baby_(1995_film)	0,005833	0
Queen_of_the_Damned	0,005833	0
A_Different_Story	0,005333	0
Morgan_Stewart's_Coming_Home	0,005333	0
Face_to_Face_(2011_film)	0,004430	0
Allie_&_Me	0,004430	0
The_People_vs,_Larry_Flynt	0,002963	0
A_Reason_to_Believe	0,002963	0

Nyní je uživateli předávána celá plejáda výsledků s rozličným ohodnocením, jeho výběr je rozmanitější a obsahuje více informace nežli tomu bylo v případě výše diskutovaného algoritmu slučování barev.

5.5.1.3 Modified Dijkstra

Modifikovaný Dijkstrův algoritmus tentokrát nestartuje ze zdroje příslušícího filmu, nýbrž ze zdroje, který identifikuje herce: `http://dbpedia.org/resource/Daniel_Craig`. Vysoké hodnoty relevance jsou zapříčiněny výpočtem nejdelší cesty ve stromě – odrážejí optimální vzdálenost od počátečního vrcholu a uvádí je tabulka 5.16.

Tabulka 5.16: DBpedia: Modified Dijkstra – výsledky po 1. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
I_Want_You_(1998_film)	4	0
Tara_Road_(film)	3	0
Obsession_(1997_film)	3	0
The_Incredible_World_of_James_Bond	3	0,9
Bond_Girls_Are_Forever	3	0
Harry_Saltzman:_Showman	3	0
Männerpension	3	0
Borat	3	0
Hilde_(film)	3	0
Infamous_(film)	2	0,3

Druhý krok, 5.17, vychází ze dvou zvolených zdrojů. Zdroj s nejvyšší relevancí ovšem není zahrnut.

Tabulka 5.17: DBpedia: Modified Dijkstra – výsledky po 2. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
The_Thin_Blue_Line_(1965_film)	2,9	0,2
Wattstax	2,9	0
Jesus_(1979_film)	2,9	0,9
Mayhem_on_a_Sunday_Afternoon	2,9	0
Visions_of_Eight	2,9	0
The_Love_Machine_(film)	2,9	0,5
That's_Entertainment!	2,9	0
Murder_in_Mississippi	2,9	0
Imagine:_John_Lennon	2,9	0
The_Bridge_at_Remagen	2,9	0

Ani tentokrát se nepodařilo algoritmu najít společné uzly v grafu propojených dat, ba co více, všechny nalezené cesty mají stejnou vzdálenost od obou vyvolených zdrojů. Tabulka 5.18 po třetím kroku je výsledkem průchodu grafu s hodnocením ze třech vrcholů.

Tabulka 5.18: DBpedia: Modified Dijkstra – výsledky po 3. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
The_Incredible_World_of_James_Bond	7,6	0
Ben-Hur_(1959_film)	5,4	0
Left_Behind:_The_Movie	5,4	0
Left_Behind_(2014_film)	5,4	0
Ben-Hur_(1925_film)	5,4	0
Left_Behind:_World_at_War	5,4	0
The_Perfect_Stranger_(film)	5,4	0
The_Ultimate_Gift	5,4	0
Elmer_Gantry_(film)	5,4	0
Nikki_and_the_Perfect_Stranger	5,4	0

V tomto případě se rozdílné hodnocení projevilo pouze u první položky tabulky 5.18. Zatímco tabulka 5.17 ani zde nepřináší žádnou směrodatnou informaci, dle níž by se mohl uživatel rozhodnout.

5.5.1.4 Spreading activation

Výchozím bodem pro traverzování ještě jednou zvolme herce, respektive herečku: `http://dbpedia.org/resource/Natalie_Portman`. První krok obsahuje hodnoty relevance odpovídající aktivačním hodnotám uzlů v první a druhé úrovni zanoření do grafu propojených dat.

Tabulka 5.19: DBpedia: Spreading activation – výsledky po 1. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
Star_Wars_Episode_I:_The_Phantom_Menace	0,500	0,9
Goya's_Ghosts	0,50	0
Eve_(2008_film)	0,50	0
Beautiful_Girls_(film)	0,50	0
Pride_and_Prejudice_and_Zombies_(film)	0,50	0
New_York,_I_Love_You	0,50	0
Me,_Natalie	0,25	0
Bachelorette_(film)	0,25	0
Natalie_(film)	0,25	0
The_Black_Swan_(film)	0,25	0,4

Po druhém kroku se v tabulce 5.20 na první místo dostává zdrojové IRI filmu Černá labuť `dbr:Black_Swan_(film)`, v předchozím kroku (tab. 5.19) byl vybrán mimo jiné i zdroj `dbr:The_Black_Swan_(film)`, ale protože jsou IRI dle rovnosti řetězců odlišná, postup algoritmu během výběru zdrojů je korektní.

Tabulka 5.20: DBpedia: Spreading activation – výsledky po 2. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
Black_Swan_(film)	0,2	0
The_Age_of_Innocence_(1934_film)	0,1	0
The_Lost_World:_Jurassic_Park	0,1	0,5
The_Far_Call	0,1	0
April_Love_(film)	0,1	0,5
Vatel_(film)	0,1	0
To_Have_and_Have_Not_(film)	0,1	0
How_to_Marry_a_Millionaire	0,1	0
The_Naked_City	0,1	0
Folies_Bergère_de_Paris	0,1	0

Na závěr se opět projevuje rozdílnost dvou vybraných vrcholů z druhého kroku, kterým chybí společný uzel. Návodná informace o dalším možném postupu uživatele dle doporučení tu zaniká (viz tabulka 5.21).

Tabulka 5.21: DBpedia: Spreading activation – výsledky po 3. kroku

IRI (dbr:)	**R** (–)	$\mathbf{R_C}$ (–)
To_Have_and_Have_Not_(film)	0,25	0
Because_They're_Young	0,25	0
The_Gambler_from_Natchez	0,25	0
Last_Summer	0,25	0
White_Fang_2:_Myth_of_the_White_Wolf	0,25	0
His_Majesty_O'Keefe	0,25	0
Divergent_(film)	0,25	0
The_Blazing_Forest	0,25	0
Attack_of_the_Crab_Monsters	0,25	0
South_Pacific_(1958_film)	0,25	0

5.5.2 LinkedMDB

Přístup k metadatům je realizován skrze přímou dereferenci HTTP URI. Prohledává se vždy nejvýše do hloubky zanoření rovné dvěma, počáteční vrchol je vždy hodnocen relevancí rovné jedné.

Vzhledem ke skutečnosti, že se jedná o dataset s velice kolísavou dostupností, nelze se spoléhat na kompletnost množiny výsledných IRI. Názvy zdrojů jsou ze strany LinkedMDB vytvořeny strojově, ke každému podstatnému je proto uvedena vysvětlivka, jakou entitu identifikuje.

5.5.2.1 Union colours

Zdroj `http://data.linkedmdb.org/resource/film/1` je počátečním vrcholem algoritmu slučování barev. Popisuje film s názvem Buffy the Vampire Slayer (Buffy, přemožitelka upírů).

Výběr IRI po prvním kroku z tabulky 5.22 sestává ze zdrojů

- `http://data.linkedmdb.org/resource/film/13170` (film Mentor) a
- `http://data.linkedmdb.org/resource/film/16208` (film Astro Boy).

Úvodní krok prozatím poskytuje nejasné rozlišení důležitějších výsledků. Jejich sestupné řazení závisí pouze na hodnotě vypočtené relevance, při shodném ohodnocení a opakovaném spuštění algoritmu tedy není zaručeno, že se stejné zdroje budou nacházet na stejných pozicích v tabulce jako nyní.

Tabulka 5.22: LinkedMDB: Union colours – výsledky po 1. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/1577	1	0
film/13170	1	1,0
film/16208	1	0,1
film/38469	1	0
film/2825	1	0
film/200	1	0
film/203	1	0
film/204	1	0
film/201	1	0
film/202	1	0

Pakliže dojde k nedostupnosti daného zdroje nestabilitou serveru, není tento vůbec ve výsledcích zahrnut, byť by mohl být hodnocen třeba nejvyšší hodnotou důležitosti. Tato situace není ze strany aplikace nijak postihnutelná.

Poněkud extrémní případ představuje tabulka výsledků 5.23, kde ani ve druhém kroku nedošlo k rozlišení filmů dle hodnot relevance. Uživatel zůstává nespokojen, proto ohodnotil pro další krok všechny vybrané položky nejvyšší hodnotou zvolené důležitosti.

Ohodnocené filmy (v tabulce 5.23 shora dolů): City Hunter, Ride the High Country, Road to Morocco, Roman Holiday, President McKinley Inauguration Footage, The Prisoner of Zenda, The Public Enemy, Pull My Daisy, Rebel Without a Cause, Republic Steel Strike Riot Newsreel Footage.

Tabulka 5.23: LinkedMDB: Union colours – výsledky po 2. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/39359	1,1	1
film/38437	1,1	1
film/38438	1,1	1
film/38439	1,1	1
film/38430	1,1	1
film/38431	1,1	1
film/38432	1,1	1
film/38433	1,1	1
film/38434	1,1	1
film/38435	1,1	1

Po radikálním přístupu k hodnocení v předchozím případě již závěrečný třetí krok (tabulka 5.24) obsahuje uspokojující hodnocení s vyšší informační hodnotou.

Tabulka 5.24: LinkedMDB: Union colours – výsledky po 3. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/38081	3	0
film/677	3	0
film/674	2	0
film/38082	2	0
film/38083	2	0
film/675	2	0
film/38084	2	0
film/38085	1	0
film/38086	1	0
film/678	1	0

5.5.2.2 Energy spreading

Pro případ s algoritmem rozprostírání energie byl jako počáteční uzel vybrán zdroj identifikující filmový žánr Conspiracy theory (filmy s nálepkou konspirační teorie) `http://data.linkedmdb.org/resource/film_genre/1`.

Tabulka 5.25: LinkedMDB: Energy spreading – výsledky po 1. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/45	1,0	0,6

Pokus v tomto případě odhalil pouze jediný film ihned v první úrovni zanoření, proto má nejvyšší hodnotu relevance. V dalším kroku však již žádné další IRI nepřibyly. Důvodem může být buďto jejich skutečná absence v popisu zdroje `http://data.linkedmdb.org/resource/film/45`, nebo výpadek spojení se serverem.

5.5.2.3 Modified Dijkstra

Startuje se z IRI `http://data.linkedmdb.org/resource/film/2016` filmu Ecstasy. Po prvém kroku byla dle tabulky 5.26 vybrána IRI

- `http://data.linkedmdb.org/resource/film/39358` (film The Falls) a
- `http://data.linkedmdb.org/resource/film/38439` (film Roman Holiday).

Tabulka 5.26: LinkedMDB: Modified Dijkstra – výsledky po 1. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/39356	3	0
film/39357	3	0
film/39358	3	0,5
film/39359	3	0
film/38437	3	0
film/38438	3	0
film/38439	3	0,6
film/4642	3	0
film/15398	3	0
film/38430	3	0

Ani po druhém kroku (viz tabulka 5.27) se se na výseku dat neprojevilo rozdílné hodnocení zdrojů, můžeme zde ale pozorovat, že všechny filmy jsou nyní společné oběma hodnoceným uzlům ve druhém kroku.

Tabulka 5.27: LinkedMDB: Modified Dijkstra – výsledky po 2. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/38464	5,1	0
film/38465	5,1	0
film/38466	5,1	0
film/38467	5,1	0
film/38468	5,1	0,9
film/38469	5,1	0,8
film/200	5,1	0,3
film/203	5,1	0,6
film/204	5,1	0
film/201	5,1	0

Do dalšího kola postupují filmy

- `http://data.linkedmdb.org/resource/film/38468` (One Foot in Heaven),
- `http://data.linkedmdb.org/resource/film/38469` (Sergeant York),
- `http://data.linkedmdb.org/resource/film/200` (Z),
- `http://data.linkedmdb.org/resource/film/203` (Planet of the Apes).

Výsledkem závěrečného doporučování je tabulka 5.28.

Tabulka 5.28: LinkedMDB: Modified Dijkstra – výsledky po 3. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/45728	5,60	0
film/2189	4,20	0
film/83057	3,90	0
film/61478	3,80	0
film/98224	3,60	0
film/40387	3,30	0
film/42760	2,80	0
film/47885	2,50	0
film/8062	2,30	0
film/41128	2,00	0

5.5.2.4 Spreading activation

Konečně poslední algoritmus necháme rozeběhnout z IRI herečky Vanessy L. Williamsové `http://data.linkedmdb.org/resource/actor/24`.

Tabulka 5.29: LinkedMDB: Spreading activation – výsledky po 1. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/20717	0,50	0,7
film/20788	0,25	0
film/1851	0,25	0
film/22413	0,25	0,8

Nalezeny byly pouze čtyři výsledky (tabulka 5.29), do dalšího kroku se probojovaly

- `http://data.linkedmdb.org/resource/film/20717` (film Light it Up),
- `http://data.linkedmdb.org/resource/film/22413` (film Hannah Montana & Miley Cyrus: Best of Both Worlds Concert).

Mezi hodnotami relevance 2. kroku se ještě objevilo číslo 0,175, které se ale společně s jemu příslušejícími IRI nedostalo do první desítky. Do posledního kroku shrnutém v tabulce 5.31 byly vybrány vrcholy

- `http://data.linkedmdb.org/resource/film/296` (film Hamlet) a
- `http://data.linkedmdb.org/resource/film/38862` (film The Moon Is Blue).

Tabulka 5.30: LinkedMDB: Spreading activation – výsledky po 2. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/299	0,375	0
film/296	0,375	0,4
film/297	0,375	0
film/38860	0,375	0
film/38861	0,375	0
film/38862	0,375	0,6
film/38863	0,375	0
film/38864	0,375	0
film/38865	0,375	0
film/22480	0,200	0

Tabulka 5.31: LinkedMDB: Spreading activation – výsledky po 3. kroku

IRI (lmdbr:)	**R** (–)	$\mathbf{R_C}$ (–)
film/39318	0,25	0
film/203	0,25	0
film/202	0,25	0
film/38072	0,25	0
film/679	0,25	0
film/139	0,25	0
film/583	0,25	0
film/586	0,25	0
film/38974	0,25	0
film/1016	0,25	0

Závěr průchodu sice obsahuje stejné hodnoty relevance, avšak mimo prvních deset nejlépe hodnocených záznamů byly položky hodnoceny také důležitostí 0,15 a 0,1.

5.6 Diskuse výsledků, shrnutí

Kvalitní datová sada na škálovatelném serveru poskytujícím přístup přes koncový bod pro dotazování SPARQL představuje ideální základ pro běh aplikace. Doba výpočtů je ovlivněna především rychlostí komunikace se serverem, výsledky pak jeho dostupností. V tomto směru je DBpedie jasným favoritem, neboť i přes občasné servisní odstávky vykazuje vyšší dostupnost nežli LinkedMDB a nabízí SPARQL endpoint i s webovým rozhraním pro dotazy.

Co se týče algoritmů, nejčastěji se vyskytujících rozdílů hodnot relevance ve výsledcích se dá dosáhnout s energy spreding a spreading activation, i když čísla relevance mohou být velmi malá, neboť počáteční (energie) se postupně snižuje (vytrácí). Spreading algoritmy se v praxi využívají právě v oblasti sémantického vyhledávání [18]. U ostatních algoritmů hodnoty důležitosti společných vrcholů grafu narůstají. Doporučena je proto vhodná normalizace výsledků.

Výsledky také jasně ukazují, že je třeba u všech algoritmů volit parametr maximální hloubky zanoření, případně jiné parametry, které s hloubkou souvisejí, nejméně roven dvěma. Důvodem této skutečnosti je rozmístění vrcholů různých typů ve struktuře grafu. Aby byla zvýšena pravděpodobnost nalezení nějakých IRI popisujících filmy z různých typů počátečních vrcholů (film, herec, kategorie, ...), je nutné zanoření až do druhé úrovně, neboť schéma uzlů může mít tvar *film* $\longrightarrow$ *kategorie* $\longrightarrow$ *film*, kde počáteční film nemusí obsahovat odkazy na jiné filmy, ale může ukazovat na kategorii, která již s vysokou pravděpodobností zavede procházení k dalším filmům, jenž do ní spadají. Totéž například u schématu *film* $\longrightarrow$ *herec* $\longrightarrow$ *film*.

Výsledné hodnoty každé tabulky v jednotlivých krocích jsou silně závislé nejen na počátečním IRI a volbě algoritmu, ale především také na struktuře grafu, která obyčejně není předem známa. Čím více filmů v následujících krocích uživatel ohodnotí, tím více narůstá pravděpodobnost, že se jejich hodnocení setkají ve společných uzlech; tudíž obdrží rozmanitější doporučení dalších filmů, a tedy pestřejší výběr pro další hodnocení.

Závěr

Ačkoliv význam a potřeba propojených dat neustále roste, setkávám se s webovými aplikacemi, které nad nimi operují, takřka sporadicky. Jak vyplývá z poznatků o doporučovacích systémech, setkáme se v praxi pouze s několika málo nástroji pro doporučování entit v propojených datech, spíše narazíme na návrhy a teoretické studie. Nejhojněji zastoupeným typem aplikace pracující s propojenými daty jsou vyhledávače pro explorative search (některé zahrnují i vylepšování výsledků doporučením), tagovací systémy a vizualizační nástroje.

V rámci této diplomové práce byly představeny principy sémantického webu, propojených dat a různé techniky doporučování spolu s analýzou algoritmů vhodných k procházení grafů. Dále též byla navržena a vyvinuta aplikace pro inteligentní procházení grafovou strukturou propojených dat v oblasti filmů a jejich doporučování uživatelům, kteří mají možnost výsledný obsah personalizovat dle svých vlastních preferencí.

Z experimentů provedených na existujících datových sadách vyplývá skutečnost, že vhodné algoritmy pro úlohu doporučování v grafech pocházejí z rodiny „spreading“, totiž spreading activation a energy spreading, které se s úspěchem používají především v sémantickém vyhledávání. Pokusy rovněž potvrdily, že si aplikace musí poradit s vysokým objemem dat z důvodu přílišné rozsáhlosti grafu, což se negativně podepisuje na době jejich zpracování. Nejslabším místem je pak pochopitelně komunikace se serverem uchovávajícím data, na jehož vysokou dostupnost aplikace spoléhá.

Okrajově je v práci naznačeno i využití výstupních hodnot grafových doporučovacích algoritmů jako vstupních parametrů klasických přístupů k doporučování, zejména pro řešení problémů studeného startu. Klasické metody doporučování a jejich spolupráce s dalšími přístupy a aplikacemi nechť jsou předmětem zamyšlení nad tématy příštích prací.

Možná rozšíření a budoucí práce

Budoucí možná návaznost na tuto práci a možné rozšiřování nejen jí, ale také představené aplikace, lze spatřovat v několika oblastech a tematických celcích otevřených diskusi i eventuálnímu vylepšování. Do těchto okruhů spadá mimo jiné například optimalizace běhu algoritmů a nové přístupy k doporučování, využití výstupů aplikace v dalších systémech, obohacení o uživatelskou část a implementace klasických doporučovacích technik nebo i zkvalitňování prezentační části a uživatelského rozhraní.

Návazným tématem na oblast vývoje doporučovacích systémů je taktéž jejich testování. Ať se již jedná o obecné doporučovací systémy, nebo úzce specializované právě na oblast propojených dat. Otevřeným okruhem se tak stává problematika měření přesnosti doporučování a kvality výsledků v závislosti na požadavcích a očekávání uživatelů.

V neposlední řadě je možnost dotknout se samotné oblasti propojených dat, neboť návrh, vývoj a správa datových sad spolu s jejich vzájemným propojováním nejen obohacuje a rozšiřuje sémantický web o nové informace, nýbrž také může zavdat impuls k vytváření nových nástrojů a aplikací nad těmito daty vystavěnými.

Literatura

[1] AGHAEI, Sareh, Mohammad Ali NEMATBAKHSH a Hadi KHOSRAVI-FARSANI. Evolution of the World Wide Web: From Web 1.0 to Web 4.0. In: *International Journal of Web & Semantic Technology (IJWesT)* [online]. Chennai, Tamil Nadu, India: AIRCC Publishing Corporation, 2012, s. 10 [cit. 2016-03-09]. 3, 1. ISSN 0975-9026. Dostupné z: `http://airccse.org/journal/ijwest/papers/3112ijwest01.pdf`

[2] ARRASCUE AYALA, Victor Anthony. *ReSPARQL: a SPARQL Extension for Generic Recommendations on RDF-graphs* [online]. Freiburg, 2014 [cit. 2016-03-16]. MN 3209050. Dostupné z: `http://dbis.informatik.uni-freiburg.de/content/team/arrascue/thesis/Arrascue-ReSPARQL-20140226.pdf`. Diplomová práce. University of Freiburg, Department of Computer Science, Chair of Databases and Information Systems. Vedoucí práce Martin Przyjaciel-Zablocki, M. Sc. Oponent: Prof. Dr. Georg Lausen.

[3] AUER, Sören, Volha BRYL a Sebastian TRAMP (eds.). *Linked open data - creating knowledge out of interlinked data: results of the LOD2 project.* VI. London: Springer, 2014. Lecture notes in computer science. ISBN 978-3-319-09845-6. ISSN 0302-9743. DOI 10.1007/978-3-319-09846-3.

[4] BERNERS-LEE, Tim. Semantic Web - XML2000. In: *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics [cit. 2016-03-11]. Dostupné z: `https://www.w3.org/2000/Talks/1206-xml2k-tbl/`. Webová prezentace.

[5] BERNERS-LEE, Tim. Linked Data. *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial

Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 2006, 2009-06-18 [cit. 2016-03-11]. Dostupné z: https://www.w3.org/DesignIssues/LinkedData.html

[6] BERNERS-LEE, Tim. The World Wide Web: A very short personal history. *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 1998/05/07 [cit. 2016-03-09]. Dostupné z: https://www.w3.org/People/Berners-Lee/ShortHistory.html

[7] BERNERS-LEE, Tim. Web architecture: Metadata. *Design Issues for the World Wide Web (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 1997 [cit. 2016-03-10]. Dostupné z: https://www.w3.org/DesignIssues/Metadata.html. Personal view, but corresponds generally to the W3C architecture for metadata.

[8] BERNERS-LEE, Tim, James HENDLER a Ora LASSILA. The Semantic Web. *Scientific American* [online]. 2001, **284**(5), 34-43 [cit. 2016-03-09]. DOI: 10.1038/scientificamerican0501-34. ISSN 00368733. Dostupné z: http://www.nature.com/doifinder/10.1038/scientificamerican0501-34

[9] BIZER, Christian a Tom HEATH. *Linked data: evolving the Web into a global data space*. 1st ed. San Rafael, Calif.: Morgan & Claypool, 2011. ISBN 9781608454303. ISBN 9781608454310. DOI 10.2200/S00334ED1V01Y201102WBE001.

[10] BLAT, Josep, Jesús IBÁÑEZ a Toni NAVARRETE. *Introduction to ontologies and tools: some examples* [online]. Barcelona, 2004 [cit. 2016-03-10]. Dostupné z: http://www.dtic.upf.edu/~jblat/material/doctorat/ontologies.pdf. Universitat Pompeu Fabra Barcelona, Departament de Tecnologies de la Informació i les Comunicacions.

[11] BRATKOVÁ, Eva. Metadata jako nový nástroj pro komunikaci webovských informačních zdrojů. *Národní knihovna Knihovnická revue*. 1999, **10**(4), 178–195. ISSN 1214-0678. Dostupné také z: http://full.nkp.cz/nkkr/pdf/9904/9904178.pdf

[12] BURKE, Robin a (eds.). Hybrid Web Recommender Systems. In: BRUSILOVSKY, Peter, Alfred KOBSA a Wolfgang NEJDL. *The Adaptive Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, s. 377-408. DOI: 10.1007/978-3-540-72079-9_12. ISBN 978-3-540-72078-2. Dostupné také z: http://link.springer.com/10.1007/978-3-540-72079-9_12

[13] BURKE, Robin. *Hybrid Recommender Systems: Survey and Experiments* [online]. Fullerton, California, 2001 [cit. 2016-03-20]. Dostupné z: http://josquin.cti.depaul.edu/~rburke/pubs/burke-umuai02.pdf. California State University, Fullerton, Department of Information Systems and Decision Sciences.

[14] CASTANO, Silvana, Alfio FERRARA a Stefano MONTANELLI. InWalk: Interactive and Thematic Walks inside the Web of Data. In: AMER-YAHIA, Sihem, Vassilis CHRISTOPHIDES, Anastasios KEMENTSI-ETSIDIS, Minos GAROFALAKIS, Stratos IDREOS a Vincent LEROY (eds.). *Proceeding of the 17th International Conference on Extending Database Technology (EDBT 2014)*. Athens, Greece: OpenProceedings.org, 2014, s. 628-631. DOI: 10.5441/002/edbt.2014.60. ISBN 978-3-89318065-3. Dostupné také z: http://openproceedings.org/html/pages/2014_edbt.html

[15] CORBY. Corese / KGRAM | wimmics: web-instrumented man-machine interactions, communities and semantics. *Wimmics* [online]. 2012 [cit. 2016-04-09]. Dostupné z: http://wimmics.inria.fr/corese

[16] CYGANIAK, Richard, David WOOD a Markus LANTHALER (eds.). RDF 1.1 Concepts and Abstract Syntax. *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 2014 [cit. 2016-03-14]. Dostupné z: https://www.w3.org/TR/rdf11-concepts/

[17] ČERNÝ, Jakub. *Základní grafové algoritmy*. Praha, 2010. Dostupné také z: http://kam.mff.cuni.cz/~kuba/ka/ka.pdf. Výukový text. Univerzita Karlova, Matemeticko-fyzikální fakulta, Katedra aplikované matematiky.

[18] DAŘENA, František, Alexander TROUSSOV a Jan ŽIŽKA. Simulating activation propagation in social networks using the graph theory. In: *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*. Brno: Mendelova univerzita v Brně, 2010, **58**(3), s. 21-28. DOI: 10.11118/actaun201058030021. ISSN 1211-8516. Dostupné také z: http://acta.mendelu.cz/58/3/0021/

[19] DEMOVIČ, Luboš, Eduard FRITSCHER, Jakub KŘÍŽ, Ondrej KUZMÍK, Ondrej PROKSA, Diana VANDLÍKOVÁ, Dušan ZELENÍK a Mária BIELIKOVÁ. Movie Recommendation Based on Graph Traversal Algorithms. In: *2013 24th International Workshop on Database and Expert Systems Applications*. Los Alamitos, CA: IEEE, 2013, s. 152-156. DOI: 10.1109/DEXA.2013.24. ISBN 978-1-4799-2138-6. ISSN 1529-4188. Dostupné také z: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6621363

[20] FOURNIER, François. Recommender Systems: Technical Report and Literature Review. In: *Knol*. Aberdeen: IDEAS research institute, Robert Gordon University, Aberdeen, 2010, s. 6. DOI: 10.13140/RG.2.1.2306.6965.

[21] FRANK, Juraj, Vladimír DZIUBAN a Martin HOMOLA. Sémantický web: Niektoré aktuálne výzvy. In: KVASNIČKA, Vladimír, Jiří POSPÍCHAL, Pavol NÁVRAT, Peter LACKO a Peter TREBATICKÝ. *Umelá inteligencia a kognitívna veda II*. Bratislava: Vydavateľstvo STU, 2010, s. 34. ISBN 978-80-227-3284-0. Dostupné také z: `http://dai.fmph.uniba.sk/~homola/papers/uikv2010.pdf`

[22] KALELI, Cihan, Alper BILGE a Huseyin POLAT, GUNES, Ihsan (ed.). Shilling attacks against recommender systems: a comprehensive survey. In: LIU, Derong (ed.). *Artificial Intelligence Review*. Springer Netherlands, 2014, **42**(4), s. 767-799. DOI: 10.1007/s10462-012-9364-9. ISSN 0269-2821. Dostupné také z: `http://link.springer.com/10.1007/s10462-012-9364-9`

[23] GUTIERREZ, Claudio. Modeling the Web of Data (Introductory Overview). In: *Reasoning Web: Semantic Technologies for the Web of Data*. International Summer School 2011, Galway, Ireland: Springer Berlin Heidelberg, 2011, s. 416-444. 6848. DOI: 10.1007/978-3-642-23032-5_8. ISSN 0302-9743.

[24] HERLOCKER, Jonathan L., Joseph A. KONSTAN, Loren G. TERVEEN a John T. RIEDL. Evaluating collaborative filtering recommender systems. In: *ACM Transactions on Information Systems*. 1. Broadway, New York: ACM, Inc., 2004, **22**(1), s. 5-53. DOI: 10.1145/963770.963772. ISSN 10468188. Dostupné také z: `http://portal.acm.org/citation.cfm?doid=963770.963772`

[25] HERRMANN, Joseph. SETT February 2011 - The Semantic Web. *OCI* [online]. Saint Louis: Object Computing, Inc., 2015 [cit. 2016-03-11]. Dostupné z: `http://sett.ociweb.com/sett/settFeb2011.html`

[26] HLINĚNÝ, Petr. *Základy teorie grafů: pro (nejen) informatiky* [online]. Brno, 2010 [cit. 2016-04-06]. Dostupné z: `http://is.muni.cz/do/1499/el/estud/fi/js10/grafy/Grafy-text10.pdf`. Učební text. Masarykova univerzita, Fakulta informatiky.

[27] JACKSI, Karwan, Nazife DIMILILER a Subhi R. M. ZEEBAREE. A Survey of Exploratory Search Systems Based on Lod Resources. In: JAMALUDIN, Zulikha, Noraziah CHEPA, Wan Hussain WAN ISHAK a Syamsul Bahrin ZAIBON (eds.). *Proceedings of the 5th International*

Conference on Computing and Informatics, ICOCI 2015. Sintok, Malaysia: School of Computing, Universiti Utara Malaysia, Sintok, 2015, s. 501-509. ISBN 978-967-0910-01-7. ISSN 2289-3784. Dostupné také z: http://www.icoci.cms.net.my/proceedings/2015/PDF/PID112.pdf

[28] JANNACH, Dietmar, Markus ZANKER, Alexander FELFERNIG a Gerhard FRIEDRICH. *Recommender systems: an introduction*. 1. pub. New York: Cambridge University Press, 2011. ISBN 978-0-521-49336-9.

[29] JAROŠ, Vojtěch. *Experimentální systém pro sémantický web* [online]. Praha, 2011 [cit. 2016-03-14]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/jarosvo1_2011dipl.pdf. Diplomová práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačů. Vedoucí práce Doc. Ing. Ivan Jelínek, CSc.

[30] KAMBIL, Ajit. What is your Web 5.0 strategy? In: *Journal of Business Strategy*. 6. Boston: Emerald Group Publishing Limited, 2008, **29**(6), s. 56-58. DOI: 10.1108/02756660810917255. ISSN 0275-6668. Dostupné také z: http://www.emeraldinsight.com/doi/abs/10.1108/02756660810917255

[31] KLÍMEK, Jakub a Martin NEČASKÝ. *Introduction to Linked Data*. Praha, 2015. Přednášková prezentace. České vysoké učení technické v Praze, Fakulta informačních technologií; Univerzita Karlova v Praze, Matematicko-fyzikální fakulta.

[32] KOIVUNEN, Marja-Riitta a Eric MILLER. W3C Semantic Web Activity. *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 2001 [cit. 2016-03-10]. Dostupné z: https://www.w3.org/2001/12/semweb-fin/w3csw

[33] LAKSHMI, Soanpet .Sree a T.Adi LAKSHMI. Recommendation Systems: Issues and challenges. *International Journal of Computer Science and Information Technologies*. 2014, **5**(4), 5771-5772. ISSN 0975-9646. Dostupné také z: http://www.ijcsit.com/docs/Volume%205/vol5issue04/ijcsit20140504207.pdf

[34] MACHANAVAJJHALA, Ashwin, Johannes GEHRKE, Mirella M. MORO, et al. Resource Description Framework. In: *Encyclopedia of Database Systems*. Boston, MA: Springer US, 2009, s. 2423. DOI: 10.1007/978-0-387-39940-9_905. ISBN 978-0-387-35544-3. Dostupné také z: http://www.springerlink.com/index/10.1007/978-0-387-39940-9_905

[35] MANOLA, Frank, Eric MILLER a Brian MCBRIDE (eds.). RDF Primer. *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 2004, 2014-02-25 [cit. 2016-03-14]. Dostupné z: `https://www.w3.org/TR/2004/REC-rdf-primer-20040210/`

[36] 13. Nejkratší cesty. MAREŠ, Martin. *Krajinou grafových algoritmů: průvodce pro středně pokročilé* [online]. Vyd. 1. Praha: ITI, 2007 [cit. 2016-04-18]. ISBN 978-80-239-9049-2. Dostupné z: `http://mj.ucw.cz/vyuka/ga/13-dijkstra.pdf`

[37] MARIE, Nicolas. *Linked data based exploratory search* [online]. Nice, France, 2014 [cit. 2016-04-09]. Dostupné z: `https://tel.archives-ouvertes.fr/tel-01130622/file/2014NICE4129.pdf`. Disertační práce. Université Nice Sophia Antipolis, École Doctorale Sciences et Technologies de l'Information et de la Communication. Vedoucí práce Fabien Gandon.

[38] MARIE, Nicolas, Fabien GANDON, Damien LEGRAND a Myriam RIBIÈRE (eds.). Exploratory Search on the Top of DBpedia Chapters with the Discovery Hub Application. In: CIMIANO, Philipp, Miriam FERNÁNDEZ, Vanessa LOPEZ, Stefan SCHLOBACH a Johanna VÖLKER. *The Semantic Web: ESWC 2013 Satellite Events*. Montpellier, France: Springer Berlin Heidelberg, 2013, s. 184-188. DOI: 10.1007/978-3-642-41242-4_21. ISBN 978-3-642-41241-7. ISSN 0302-9743. Dostupné také z: `http://link.springer.com/10.1007/978-3-642-41242-4_21`

[39] MARIE, Nicolas a Damien LEGRAND. *Discovery Hub / Beta* [online]. 2013 [cit. 2016-04-09]. Dostupné z: `http://discoveryhub.co/`

[40] MATOUŠEK, Jiří a Jaroslav NEŠETŘIL. *Kapitoly z diskrétní matematiky*. 4., upr. a dopl. vyd. Praha: Karolinum, 2009. ISBN 978-80-246-1740-4.

[41] MATULÍK, Petr a Tomáš PITNER. Sémantický web a jeho technologie. ÚSTAV VÝPOČETNÍ TECHNIKY MASARYKOVY UNIVERZITY. *Zpravodaj ÚVT MU*. 2004, **14**(3), 15-17. ISSN 1212-0901. Dostupné také z: `http://webserver.ics.muni.cz/zpravodaj/clanky_tisk/296.pdf`

[42] MELVILLE, Prem a Vikas SINDHWANI. Recommender Systems. SAMMUT, Claude a Geoffrey I. WEBB (eds.). *Encyclopedia of Machine Learning* [online]. 1. New York: Springer-Verlag Berlin Heidelberg, 2010, kap. 338, s. 9 [cit. 2016-04-28]. DOI: 10.1007/978-0-387-30768-8. ISBN 978-0-387-30768-8. Dostupné z: `http://www.prem-melville.com/publications/recommender-systems-eml2010.pdf`

[43] MIRIZZI, Roberto, Tommaso DI NOIA, Azzurra RAGONE, Vito Claudio OSTUNI a Eugenio DI SCIASCIO. Movie Recommendation with DBpedia. In: *IIR 2012 Italian Information Retrieval Workshop*. Bari, Italy: CEUR-WS, 2012, s. 101-112. ISSN 1613-0073. Dostupné také z: http://ceur-ws.org/Vol-835/paper12.pdf

[44] DI NOIA, Tommaso a Eugenio DI SCIASCIO, MIRIZZI, Roberto a Azzurra RAGONE (eds.). Semantic Wonder Cloud: Exploratory Search in DBpedia. In: DANIEL, Florian a Federico MICHELE FACCA. *Current Trends in Web Engineering*. Germany: Springer-Verlag Berlin Heidelberg, 2010, s. 138. DOI: 10.1007/978-3-642-16985-4_13. ISBN 978-3-642-16984-7. ISSN 0302-9743. Dostupné také z: http://link.springer.com/10.1007/978-3-642-16985-4_13

[45] MIRIZZI, Roberto, Azzurra RAGONE, Tommaso DI NOIA a Eugenio DI SCIASCIO. Lookup, Explore, Discover: how DBpedia can improve your Web search. *IOS Press*. Bari, Italy: Politecnico di Bari, 2010, **Undefined**(0-1), 11. Dostupné také z: http://www.semantic-web-journal.net/sites/default/files/swj101.pdf

[46] NAVARA, Mirko. *Markovovy řetězce* [online]. Praha, 2014 [cit. 2016-04-18]. Dostupné z: http://cmp.felk.cvut.cz/~navara/psi/Markov_print.pdf. Učební text. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra kybernetiky, Centrum strojového vnímání.

[47] NAVATHE, Shamkant B. Evolution of data modeling for databases. *Communications of the ACM*. 1992, **35**(9), 112-123. DOI: 10.1145/130994.131001. ISSN 00010782.

[48] O'REILLY, Tim. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. In: O'REILLY MEDIA, INC. *O'Reilly Media - Technology Books, Tech Conferences, IT Courses, News* [online]. Boston: O'Reilly Media, Inc., 2005/09/30 [cit. 2016-03-09]. Dostupné z: http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html

[49] PASSANT, Alexandre. SEEVL LTD. *Seevl: mining music connections to bring context, search and discovery to the music you like*. Galway, Ireland, 2011. Dostupné také z: http://challenge.semanticweb.org/submissions2011/swc2011_submission_11.pdf. Digital Enterprise Research Institute, National University of Ireland.

[50] PATTUELLI, M. Cristina, Matt MILLER, Leanora LANGE, Sean FITZELL a Carolyn LI-MADEO. Crafting Linked Open Data for Cultural Heritage: Mapping and Curation Tools for the Linked Jazz Project. *The Code4Lib Journal*. 2013, (21). ISSN 1940-5758. Dostupné také z: http://journal.code4lib.org/articles/8670

[51] PEREGRIN, Jaroslav. *Úvod do teoretické sémantiky: principy formálního modelování významu*. 2. aktualiz. vyd. Praha: Karolinum, 2003. Učební texty Univerzity Karlovy v Praze. ISBN 80-246-0635-6.

[52] PHILLIPS, Addison, Martin J. DÜRST a Richard ISHIDA. Editing 'Internationalized Resource Identifiers (IRIs)'. *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 2005, 2010-09-16 [cit. 2016-03-11]. Dostupné z: `https://www.w3.org/International/iri-edit/`

[53] RADECKÝ, Michal. *Internetové technologie: sémantický web* [online]. Ostrava, 2015 [cit. 2016-03-10]. Dostupné z: `http://www.cs.vsb.cz/radecky`. Přednášková prezentace. Vysoká škola báňská - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky , Katedra informatiky.

[54] RICCI, Francesco. *Recommender systems handbook*. New York: Springer, 2011. ISBN 978-0-387-85820-3.

[55] RICCI, Francesco, Lior ROKACH a Bracha SHAPIRA (eds.). *Recommender Systems Handbook*. 2. vydání. New York: Springer, 2015. ISBN 978-1-4899-7636-9 (978-1-4899-7637-6). DOI 10.1007/978-1-4899-7637-6.

[56] SEGARAN, Toby, Colin EVANS a Jamie TAYLOR. *Programming the semantic web*. 1st ed. Cambridge: O'Reilly, c2009. ISBN 978-0-596-15381-6.

[57] SCHREIBER, Guus a Yves RAIMOND (eds.). RDF 1.1 Primer. *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 2014 [cit. 2016-03-14]. Dostupné z: `https://www.w3.org/TR/rdf11-primer/`

[58] SKLENÁK, Vilém. Sémantický web – 10 let poté. In: *INFORUM 2011: 17. konference o profesionálních informačních zdrojích* [online]. Praha: Albertina icome Praha, 2011, s. 10 [cit. 2016-03-07]. ISSN 1801-2213. Dostupné z: `http://www.inforum.cz/pdf/2011/sklenak-vilem.pdf`

[59] SMITKA, Jan. *Sémantický web v EEG/ERP doméně* [online]. Plzeň, 2013 [cit. 2016-03-07]. Dostupné z: `https://otik.uk.zcu.cz/bitstream/handle/11025/8688/Smitka_BPINI.pdf?sequence=1`. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Ing. Roman Mouček, Ph.D.

[60] STRNAD, Radek. *Využití preferencí zájemců při obchodování s nemovitostmi* [online]. Praha, 2014 [cit. 2016-03-16]. Dostupné z: `https://is.cuni.cz/webapps/zzp/download/120175616/?lang=cs`. Diplomová práce. Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, Katedra softwarového inženýrství. Vedoucí práce RNDr. Michal Kopecký, Ph.D.

[61] SU, Xiaoyuan a Taghi M. KHOSHGOFTAAR. A Survey of Collaborative Filtering Techniques. In: HONG, Jun (ed.). *Advances in Artificial Intelligence*. Boca Raton, Florida: Florida Atlantic University, Department of Computer Science and Engineering, 2009, **2009**, s. 1-19. DOI: 10.1155/2009/421425. ISSN 1687-7470. Article ID: 421425. Dostupné také z: `http://www.hindawi.com/journals/aai/2009/421425/`

[62] SVOBODOVÁ, Kristýna. *Topic Maps* [online]. Brno, 2014 [cit. 2016-03-10]. Dostupné z: `http://is.muni.cz/th/342388/ff_m/DP_-_Kristyna_Svobodova_-_342388.pdf`. Diplomová práce. Masarykova univerzita v Brně, Filozofická fakulta, Ústav české literatury a knihovnictví, Kabinet informačních studií a knihovnictví. Vedoucí práce Mgr. Tomáš Bouda.

[63] TAGHIPOUR, Nima, Ahmad KARDAN a Saeed Shiry GHIDARY. Usage-based web recommendations. In: *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07*. New York, New York, USA: ACM Press, 2007, s. 113-120. DOI: 10.1145/1297231.1297250. ISBN 9781595937308. Dostupné také z: `http://portal.acm.org/citation.cfm?doid=1297231.1297250`

[64] UEHARA, Ryuhei a Yushi UNO. Efficient Algorithms for the Longest Path Problem. In: FLEISCHER, Rudolf a Gerhard TRIPPEN (eds.). *Algorithms and Computation*. Hong Kong, China: Springer Berlin Heidelberg, 2004, s. 871-883. DOI: 10.1007/978-3-540-30551-4_74. ISBN 978-3-540-24131-7. ISSN 0302-9743. Dostupné také z: `http://link.springer.com/10.1007/978-3-540-30551-4_74`

[65] SACK, Harald, WAITELONIS, Jörg (ed.). Towards exploratory video search using linked data. In: FURHT, Borko. *Multimedia Tools and Applications*. New York: Springer US, 2012, **59**(2), s. 645-672. DOI: 10.1007/s11042-011-0733-1. ISSN 1380-7501. Dostupné také z: `http://link.springer.com/10.1007/s11042-011-0733-1`

[66] WANG, Laung-Terng, Yao-Wen CHANG a Kwang-Ting CHENG. *Electronic design automation: synthesis, verification, and test*. 1. vydání. Boston: Morgan Kaufmann/Elsevier, 2009. Morgan Kaufmann series in systems on silicon. ISBN 01-237-4364-8.

[67] WHITE, Ryen W. a Resa A. ROTH. Exploratory Search: Beyond the Query-Response Paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*. 2009, **1**(1), 1-98. DOI: 10.2200/S00174ED1V01Y200901ICR003. ISSN 1947-945x. Dostupné také z: http://www.morganclaypool.com/doi/abs/10.2200/S00174ED1V01Y200901ICR003

[68] WOOD, David (ed.). What's New in RDF 1.1. *World Wide Web Consortium (W3C)* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 2014 [cit. 2016-03-14]. Dostupné z: https://www.w3.org/TR/rdf11-new/

[69] The Web 3.0 Debate. *Chicago Startups | Built In Chicago: Chicago's hub for startups and tech* [online]. Chicago: Codal, 2015 [cit. 2016-03-08]. Dostupné z: http://www.builtinchicago.org/blog/web-30-debate

[70] Tutorial 3: Semantic Modeling. *Linked Data Tools Free Downloads Semantic Web* [online]. Cambridge, England (United Kingdom): LinkedDataTools.com, 2015 [cit. 2016-03-13]. Dostupné z: http://www.linkeddatatools.com/semantic-modeling

[71] *W3C Semantic Web Activity* [online]. Keio, Beihang: MIT Computer Science and Artificial Intelligence Laboratory | MIT CSAIL, ERCIM - The European Research Consortium for Informatics and Mathematics, 2013, 2013-06-19 [cit. 2016-03-09]. Dostupné z: https://www.w3.org/2001/sw/

[72] *Aemoo* [online]. [cit. 2016-04-09]. Dostupné z: http://wit.istc.cnr.it/aemoo/

Příloha **A**

Seznam použitých zkratek

API Application Programming Interface

BFS Breadth-first Search

DFS Depth-first Search

DTD Document Type Definition

FIFO Fisrt In, First Out

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

IQL inCloud Query Language

IRI Internationalized Resource Identifier

JSON JavaScript Object Notation

JSON-LD JavaScript Object Notation for Linked Data

KGRAM Knowledge Graph Abstract Machine

LED Lookup Explore Discovery

LOD Linked Open Data

LIFO Last In, First Out

NOT Not Only Tag

RDF Resource Description Framework

RDFa Resource Description Framework in Attributes

REST Representational State Transfer

RIF Rule Interchange Format

RS Recommender System

SPARQL SPARQL Protocol and RDF Query Language / Simple Protocol and RDF Query Language

SQL Structured Query Language

SWOC Semantic Wonder Cloud

URI Uniform Resource Identifier

URL Uniform Resource Locator

WWW World Wide Web

WYSIWYM What You See Is What You Mean

XSD XML Schema Definition

XML Extensible Markup Language

Příloha **B**

Obsah přiloženého CD

- `diplomova_prace` ..... adresář textu diplomové práce
  - `zdrojove_soubory` .. zdrojové soubory LaTeXu s přílohami textu práce
    - `DP_Chouň_Martin_2016.tex` ... hlavní zdrojový soubor textu práce
    - ...
  - `DP_Chouň_Martin_2016.pdf` ..... text práce v PDF
- `projekt` ..... adresář projektu aplikace
  - `GraphBasedRecommendationAlgorithmsForLinkedData` ..... adresář NetBeans 8.0.2 projektu se zdrojovými kódy a veškerými vyžadovanými soubory
    - ...
  - `dokumentace` ..... adresář dokumentace aplikace
    - `latex` ..... zdrojové soubory LaTeXu s přílohami dokumentace
      - `gbrafld-dokumentace.tex` ..... hlavní zdrojový soubor textu dokumentace
      - ...
    - `gbrafld-dokumentace.pdf` ..... text dokumentace v PDF
  - `GraphBasedRecommendationAlgorithmsForLinkedData.7z` .... archiv projektu aplikace ve formátu 7ZIP zabalený metodou LZMA2 ultra
- `obsah_CD.txt` ..... popis obsahu CD