

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Webový portál věnovaný syntaktické analýze přirozeného jazyka

BAKALÁŘSKÁ PRÁCE

Lukáš Macháček

Brno, jaro 2002

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: unknown

Shrnutí

S rostoucím množstvím vystavených vědeckých článků v síti Internet je stále větší problém vyhledat dokumenty z požadované vědecké oblasti. Tematicky specializované vyhledávací systémy provozované většinou na univerzitách nám mohou v takovéto situaci hodně pomoci. Tyto vyhledávače indexují ohromné množství článků a informací o jejich autorech.

Vedle vyhledávačů existují v síti Internet tzv. webové portály, které se snaží ulehčit přístup k žádaným dokumentům tak, že plní své informační zdroje výhradně dokumenty, které se zabývají určitou tematickou oblastí. K plnění svých vlastních informačních zdrojů využívají webové portály zpravidla právě činnosti tematicky specializovaných vyhledávačů.

V rámci projektu byl implementován systém, který získává texty článků zabývajících se daným tématem, informace o jejich vzájemných citacích a jejich autorech. Ke každému autorovi ještě také zjišťuje informace o jeho působišti. Tento systém je určen pro pozdější integraci do webového portálu věnovanému syntaktické analýze přirozeného jazyka.

Klíčová slova

webový portál, internetový vyhledávač, webový vyhledávač, ResearchIndex, HP Search

Obsah

I.	Webové portály	3
1.	Úvod	4
2.	Webový portál	5
2.1.	Tematická orientace	5
2.1.1.	Horizontální portály	5
2.1.2.	Vertikální portály	6
2.2.	Struktura	7
2.2.1.	Autonomní část	7
2.2.2.	Část závislá na chování uživatele	7
2.3.	Rozdělení z uživatelského pohledu	7
2.4.	Motivace	8
3.	Vyhledávání informací s vědeckým obsahem na Internetu	9
3.1.	ResearchIndex	9
3.1.1.	Vyhledávání dokumentů	10
3.1.2.	Analýza dokumentů	10
3.1.3.	Zpracování dotazů	10
3.2.	HP Search	11
3.2.1.	Vyhledávání	11
3.2.2.	Hodnocení nalezených stránek	12
3.2.3.	Aktualizace dat	12
3.3.	DBLP	12
3.4.	Mops	13
II.	Realizace webového portálu	14
4.	Analýza zadání a návrh systému	15
4.1.	Analýza	15
4.2.	Návrh vyhledávacího systému	16
4.2.1.	Sub-systém FindNew	16
4.2.2.	Sub-systém ArticleDownload	16
4.2.3.	Sub-systém AuthorSearch	17
4.3.	Datový model	17
4.3.1.	Tabulka article	18
4.3.2.	Tabulka citation	18
4.3.3.	Tabulka similar	19
4.3.4.	Tabulka author	19
4.3.5.	Tabulka article.author	19

4.3.6.	Tabulka affiliation	19
4.3.7.	Tabulka author_affiliation	20
5.	Implementace vyhledávacího systému	21
5.1.	Jednotlivé skripty	21
5.1.1.	FindNew	21
5.1.2.	ArticleDownload	22
5.1.3.	AuthorSearch	22
5.2.	Použité třídy	22
5.2.1.	MySql	22
5.2.2.	Log	23
5.2.3.	Clanek	23
5.2.4.	Autor	24
5.2.5.	Affiliation	24
5.2.6.	Chyba	24
6.	Výsledky a zhodnocení	25
6.1.	Test činnosti jednotlivých sub-systémů	25
6.1.1.	FindNew	25
6.1.2.	ArticleDownload	25
6.1.3.	AuthorSearch	26
6.2.	Shrnutí výsledků testování a návrh na další řešení	27
7.	Závěr	28
A.	Instalace	29
A.1.	Požadavky pro používání systému	29
A.2.	Adresářová struktura	29
A.3.	Vytvoření databáze	30
A.4.	Konfigurace systému	30

Seznam obrázků

3.1. Vznik indexu	11
4.1. Stavový diagram sub-systému FindNew	16
4.2. Stavový diagram sub-systému ArticleDownload	17
4.3. Stavový diagram sub-systému AuthorSearch	17
4.4. Výsledný ERA diagram	18

Část I.

Webové portály

Kapitola 1

Úvod

Množství informací obsažené v síti Internet expanduje den co den nezadržitelným tempem a pravděpodobnost, že se zde nachází přesně ta informace, kterou kupříkladu já potřebuji, je tak čím dál tím větší. Avšak stále velkým problémem výše zmiňované celosvětové sítě je organizace takového kvanta dat a následné vyhledávání v něm. Z tohoto problému vyplývá, že pouhé obsažení dané informace nemusí nutně znamenat, že se mi podaří k ní bez vynaložení velkého úsilí nebo vůbec dostat. Veškerá data jsou totiž distribuována na obrovském množství počítačů po celém světě a existuje naprostá volnost při jejich zveřejňování.

Tato tzv. informační anarchie a neexistence jakékoliv centrální správy dokumentů vedla ke vzniku spousty at' už univerzálních nebo specializovaných internetových vyhledávacích služeb. Mezi nejznámější univerzální webové vyhledávače patří například dnes velice úspěšný Google. Univerzální protože se snaží shromažďovat informace o všech datech na Internetu zveřejněných nezávisle na tom, jaké tematické oblasti se týkají. Ze specializovaných webových vyhledávačů lze uvést například ResearchIndex, který se zaměřil na vyhledávání a zároveň rozšiřování vědecké literatury.

Ve snaze ulehčit orientaci jak ve vyhledávaných informacích ale vůbec v samotné celosvětové síti, vnést do práce uživatele s Internetem i jistý komfort a získat tak širokou obec uživatelů, začaly se budovat tzv. „webové portály“. Vznikly buď jako součást výše zmiňovaných vyhledávačů nebo úplně samostatně.

Cílem projektu je implementovat systém, který bude vyhledávat za pomoci specializovaných internetových vyhledávačů články, informace o jejich autorech a působištích těchto autorů a následně získaná data ukládat. Tento systém bude využit k plnění informačních zdrojů pro webový portál věnovaný konkrétně syntaktické analýze přirozeného jazyka.

Kapitola 2

Webový portál

Webový portál by se dal definovat jako vstupní bod - výchozí místo pro surfování po Webu nabízející širokou škálu služeb a informací s možností jejich přizpůsobení uživateli podle osobních potřeb a zájmů. Nabídka obecně zahrnuje vyhledávání webových zdrojů, jak prostřednictvím vyhledávacího stroje, tak prostřednictvím předmětového katalogu, prohledávání webového prostoru v určité zeměpisné, jazykové nebo tematické oblasti.

Dále nabízí ještě spoustu dalších služeb, které se u jednotlivých portálů už ovšem liší podle toho, na jakou skupinu uživatelů jsou zaměřeny a jaká činnost je s touto skupinou lidí spjata.

2.1. Tematická orientace

Možností, jak rozdělit zaměření webových portálů, je samozřejmě více, ale jedním z hlavních faktorů, který určuje, jak bude daný komunikační systém vypadat, je tematická orientace. Podle tohoto kritéria lze webové portály rozdělit do dvou skupin:

- horizontální portály (horizontal portals)
- vertikální portály (vertical portals, vortals)

2.1.1. Horizontální portály

Horizontální portály jsou charakteristické tématicky širokým (obecným) zaměřením. Název horizontální je tedy odvozen od toho, že jejich činnost se soustředí na co nejširší informační spektrum. Obecnost je dána především nutností uspokojit co největší masu uživatelů a tedy komerčními důvody.

Naprostá většina těchto portálů se na Webu objevila původně jen jako jeden ze základních typů vyhledávacích nástrojů umožňujících orientaci v informačních zdrojích na Internetu. Časem se z nich staly nejnavštěvovanější webové stránky, a tak se jejich provozovatelé pokusili využít popularity a rozšířili nabídku o další funkce s cílem udržet uživatele na svých stránkách i poté, co si vyhledali potřebné informace. Firmy provozující vyhledávací systémy začaly nakupovat další servery, buď kvůli zajímavým technologiím nebo kvůli atraktivnímu obsahu. Vznikla tak opravdu „gigantická sídla“ s širokým tematickým rozsahem tak, jak je známe dnes.

Do palety poskytovaných služeb patří denní zpravodajství, tematicky orientované kanály (uživatelé se podle svých osobních zájmů mohou přihlásit k jejich odběru), ekonomické informace a burzovní zprávy, chat, přehled počasí, mapy, horoskopy, vtipy, kalendáře, vyhledávání osob, e-mailových adres a telefonních čísel (tzv. white pages), vyhledávání firem (tzv. yellow pages), hry, elektronické obchody, free-mailové služby, bezplatný prostor pro publikování webových stránek, zpřístupnění důležitých informací prostřednictvím mobilních telefonů a kapesních počítačů apod. Naprostá většina nabízených služeb je pro koncového uživatele bezplatná díky ziskům z všudypřítomné reklamy.

Jako příklad takových horizontálních webových portálů lze krátce uvést Google, Yahoo!, Lycos, NBCi a z domácích potom Seznam (jeden z prvních u nás), Centrum, Atlas, Volny a mnoho dalších.

2.1.2. Vertikální portály

Vertikální portály jsou na rozdíl od horizontálních tematicky úzce zaměřené a jsou určeny pro specifický typ uživatelů, proto se také někdy nazývají „community portals“.

Zpravidla jsou nekomerční nebo také neziskové. Patří mezi ně především vědecky orientované informační systémy, jejich provoz zajišťují univerzity nebo jiná vzdělávací a výzkumná centra. Shromažďují články, texty publikací, dále informace o jejich autorech a případně i o působišti těchto lidí, dále pak výsledky výzkumů, záznamy z konferencí atd.

Dnes se pomalu ukazuje, že na rozdíl od v současné době hojně rozšířených horizontálních portálů, které v mnoha případech nesplnily očekávání svých provozovatelů, mají vertikální portály podle analytiků budoucnost mnohem zářivější i při komerčním využití. Například v oblasti zahraničního obchodu je ideální sdružit více subjektů z určitého výrobního odvětví v rámci jednoho projektu a zájemcům nabízet související zboží a služby pod jednou střechou.

Nahlédneme-li do menu nabízených služeb, zjistíme, že v porovnání s komerčními portály je nabídka podstatně chudší, avšak postačující pro účely, k nimž byl portál zbudován. Většina služeb se odvíjí od témat, jimž se systém věnuje. Například z uložených a získaných dokumentů lze automaticky generovat záznamy ve formátu BibTeX, zjišťovat informace o citacích v člancích, případně jejich podobnosti a příbuznosti, jejich převody do jiných formátů nebo přidávání komentářů a zakládání diskusí či diskusních skupin nebo konferencí na téma, kterému se daný informační zdroj věnuje.

Do skupiny vertikálních portálů patří například CNET¹ zaměřený na oblast technologií nebo Blackboard² a WebCT³ věnované vzdělávání.

¹<http://www.cnet.com>

²<http://www.blackboard.com>

³<http://www.webct.com>

2.2. Struktura

Webové portály se většinou skládají z několika systémů, které pracují nezávisle na sobě. Dají se rozdělit do dvou skupin. Do první patří ty, co pracují autonomně a pro uživatele je jejich činnost skryta. Říkejme jí třeba autonomní část webového portálu. Druhou skupinou jsou naopak aplikace, kde výsledek o jejich činnosti je nabídnut uživateli a sama činnost je také závislá na uživatelském vstupu.

2.2.1. Autonomní část

Autonomní část se stará o prohledávání Internetu a obstarává data z předem daných informačních zdrojů. K této činnosti většina portálů využívá tzv. robotů (též spiders, crawlers), které automaticky procházejí hypertextovou strukturu Webu, načítají dokumenty a rekurzivně procházejí v nich obsažené odkazy. Do této skupiny také patří programy, jenž získaná data následně analyzují a podle potřeb připravují pro následné zpracovávání samotným webovým rozhraním. Například vytvářejí indexy nebo ukládají potřebné informace do databází.

2.2.2. Část závislá na chování uživatele

Část závislá na chování uživatele se sestává z aplikací, které reagují na dotazy uživatele a jejich výstup není uživateli na rozdíl od autonomní části skryt. Komunikace probíhá pomocí webového rozhraní. Podle požadavků od uživatele, operuje systém nad předem získanými daty a umožňuje do nich podle potřeby zasahovat.

2.3. Rozdělení z uživatelského pohledu

Webový portál se z pohledu uživatelského zpravidla skládá z několika vrstev, kde každá nabízí každé skupině uživatelů jiné služby.

Základní vrstva je vždy administrativní nebo také vývojová. Je určena pro lidi starající se o správný chod celého webového portálu. Takže je možné zde ovlivňovat například i chod autonomně pracujících systémů. Proto je pro přístup vyžadována autentifikace.

Další vrstva už je pro běžné uživatele, kterým je portál určen. Nabízí tedy služby, kterými by neměl být narušen správný chod celého systému. V případě, že jsou běžní uživatelé rozlišováni do dalších skupin, lze tuto vrstvu rozčlenit na další právě podle těchto skupin a každé pak přidělit různá práva s ohledem na to k jakým službám a informacím mohou přistupovat. Například u zpravodajského portálu bude skupina redaktorů, kteří mohou vkládat do systému své články a další skupinu budou tvořit čtenáři, kteří již pravomoc pro vkládání nových článků nemají. Jak je vidět, každý má pak různá práva pro zacházení s informacemi a pro pohyb uvnitř portálu. Tato uživatelská vrstva by se tedy rozdělila na redaktorskou a čtenářskou. Je to ovšem pouze ilustrační příklad a v praxi by konkrétně u tohoto portálu bylo rozdělení ještě mnohem složitější.

2.4. Motivace

Jak je z předchozího textu patrné, usnadňují webové portály orientaci v nepřehledné pavučině informací a služeb dostupných na Internetu. Zároveň s tím vnáší do naší činnosti jistý komfort a pohodlí, ať už se jedná o práci nebo zábavu. Mám-li svůj oblíbený portál, nemusím znát neúnosné množství adres webových stránek, ale stačí mi jen jedna a přitom se dostanu k informacím, které mě zajímají, snadno a rychle.

Kapitola 3

Vyhledávání informací s vědeckých obsahem na Internetu

Vědci z mnoha výzkumných oblastí dnes publikují své nejnovější objevy elektronicky na svých webových stránkách dlouho před tím, než se objeví na konferencích a v časopisech. Zatímco se před desetiletími dalo k výsledkům takovýchto konferencí a časopisům týkajících se právě těchto nových objevů dostat jen v knihovnách daných výzkumných center, je dnes situace poněkud jiná. Internet nám otevírá nové možnosti, jak se k takovýmto informačním zdrojům dostat. Tradiční vyhledávací systémy nám v tom ovšem moc nepomohou, neboť nejsou schopny prohledávat kupříkladu postscriptové soubory (výjimkou je Google), což je právě formát vhodný k elektronické přípravě vědeckých dokumentů a jejich publikování v již zmiňované síti Internet.

Naštěstí existují na Webu specializované služby, které si kladou za cíl usnadnit vyhledávání informací z vědecké oblasti. Tyto vyhledávací stroje fungují převážně při univerzitách jako nástroj vědy a výzkumu jen zřídka se jedná o produkty komerčních softwarových firem.

3.1. ResearchIndex

ResearchIndex (formálně CiteSteer¹) je autonomně pracující systém určený k vyhledávání a zároveň rozšiřování vědecké literatury. Vyvíjejí jej Steve Lawrence, Kurt Bollacker a C. Lee Giles v NEC Research Institute.

Hlavní oblastí, kterou se tento systém zabývá, jsou informační technologie. Mezi další už ne tak významné patří pokročilá nauka o materiálech, chemie, fyzika, biofyzika a bioinformatika. Celkem tak indexuje něco přes 500000 vydaných článků, prezentací, záznamy konferencí a technických zpráv (údaj z května 2002).

ResearchIndex zpracovává informace o citacích, které je následně možné využít například k sledování vývoje určitých témat v závislosti na čase nebo různé návaznosti dokumentů. Zabraňuje tak i zbytečnému plýtvání prostředků, když se vědci v různých výzkumných centrech zabývají tímž problémem. Hlavním nástrojem systému je autonomní vyhledávání a indexace citací (Autonomous Citation Indexing, ACI).

¹<http://citeseer.nj.nec.com/cs>

3.1.1. Vyhledávání dokumentů

Systém ACI vyhledává články prohledáváním Webu, přímo na domovských stránkách autorů nebo monitorováním diskusních skupin. Také je možné poskytnout systému dokumenty „ručně“. Při prohledávání Webu se využívá běžných vyhledávačů jako AltaVista, HotBot nebo Excite pro určení vhodných startovních bodů pro robota, který se při svém pohybu po síti zajímá pouze o soubory ve formátech Postscript a PDF. Zpracovávány jsou výhradně dokumenty obsahující reference nebo bibliografickou část, jinak nejsou považovány za vědecký článek.

3.1.2. Analýza dokumentů

V získaných dokumentech jsou pomocí heuristik ACI extrahovány jednotlivé citace a následně analyzovány. Hledají se pole název, autor, rok vydání, počet stran a identifikátor citace. S využitím regulárních výrazů lze rozlišit odchylky, kupříkladu je-li uveden seznam autorů u citace nebo pouze první z tohoto seznamu.

Pomocí metody *invariants first* se systém pokouší převést citace do uniformní podoby. To znamená, že záznamy, které mají relativně odlišnou syntaxi a pozici v dokumentu, jsou zpracovávány nakonec. Jako pomocný nástroj využívá k identifikování jednotlivých oblastí databázi jmen autorů, článků atp.

Při analýze dokumentu se může stát, že se stejné citace vyskytnou vícekrát na různých místech a ukládaly by se tak duplicitní záznamy, čemuž se musí předejít. K rozeznání a seskupení identických citací používá systém ACI následující čtyři skupiny metod:

- poměrování vzdálenosti mezi řetězci s využitím algoritmu LikeIt od Peter N. Yianilos
- poměrování četnosti slov založené na statistice výskytu slov běžných pro každý řetězec
- znalosti o složkách a struktuře dat (u citací například jméno autora, název článku, rok publikování atd.)
- pravděpodobnostní modely využívající známé bibliografické informace pro identifikaci slov ve struktuře citací

Normalizační algoritmy použité v systému v současné době jsou dostačující pro praktické využití, avšak autoři uvažují o možném zlepšení použitím metod strojového učení.

3.1.3. Zpracování dotazů

ResearchIndex může vracet seznam citací nebo indexovaných dokumentů odpovídajících zadanému dotazu pomocí klíčových slov. Seznamy se pak nadále dají procházet a přes odkazy lze zobrazit o dokumentech a citacích podrobnější informace. Mezi ty patří například seznam dokumentů podobných nejen podle textu ale i na úrovni vět nebo seznam citovaných a současně i citujících článků. Pro citace je uveden také graf zachycující počet

citací v člancích v daném roce. Mezi poskytnutými informacemi jsou mimo jiné také adresy ke zdroji, odkud daný dokument pochází, a na soubory daného dokumentu uložené v jiných formátech přímo na straně vyhledávacího systému.

3.2. HP Search

HP Search² je tematicky zaměřený informační systém pro vyhledávání a sledování osobních webových stránek uznávaných vědců, implementovaný v jazyce Java. Domovské stránky zaujímají důležitou pozici ve vědecké komunikaci, neboť obsahují důležitá data jako například kontaktní informace, zprávy o činnosti, popisy projektů a v neposlední řadě i texty dokumentů.

3.2.1. Vyhledávání

HP Search se specializuje na vyhledávání v relativně malé výzkumné oblasti, což dovoluje nacházet stále nové a nové dokumenty. Jak ukazuje obrázek, dá se způsob budování vyhledávacího indexu rozdělit do tří kroků.



Obrázek 3.1.: Vznik indexu

- V prvním kroku se získá seznam jmen lidí, kteří jsou stále aktivní ve výzkumu a vědecké činnosti. Jedna z cest, jak tato data získat, je prohledávat obsah výsledných zpráv z vědeckých konferencí nebo v časopisech a publikacích. Druhým zdrojem je pak DBLP server podrobněji popsany dále. Kvalita výsledného indexu nejvíce závisí na takto získaném seznamu jmen vědců, a proto jsou bráni v potaz jen uznání členové. Tato informace se získá z elektronických bibliografií, kde je pod daným jménem údaj „certified“.
- V kroku druhém se za pomoci běžných vyhledávačů jako Alta Vista, Fast, Google, Excite, Hotbot, Infoseek a Northern Light získají podle údajů z již připraveného seznamu jmen domovské stránky. Ty jsou ohodnoceny a výsledek se uloží.
- V posledním kroku se použijí nalezené domovské stránky s nejvyšším ohodnocením jako startovní bod pro speciální vyhledávací nástroj zvaný Mops, jehož výstupem je pak konečný vyhledávací index.

²<http://hpsearch.uni-trier.de>

3.2.2. Hodnocení nalezených stránek

K ohodnocení domovských stránek HP Search nepoužívá klasické metody jako rozhodovací stromy, ale vlastní algoritmus, protože posuzuje i strukturu URL, odkazy vedoucí k dokumentu z jiných stránek a při prvním kroku vyhledávání není ještě text, nad nímž by se daly tyto metody použít, dostupný.

V prvním kroku HP Search zhodnotí získané údaje jako název stránky, URL, popis a pozici, na které vyhledávací systém adresu uvedl. Poté vybere kandidáta s nejvyšším skóre a pro něho vypočítá konečné hodnocení, k jehož stanovení použije hlavičku, odkazy, meta-tagy a samotný text dokumentu.

3.2.3. Aktualizace dat

Aktuálnost dat nabízených systémem HP Search zajišťují dva parametricky říditelné mechanismy, které pracují v pravidelných časových intervalech a případně jsou i volány specifickými událostmi.

- Vyhodnocování záznamů přístupu k DBLP
- Ověřování platnosti URL

Při vyhodnocování přístupových záznamů k DBLP se zjišťuje, zda někdo nezadal nové jméno, které ještě není v databázi, a nebo se prověřují jména, která se vyhledávala před delším časovým úsekem, než je nastavené maximum, a mohlo by to tak znamenat zastaralost záznamu. Tento proces probíhá denně.

Platnost URL se ověřuje jednou týdně pro každé jméno vědce z databáze pouze pro nejvíce ohodnocené stránky. Pokud je zadaná adresa neplatná, kontroluje se ještě minimálně dvakrát, než je definitivně odstraněna z databáze.

3.3. DBLP

Server DBLP³ (zpočátku DataBase systems and Logic Programing nyní Digital Bibliography & Library Project) poskytuje bibliografické informace o výsledcích hlavních konferencí a publikacích týkajících se počítačové vědy. Na tomto serveru je indexováno více než 270000 článků a k tomu obsahuje i zhruba 100000 odkazů na domácí stránky vědců zabývajících se informatikou (údaje z května 2002). Indexovány jsou články publikované nejen v elektronické formě, ale i ty, jež vyšly pouze v tištěné podobě.

³<http://www.informatik.uni-trier.de/~ley/db/index.html>

3.4. Mops

Mops⁴ (Martin's Online Paper Search) je relativně jednoduchý vyhledávací systém, sestávající se ze skupiny skriptů v Perlu. Běží zhruba 30 hodin týdně a to většinou přes víkend. Má za cíl poskytovat index z vědecky zaměřených dokumentů. Vstupem tohoto nástroje je seznam adres, které používá jako startovní body pro svou další činnost.

Za úkol má hledat komprimované nebo přímo ps, dvi a pdf soubory, neboť, jak už jsem se v některé z předcházejících částí zmínil, vědecké dokumenty jsou ukládány zejména v těchto formátech. Ze startovní stránky pak následuje odkazy do hloubky 1 nebo 2, kde opět hledá odkazy na požadované soubory. Prohledávání takto malé oblasti, kolem předané adresy, má za následek poměrně vysokou rychlost.

Nalezené dokumenty ukládá v ASCII formátu, což následně umožňuje snadné vytvoření indexu. Dále se ke každému ještě uchovává datum jeho nalezení, posloupnost URL, které k němu vedly a jméno odkazu, které uvedl autor stránky jako popis. Ke stejným dokumentům ovšem mohou vést různé cesty, kupříkladu server, kde jsou uloženy může být dosažitelný pod více než jedním jménem. Mops se snaží takovéto duplicity odstranit a zabránit jejich výskytu ve výsledném indexu.

⁴<http://mops.uni-trier.de>

Část II.

Realizace webového portálu

Kapitola 4

Analýza zadání a návrh systému

Jako první krok, který jsem učinil při práci na projektu, byla analýza zadání a návržení možného řešení.

4.1. Analýza

V první kapitole jsem se zabýval obecně rozdělením, strukturou a činností webových portálů. Webový portál věnovaný syntaktické analýze jazyka není nic výjimečného, co by se vymykalo tomuto obecnému popisu.

Jak je již z názvu patrné, jedná se o systém zaměřený na tematicky úzkou oblast, konkrétně syntaktickou analýzu přirozeného jazyka. Spadá tedy do skupiny vertikálních portálů. Bude provozován pro vědecké a výzkumné účely, nebylo tedy potřeba zabývat se implementací služeb běžných pro komerční webové portály jako například free-mail, chat, horoskopy, vtipy atd. Naopak bylo potřeba zabývat se návrhem služeb využitelných právě k vědeckým účelům, jako například indexace článků, zpráv z výzkumu, publikací, správa seznamu pracovníků atd.

Základní struktura systému se neméně liší od jiných webových portálů a byla tudíž také potřeba navrhnout a následně implementovat následující části:

- Autonomní vyhledávací systém, který bude na Internetu v zadané tematické oblasti pomocí seznamu klíčových slov, získávat a aktualizovat následující data:
 - dokumenty obsahující texty článků a informace o nich
 - jména vědců a vědeckých pracovníků zabývajících se danou tematikou a adresy jejich domovských stránek a stránek obsahujících seznamy jejich publikací
 - informace o vědeckých pracovištích a výzkumných centrech jako působištích daných vědců a vědeckých pracovníků
- Samotné webové rozhraní, které bude zpřístupňovat zhruba tyto hlavní služby:
 - procházení seznamem indexovaných článků, autorů a také působišť a samozřejmě možnost zobrazení jejich základních informací
 - fulltextové prohledávání v uložených dokumentech
 - dodatečné přidávání komentářů k článkům

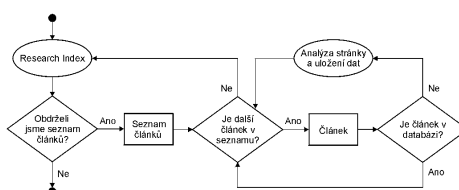
- zobrazení seznamu citací a podobností u článků
- u každého autora možnost upravit či doplnit údaje jako URL domácí stránky, URL stránky obsahující seznam publikací (pokud se takový seznam na zadané osobní stránce nenachází)

Z analýzy vyplynulo, že je potřeba navrhnout a implementovat dva nezávisle pracující sub-systémy. V rámci tohoto projektu bylo mým úkolem implementovat pouze první z nich. Proto se budu dále zabývat návrhem a postupem při implementaci pouze prvního systému a to autonomního v uvozovkách robota starajícího se o vyhledávání a následné získávání relevantních vědeckých dokumentů a dalších s nimi souvisejících informací.

4.2. Návrh vyhledávacího systému

Mezi základními požadavky na vyhledávací nástroj byla snadná ovladatelnost, průhlednost a možnost kontrolovat jeho činnost. Z těchto důvodů jsem se rozhodl sestavit vyhledávací systém ze tří nezávislých sub-systémů, kde se každý bude starat o své pole působnosti. Jednotlivé sub-systémy popisují následující stavové diagramy.

4.2.1. Sub-systém FindNew

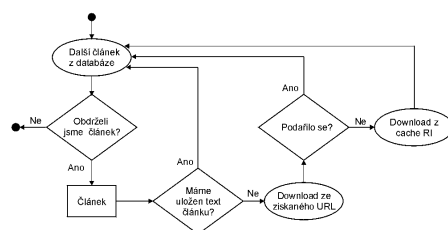


Obrázek 4.1.: Stavový diagram sub-systému FindNew

Aplikace FindNew si nejprve v prvním (startovním) bodě vyžádá od ResearchIndexu pod zadanými klíčovými slovy seznam článků. V jednom dotazu jich ResearchIndex nabízí maximálně 50. Pro každý článek zjistí, zda je již indexován. Pokud ne podívá se na stránku obsahující informace o daném článku, analyzuje ji a získané údaje uloží do databáze. Tento krok opakuje pro všechny články v seznamu dokud nedojde na konec. Poté se vrací do počátečního bodu a vše opakuje pro další obdržení seznam. Činnost tohoto sub-systému končí v případě, že již ResearchIndex nenabízí žádný článek.

4.2.2. Sub-systém ArticleDownload

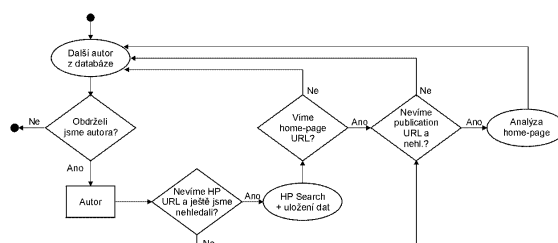
V prvním kroku vezme sub-systém ArticleDownload z databáze informace o článku a podívá se, zda už je uložen ps nebo pdf soubor pod id daného článku. V případě že není, pokusí se získat požadovaný soubor v patřičném formátu z adresy uložené v databázi,



Obrázek 4.2.: Stavový diagram sub-systému ArticleDownload

kam ji uložil předchozí skript v průběhu své činnosti. Pokud se mu nepodaří soubor uložit, podívá se do vyrovnávací paměti na straně serveru ResearchIndexu a snaží se dokument získat tam. Aplikace opakuje celý děj vždy od prvního kroku pro každý článek až do doby, kdy už není na řadě v databázi žádný článek a ukončí proto svou činnost.

4.2.3. Sub-systém AuthorSearch

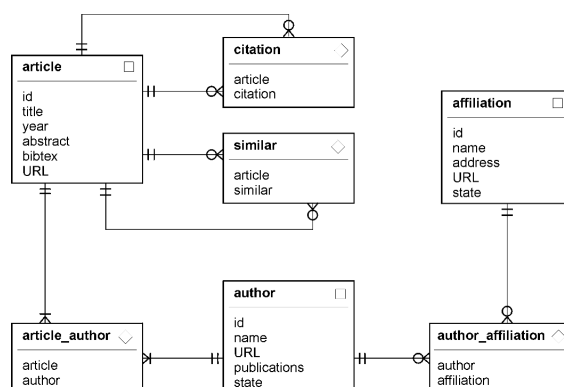


Obrázek 4.3.: Stavový diagram sub-systému AuthorSearch

V počátečním bodě si sub-systém AuthorSearch vyžádá z databáze informace o autorovi. V případě, že není známo URL a přitom jsme ho ještě nehledali, pokusí se vyhledat s využitím služeb vyhledávacího systému HP Search adresu domovské stránky. Pokud ani po tomto kroku stále neznáme požadované URL, vrátí se zpět do počátečního bodu. Jinak analyzuje domovskou stránku a snaží se najít odkaz na seznam publikací, není-li takový seznam součástí dané stránky. Zároveň se také pokouší najít nějaké základní informace o působišti, jako třeba název nebo adresa. Dále pokračuje stejným způsobem opět od počátečního kroku dokud není seznam autorů vyčerpán.

4.3. Datový model

Abych zajistil správnou funkčnost systému, navrhl jsem datový model. Nejprve jsem identifikoval jednotlivé entity a určil vazby mezi nimi (tzv. mřížku). Po sestavení modelu univerzální relace jsem doplnil kardinality a parciality. Výsledek ukazuje diagram na následujícím obrázku a pod ním popis jednotlivých entit a atributů.



Obrázek 4.4.: Výsledný ERA diagram

V tomto konkrétním případě lze jednotlivé entity reprezentovat tabulkou v relační databázi a jejich atributy pak jako jednotlivé sloupce dané tabulky.

4.3.1. Tabulka article

Tabulka article je určena k ukládání následujících informací o jednotlivých člancích:

id jednoznačný identifikátor daného článku

title název daného článku

year rok vydání daného článku

abstract shrnutí daného článku

bibtex bibliografický záznam o daném článku

URL adresa odkud byl nebo teprve bude stažen ps nebo pdf soubor, který obsahuje samotný text daného článku.

4.3.2. Tabulka citation

Tabulka citation slouží jako vazební tabulka mezi různými řádky tabulky article. Tento vztah popisují následující sloupce:

article id článku, který je citován článkem, který má id shodné s citation.

citation id článku, který cituje článek, který má id shodné s article.

4.3.3. Tabulka similar

Tabulka similar slouží jako vazební tabulka mezi různými řádky tabulky article. Tento vztah popisují následující sloupce:

article id článku, který je podobný s článkem, který má id shodné se similar.

similar id článku, který je podobný s článkem, který má id shodné s article.

4.3.4. Tabulka author

Tabulka author je určena k ukládání následujících informací o jednotlivých autorech:

id jednoznačný identifikátor daného autora

name jméno daného autora

URL adresa domovské stránky daného autora

publications adresa stránky obsahující seznam publikací

state stavový atribut.

4.3.5. Tabulka article.author

Tabulka article.author slouží jako vazební tabulka mezi různými řádky tabulky article a řádky tabulky author. Tento vztah popisují následující sloupce:

article identifikátor řádku v tabulce article

author identifikátor řádku v tabulce author.

4.3.6. Tabulka affiliation

Tabulka affiliation je určena k ukládání následujících informací o jednotlivých působištích:

id jednoznačný identifikátor daného působiště

name název daného působiště

address adresa daného působiště

URL webová stránka daného působiště

state stavový atribut.

4.3.7. Tabulka author_affiliation

Tabulka author_affiliation slouží jako vazební tabulka mezi různými řádky tabulky author a řádky tabulky affiliation. Tento vztah popisují následující sloupce:

author identifikátor řádku v tabulce author

affiliation identifikátor řádku v tabulce affiliation.

Kapitola 5

Implementace vyhledávacího systému

Po vytvoření prvních návrhů stavových diagramů a datového modelu jsem se začal zabývat samotnou implementací vyhledávacího systému. Zjistil jsem, že bude nutné analyzovat webové stránky a že k tomu je vhodné využít regulárních výrazů. Jelikož už jsem měl drobné zkušenosti se skriptovacím jazykem PHP, který také umožňuje pracovat právě s regulárními výrazy, zvolil jsem ho jako prostředek k implementaci samotného vyhledávacího systému.

Pro úplnost dodávám, že všechny skripty, které jsou součástí systému, lze nalézt na internetové adrese http://nlp.fi.muni.cz/projekty/parsing_portal.

5.1. Jednotlivé skripty

5.1.1. FindNew

Skript FindNew vyhledává nové články a s nimi související informace. K tomu využívá služeb vyhledávacího systému ResearchIndex popsaného podrobněji ve třetí kapitole.

Při implementaci tohoto skriptu jsem narazil na velice závažný problém. Jednalo se, v tomto případě, o nemilé omezení skriptovacího jazyka PHP týkající se maximálního času určeného pro běh skriptu. Skript totiž nikdy nestihl během této doby získat a analyzovat všechny informace nabídnuté ResearchIndexem a následně je uložit. Nejvíce časově náročné bylo vždy stáhnutí požadovaného dokumentu z Internetu. Problém se mi podařilo naštěstí úspěšně vyřešit.

Zvolil jsem takový postup, při kterém se v jednom volání skriptu analyzuje omezený počet stránek získaných od ResearchIndexu. Po ukončení činnosti zavolá skript sám sebe s jinými parametry odkazujícími na další skupinu článků. Aby bylo ale možné odeslat hlavičky určené k opětovnému spuštění skriptu, nemohl jsem vypisovat na výstup žádný text a informovat tak uživatele o činnosti, kterou skript vykonává. To mě vedlo k vytvoření třídy Log, která zaznamenává potřebné údaje o činnosti do speciálního souboru. Jak se ukázalo později, bylo toto řešení potřebné i v dalších skriptech, kde nastávaly obdobné potíže. Po ukončení činnosti skriptu se vypíše ze záznamového souboru na stránku nej důležitější údaje jako například, o kterých člancích byly informace uloženy do databáze a o kterých již máme záznam, jak dlouho skript pracoval a případně kde vznikla jaká chyba.

5.1.2. ArticleDownload

Skript ArticleDownload vyhledává postupně pro každý článek PostScriptový nebo PDF soubor s jeho přesným zněním. Nejprve hledá na adrese získané od ResearchIndexu. Pokud se mu podaří daný soubor najít stáhne ho a uloží na disk. V případě že se mu to nepodaří, hledá ještě ve vyrovnávací paměti přímo na serveru ResearchIndexu.

5.1.3. AuthorSearch

Skript AuthorSearch dohledává chybějící informace o autorech článku. Mezi tyto chybějící informace může patřit URL domovské stránky nebo URL seznamu publikací a název působiště.

Rozpoznat správně na domácích stránkách autorů název jejich působiště je opravdu veliký problém. Každá stránka je jinak koncipována a najít a implementovat tak nějaký obecný mechanismus, podle kterého by se dalo postupovat při analýze daných stránek, je již nad rámec tohoto projektu. Mohlo by to být kupříkladu vhodné téma pro další bakalářský projekt.

Prozatím pracuje skript pouze s jednoduchou technikou, kdy zjišťuje informace o působišti pomocí URL domácí stránky autora. Vezme základ URL jako adresu působiště daného autora a tam se případně snaží najít záznamy obsahující název a fyzickou adresu daného působiště.

5.2. Použité třídy

Pro zpřehlednění výsledného systému a usnadnění práce možností použít stejný kód ve více skriptech jsem se rozhodl využít technik objektově orientovaného programování. Tato technika víceméně věrně napodobuje způsob, jakým zacházíme s předměty reálného světa, a proto mi také umožnila lépe nahlížet na řešené problémy. Další vlastností, která mě k této volbě vedla, bylo zapouzdření objektů. Rázem mi tak ve skriptovacím jazyce PHP, kterého jsem pro implementaci použil, odpadla nepohodlná práce s globálními proměnnými a celkově mi to usnadnilo práci s daty.

V následujícím výčtu se pokusím charakterizovat všechny třídy, které jsem postupně implementoval pro využití v hlavních skriptech.

5.2.1. MySql

Třída MySql je navržena tak, aby zjednodušila komunikaci s MySQL serverem. Zavoláním konstruktoru s příslušnými parametry se vytvoří perzistentní spojení s MySQL serverem. Vzniklé chyby se vždy zpracovávají pomocí instance třídy Chyba, na kterou se předal odkaz při volání konstrukturu. Následuje výčet nejdůležitějších metod:

Query Proveďte dotaz nad předem specifikovanou databází.

GetData Výstupem je asociativní pole obsahující všechna data předaná MySQL serverem po položení dotazu.

GetLastId Vrací hodnotu identifikátoru naposled vloženého záznamu do tabulky se sloupcem typu *auto_increment*.

5.2.2. Log

Třída Log je určena ke správě záznamů o činnosti (tzv. log souborů) jednotlivých skriptů. K vytvoření jednotky obsahující tuto třídu mě vedla nutnost umožnit navázání činnosti skriptu v bodě posledního přerušení po občasných výpadcích vyhledávacích serverů, například v případech kdy byly přetíženy. Následuje výčet nejdůležitějších metod:

Start Zjistí, zda při předchozím spuštění skriptu nedošlo k přerušení. Pokud ano najde řádek s popisem chyby a identifikátor posledního kroku, který se dá poté použít pro navázání přerušené činnosti. Proto se volá se na začátku skriptu.

Finish Volá se při úspěšném ukončení skriptu, aby se zapsala do záznamového souboru patřičná informace a aby se při opětovném spuštění skriptu začalo opět od začátku.

AnalyzeResult Přečte záznamový soubor a analyzuje ho. Získané informace uloží do pole seznam, kde se již mohou dále zpracovávat.

SaveRecord Uloží do záznamového souboru informace v patřičném formátu, aby je bylo možné následně analyzovat.

5.2.3. Clanek

Třída Clanek je jedna z klíčových. Slouží jako kontejner pro data popisující daný článek. Jsou zde definovány veškeré operace nad entitou *article*. Podrobněji jsou popsány v následujícím výčtu nejdůležitějších metod:

Parse Analyzuje HTML stránku obsahující informace o článku, jehož jasný identifikátor je zadán při volání konstruktoru této třídy.

CompleteAuthors Zjistí pro každého autora článku, zda se nachází v databázi. V případě, že nenachází, uloží údaje, které zná (jméno získané z informací o článku) do příslušných tabulek.

Find Zjistí, zda je daný článek uložen v databázi.

Load Načte potřebné informace o článku z databáze.

Save Uloží všechny informace o článku do databáze.

Update Aktualizuje všechny informace o článku v databázi.

CheckFile Zjistí, zda je uloženo pro daný článek jeho znění ve formě ps nebo pdf souboru.

SaveFile Snaží se z Webu získat ps nebo pdf soubor se zněním daného článku. Nejprve hledá pod adresou získanou z ResearchIndexu jako původní zdroj. Pokud se to z nějakého důvodu nepodaří, má za úkol, zkusit štěstí přímo ve vyrovnávací paměti na serveru ResearchIndexu.

5.2.4. Autor

Třída *Autor* slouží opět jako kontejner, tentokrát však pro data popisující daného autora. Jsou zde definovány veškeré operace nad entitou *author*. Podrobněji jsou popsány v následujícím výčtu nejdůležitějších metod:

Parse Analyzuje domácí stránku daného autora a snaží se na ní objevit odkaz na stránku obsahující seznam publikací a informace o jeho působišti.

CompleteAffiliations Zjistí, zda již existuje záznam o působišti daného autora v databázi. V případě, že neexistuje, uloží informace, které zná do databáze.

Find Zjistí, zda je daný autor uložen v databázi.

Load Načte potřebné informace o daném autorovi z databáze.

Save Uloží všechny informace o daném autorovi do databáze.

Update Aktualizuje všechny informace o daném autorovi v databázi.

5.2.5. Affiliation

Třída *Affiliation* slouží jako kontejner pro data popisující dané působiště. Jsou zde definovány veškeré operace nad entitou *affiliation*. Podrobněji jsou popsány v následujícím výčtu nejdůležitějších metod:

Parse Analyzuje domácí stránku daného působiště a snaží se na ní objevit název působiště a jeho adresu.

Find Zjistí, zda je dané působiště uloženo v databázi.

Load Načte potřebné informace o daném působišti z databáze.

Save Uloží všechny informace o daném působišti do databáze.

Update Aktualizuje všechny informace o daném působišti v databázi.

5.2.6. Chyba

Třída *Chyba* se využívá prakticky ve všech ostatních třídách a je určena ke zpracování vzniklých chyb, sestavení chybové hlášky a její vypsání na výstup případně s využitím třídy *Log* do záznamového souboru. Následuje výčet nejdůležitějších metod:

Set Nastaví identifikátor vzniklé chyby, uloží čas jejího vzniku a upraví znění chybové hlášky.

Get Vrací identifikátor posledně vzniklé chyby. Dá se použít při ověřování, zda vznikla nějaká chyba v předešlých krocích běhu skriptu.

Kapitola 6

Výsledky a zhodnocení

Úkolem testování bylo ověřit funkčnost vyhledávacího systému a zhodnotit kvalitu získaných dat. Nejprve jsem pomocí jednotlivých skriptů získal potřebná data a ověřil tak úspěšnost při jejich činnosti. Poté jsem manuálně posuzoval kvalitu získaných dat. Tento posudek jsem ovšem provedl pouze nad omezeným vzorkem dat, neboť jak se ukázalo, jednotlivé skripty získaly až tisíce záznamů a jejich manuální procházení je abnormálně časově náročný proces.

6.1. Test činnosti jednotlivých sub-systémů

6.1.1. FindNew

Pro vyhledání nových článků jsem zadal jako vstup skriptu FindNew následující klíčová slova ve tvaru:

`parsing OR parse.`

Skript FindNew uložil do databáze informace o 1072 článcích. Po manuální kontrole, kolik jich ve skutečnosti ResearchIndex nabídl, jsem zjistil, že je to pouze osmina ze všech nalezených. Důvod, proč se nepodařilo získat informace i o dalších článcích, byl ovšem na straně ResearchIndexu, neboť ten je neustále vytížen a nabízí tak z vyhledaných záznamů pouze omezené množství. Kolikrát se mi například během ladění skriptu stávalo, že nabídl ResearchIndex jen stovku z celkem šesti tisíc nalezených záznamů a na další už se nebylo možné dostat.

K těmto 1072 záznamům s informacemi o jednotlivých článcích bylo uloženo do databáze dalších 1780 záznamů s informacemi o jejich autorech. Dále bylo zapsáno 3021 záznamů o citacích a 4286 záznamů o podobnosti jednotlivých článků.

Z těchto údajů lze vyvodit různé statistické závěry jako například to, že na jeden článek připadají přibližně dva autoři nebo že jeden článek je citován průměrně ve třech dalších článcích a je podobný se čtyřmi dalšími články.

6.1.2. ArticleDownload

Před spuštěním skriptu ArticleDownload už není potřeba zadávat žádný vstup. Tento skript pracuje samostatně nad záznamy uloženými v databázi, proto již popíše pouze dosažené výsledky.

Po spuštění se tomuto sub-systému podařilo k 1072 uloženým záznamům s informacemi o článcích získat 837 ps nebo pdf souborů, kde by měl být samotný text jednotlivých článků. PostScriptových souborů bylo 523 a ve formátu pdf jich bylo zbylých 314.

Abych zjistil, do jaké míry jsou stažené soubory použitelné pro fulltextové vyhledávání, převedl jsem je pomocí programů `ps2ascii` a `pdftotext` do souborů obsahujících holý text. Po podrobnějším prozkoumání jsem zjistil, že z celkem 523 PostScriptových souborů bylo 261 převedeno v pořádku. Pro dalších 261 souborů hlásil program `ps2ascii` buď jednu z chyb *undefined in ch-xoff* a *stackunderflow in dup* nebo byly v textovém souboru jen tečky. Takové soubory obsahovaly obrázky nebo byly příliš velké, ale zobrazit se daly například pomocí programu Gnome GhostView 1.1.93 bez problému. Zbylý jeden případ se nepodařilo převést, protože nebyl soubor ve formátu PostScript. Co se týče pdf souborů, bylo z celkem 314 převedeno v pořádku 303 souborů do holého textu a zbylých 11 se nepovedlo převést, protože nebyly ve formátu pdf.

6.1.3. AuthorSearch

Ke spuštění skriptu AuthorSearch také není potřeba zadávat žádný vstup, neboť opět pracuje samostatně nad záznamy uloženými v databázi a popíše tedy pouze dosažené výsledky.

Po spuštění tento sub-systému našel u 472 autorů z celkem 1780 adresu domovské stránky. Z množiny autorů, u kterých se podařilo zjistit URL jejich domovské stránky, zjistil skript pouze pro 118 autorů adresu webové stránky obsahující seznam publikací. Dále se díky této množině autorů podařilo zjistit informace o 295 působištích.

URL stránky se seznamem publikací se nepodařilo najít v případech, kdy nebyl odkaz na seznam publikací na domovské stránce autora nebo tam byl, a pak nastaly dvě možnosti, při kterých byl skript neúspěšný. Buď název odkazu nespadal do množiny výrazů pod, kterými se hledá, anebo byl název odkazu identifikován ale adresa samotného odkazu už ne. Pro ilustraci druhého případu uvádím příklad zdrojového kódu stránky¹, kde daná situace nastala:

```
<td ALIGN=CENTER>
<br>
<a href="publ.html"> <font SIZE=4>
<HREF=http://www.imc.pi.cnr.it/~codenotti/publ.html/> Publications
</a></font><br>
</td></tr>
```

Jak je vidět z tohoto příkladu, naráží občas skript při své činnosti i na webové stránky, kde nejsou jednotlivé tagy správně vnořeny nebo neodpovídají specifikaci formátu HTML. Aby byl skript dokonalejší, je potřeba implementovat i rozpoznávání takovýchto případů.

Pro zhodnocení kvality uložených informací o působištích jsem vybral vzorek s 40 náhodně vybranými zástupci z celkem 295 záznamů. Z tohoto vzorku mělo 8 působišť

¹<http://www.imc.pi.cnr.it/~codenotti>

špatný název. Názvy obsahovaly například jméno vyhledávacího serveru, jméno serveru provozujícího domovské stránky, adresa serveru nebo uvítací zpráva programu Apache.

6.2. Shrnutí výsledků testování a návrh na další řešení

Nyní u každého sub-systému postupně shrnu jeho dosažené výsledky, popíši nedostatky a pokusím se navrhnout možná řešení těchto nedostatků.

Skript FindNew získal jen osminu ze všech článků, ve kterých ResearchIndex našel odpovídající klíčová slova. Pokud se ovšem k těm dalším nebylo možné dostat ani manuální cestou, dá se říci, že sub-systém FindNew dosáhl výborných výsledků.

Pro sub-systém FindNew by bylo dobré vymyslet způsob, jak se dostat k dalším informacím o článcích, které už ResearchIndex nechce z důvodu velkého vytížení systému poskytnout. Nejsem si ovšem jist, je-li to vůbec možné.

Skript ArticleDownload uložil soubory s textem zhruba ke čtyřem pětinám ze všech článků. Z těchto souborů se pak podařilo správně převést do holého textu pouze polovina. Velkou slabinou sub-systému ArticleDownload je jeho neschopnost rozpoznat i jiné formáty než PostScript a pdf. Jednu pětinu článků se nepodařilo získat právě z tohoto důvodu. Zpravidla byly soubory obsahující text těchto článků nějakým způsobem komprimovány.

Do sub-systému ArticleDownload by bylo vhodné integrovat metody, které by umožnily získat text i z jiných formátů než PostScript a pdf. Také by se mohly vyzkoušet i jiné programy pro převod PostScriptových souborů do textu, neboť ps2ascii v tomto případě nedosáhl moc dobrých výsledků.

A nakonec skript AuthorSearch zjistil URL domovské stránky přibližně u jedné čtvrtiny z celkového počtu autorů. V případě, že byla u autora zjištěna adresa domovské stránky, našel skript na této stránce odkaz na seznam publikací až na výjimky vždy. Co se týče působišť, byly zjištěné informace o názvu správně zhruba ve čtyřech pětinách. Největším nedostatkem tohoto sub-systému je vyhledávání adres domovských stránek autorů.

Sub-systém AuthorSearch se stará jak o dohledávání informací k autorům tak i k působištím. Pro zlepšení přehlednosti a kontrolovatelnosti by mohl zjišťovat informace o působištích další nezávislý sub-systém. Za účelem zlepšit výsledky, kterých skript AuthorSearch dosahuje při hledání URL domovské stránky autora, by bylo do budoucna potřeba navrhnout metodu jak rozpoznat ve výsledcích, jež HP Search nabízí, který záznam patří autorovi, o kterého se zajímáme.

Kapitola 7

Závěr

V rámci projektu se mi podařilo navrhnout a následně implementovat vyhledávací systém, který lze následně využít k plnění informačních zdrojů vhodných pro činnost webového portálu věnovaného syntaktické analýze přirozeného jazyka. Pomocí tohoto vyhledávače jsem získal informace přibližně o tisíci článcích týkajících se dané problematiky a k tomu také informace o autorech těchto článků a o působištích jednotlivých autorů. Zbývá tedy ještě vytvořit patřičné webové rozhraní, pomocí kterého se bude moci se získanými daty pracovat.

Při vývoji vyhledávacího systému jsem narazil na několik velkých problémů, jejichž výskyt byl podmíněn použitím skriptovacího jazyka PHP. Ve zpětném pohledu přemýšlím nad tím, zda by se spousta z nich objevila v případě, že bych použil jiného skriptovacího jazyka jako například Perl nebo přímo některého programovacího jazyka jako C++ nebo Java.

Doufám, že bude tato práce poučením pro další zájemce, kteří by rádi implementovali vyhledávací systémy a chtěli k tomu použít pouze skriptovacího jazyka PHP.

Příloha A

Instalace

A.1. Požadavky pro používání systému

Celý systém byl vyvinut pod operačním systémem Debian Linux, ale fungovat by měl i na jiných platformách Unixu a také pod operačním systémem Windows, neboť je založený na platformě nezávislých technologiích. Na následujících řádcích bude vyjmenováno, co všechno je potřeba mít na počítači nainstalováno, aby byl systém funkční:

- je nutné mít interpret PHP skriptů¹ (verze 4.06 a vyšší).
- dalším důležitým prvkem je databáze. Zde byla použita volně šířená databáze MySQL².
- pro spouštění jednotlivých skriptů přes Webové rozhraní je požadavkem mít na počítači zprovozněný WWW server. Využít se dá například volně šířený Apache³.

A.2. Adresářová struktura

Instalace systému je jednoduchá. Po uživateli je vyžadována pouze základní znalost operačního systému, na kterém chce danou aplikaci provozovat.

Systém je distribuován jako jeden soubor zkomprimovaný programem zip. Po jeho rozbalení se vytvoří jednoduchá adresářová struktura, kde jednotlivé složky mají následující význam:

inc Obsahuje soubor konfig.php, kde jsou definovány všechny konstanty používané v ostatních skriptech. Bližší popis najdete v části Konfigurace systému.

lib Zde jsou knihovny všech výše popsaných tříd.

log Je určen k ukládání záznamů o průběhu činnosti jednotlivých skriptů.

¹<http://www.php.net>

²<http://www.mysql.com>

³<http://www.apache.org>

ps Sem se ukládají získané soubory ve formátu ps nebo případně pdf, pokud se PostScript nepodaří najít či stáhnout.

txt Tento adresář je určen pro holé texty článků extrahované z ps nebo pdf souborů z předešlého adresáře. Bude se využívat pro budoucí fulltextové vyhledávání.

Bude-li systém provozován na počítači s operačním systémem, kde lze přidělovat přístupová práva, je nutné u adresářů log, ps a txt nastavit práva pro zápis.

A.3. Vytvoření databáze

Prvním krokem je vytvoření uživatele v MySQL a nastavení jeho hesla. Přihlásíte se jako administrátor do mysql a zadáte příkaz:

```
INSERT INTO user (Host,User,Password)
-> VALUES('host','uživatel',PASSWORD('heslo'));
```

kde host je buď adresa serveru nebo přímo řetězec *localhost* podle toho, zda bude uživatel k mysql přistupovat z jiného stroje nebo ne.

Dalším krokem je vytvoření databáze s patřičnými tabulkami. Přihlásíte se do mysql pod nově vytvořeným uživatelem a zadáte příkaz:

```
\. createdb.sql;
```

A.4. Konfigurace systému

Důležitým krokem je nastavení parametrů pro správný chod vyhledávacího systému. V souboru konfig.php, který se nachází v adresáři inc, lze nastavit následující:

MYSQL_HOST adresa MySQL serveru, kde je vytvořena databáze potřebná k činnosti vyhledávacího systému.

MYSQL_UZIVATEL jméno uživatele, pod kterým budou skripty přistupovat k dané databázi na MySQL serveru.

MYSQL_HESLO heslo, které je nutné pro přístup do databáze.

MYSQL_DATABASE jméno databáze, jenž je určena k ukládání nalezených informací.

QUERY_STRING řetězec, který se použije jako dotaz pro ResearchIndex při vyhledávání nových článků. Lze použít binární operátory jako OR, AND.

AMOUNT počet stránek, jež budou chtít skripty od ResearchIndexu poskytnout. Pokud je například ResearchIndex přetížen a dochází tak k přerušení činnosti jednotlivých skriptů, je dobré nastavit zde nižší hodnotu.

MAX_POKUSU udává počet opakování pokusu o přístup k požadovanému dokumentu v případě, kdy server neodpovídá.

Literatura

- [Axmark] *MySQL manual*, David Axmark, Michael Widenius, Jeremy Cole, Arjen Lentz a Paul DuBois, NuSphere, 2002, <http://www.mysql.com/documentation>.
- [Bakken] *PHP manual*, Stig Sather Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Andrei Zmievski a Jouni Ahto, PHP Documentation Group, 2002, <http://www.php.net/manual>.
- [Blaták] *Extrakce textu z PostScriptu a PDF*, Jan Blaták, Bakalářská práce FI MU, Brno 2000.
- [Castagnetto] *Programujeme PHP profesionálně*, Jesus Castagnetto, Harish Rawat, Sascha Schumann, Chris Scollo a Deepak Veliath, Computer Press, Praha 2001, 80-722-310-2.
- [Hoff] *Finding Scientific Papers with HPSearch and Mops*, Gerd Hoff a Martin Mundhenk, 1999, Universität Trier.
- [Lawrence] *Digital Libraries and Autonomous Citation Indexing*, Steve Lawrence, C. Lee Giles a Kurt Bollacker, 1999, NEC Research Institute.
- [Ley] *Computer Science Bibliography, DBLP FAQ*, Michael Ley, 1999, Universität Trier, <http://www.informatik.uni-trier.de/~ley/db/about/faq.html>.
- [Mariánek] *Intelligentní vyhledávání na www*, Josef Mariánek, Diplomová práce FI MU, Brno 2001.
- [Mundhenk] *Martin's Online Paper Search - a Distributed virtual digital library*, Martin Mundhenk, <http://mops.uni-trier.de/~mops/about.html>.
- [Sochor] *Analýza a návrh systémů*, Jiří Sochor, Skripta FI MU, Brno 2001.
- [Šimůnek] *SQL kompletní kapesní průvodce*, Milan Šimůnek, GRADA Publishing, Praha 1999, 80-7169-692-7.

Rejstřík

ACI, 9, 10
Affiliation, 24
affiliation, 19, 20
article, 18, 19
article_author, 19
ArticleDownload, 16, 22, 25, 27
author, 19, 20
author_affiliation, 20
AuthorSearch, 17, 22, 26, 27
AuthorSearch, 27
Autor, 24

BibTex, 6

Chyba, 24
citation, 18
Clanek, 23
crawlers, 7

datový model, 17
DBLP, 12

FindNew, 16, 21, 25, 27

Google, 4, 6, 9

horizontální portály, 5
HP Search, 11, 12, 17, 27

Log, 21, 23, 24

Mops, 11, 13
MySQL, 22, 30

pdftotext, 26
PHP, 21, 22, 28, 29
ps2ascii, 26, 27

ResearchIndex, 4, 9, 10, 16, 17, 21–23, 25,
27, 30

similar, 19
spiders, 7

vertikální portály, 5
vortals, 5
vyhledávací systém, 5, 9, 12, 13, 15, 16, 28

webový portál, 4, 5, 7, 8, 15
white pages, 6

yellow pages, 6