VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PRŮVODCE KONFERENCÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE PETR VIKTORIN
AUTHOR

BRNO 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PRŮVODCE KONFERENCÍ

CONFERENCE PLANNER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — PETR VIKTORIN

AUTHOR

VEDOUCÍ PRÁCE — Doc. RNDr. PAVEL SMRŽ, Ph.D.

SUPERVISOR

BRNO 2009

Abstrakt

Cílem této práce je seznámení se s metodami získávání sémantických informací z textu. Je popsána *ad hoc* metoda pro zpracování programů konferencí a dvě metody odhadu podobnosti krátkých textů založené na formálním popisu. Tyto metody jsou aplikovány na problém doporučení individuálního programu vědecké konference.

Abstract

This work explores the methods of extracting semantic information from natural language texts. An *ad-hoc* method for analyzing conference programmes and two formal methods for estimating the relatedness of two short texts are explained. These methods are applied to the problem of recommending an individual schedule for a scientific conference.

Klíčová slova

Sémantické informace, program konference

Keywords

Semantic information, conference programme

Citace

Petr Viktorin: Průvodce konferencí, bakalářská práce, Brno, FIT VUT v Brně, 2009

Průvodce konferencí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Doc. RNDr. Pavla Smrže, Ph.D.
Další informace mi poskytl Ing. Marek Schmidt.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

. . . . . .
Petr Viktorin
20. května 2009

© Petr Viktorin, 2009.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod **3**

2 Teoretický základ **4**
2.1 Podobnost podle D. Lina 4
2.2 Model vektorového prostoru 5
2.3 Explicitní sémantická analýza 5
2.4 Výpočet nejdelší cesty v orientovaném acyklickém grafu 6

3 Zpracování programu konference **7**
3.1 Úvod k implementaci 7
3.2 Schéma databáze 7
Přednáška 7
Konference 8
Člověk 8
Další entity 8
3.3 Vstupní data 8
3.4 Návrh extrakčního skriptu 10
Extrakce informací z HTML 10
hCalendar 11
3.5 Použití jiných extrakčních skriptů 12

4 Navržení individuálního programu **13**
4.1 Matematický model 13
4.2 Algoritmus 14
4.3 Ohodnocení jednotlivých přednášek 15
Trénovací data 16
Podobnost podle D. Lina 16
Explicitní sémantická analýza 17
4.4 Výstup programu 18

5 Vyhodnocení výsledků **19**
5.1 Vyhodnocení extrakce programu konference 19
Testovací data 19
Výsledky testování 19
5.2 Úspěšnost hodnocení přednášek 21
Testovací data 21
Metoda vyhodnocení 21
Výsledky vyhodnocení 22

6 Závěr **24**

A Obsah CD **26**

B Další tabulky **27**

Kapitola 1

Úvod

S cílem usnadnit zadávání, uchovávání a vyhledávání informací o určitém vědeckém oboru vznikl na Fakultě informačních technologií VUT projekt Generátor vědeckých portálů (GVP). Projekt má sloužit studentům, vědcům a výzkumníkům k rychlé orientaci v daném oboru prostřednictvím dlouhodobě budované báze dat.[8] Nedílnou součástí tohoto projektu je práce s informacemi o vědeckých konferencích.

Tato práce popisuje systém, který analyzuje zadaný program konference a navrhne uživateli vhodný individuální program. Jsou v ní přiblíženy metody získávání sémantických vazeb z textu a aplikace těchto metod na automatické doporučení přednášek uživatelům.

V první kapitole jsou uvedeny teoretické základy, z nichž zbytek textu vychází.

Ve druhé kapitole je popsán způsob extrakce informací z programu konference, včetně analýzy programů použitých jako vstupní data a schématu výsledné databáze.

Ve třetí kapitole je uveden mechanismus navržení optimálního individuálního programu. Také jsou zde diskutovány použité způsoby ohodnocení jednotlivých přednášek podle zadaných klíčových slov.

Ve čtvrté kapitole je vyhodnocena úspěšnost použitých metod.

Kapitola 2

Teoretický základ

2.1 Podobnost podle D. Lina

Jednou z možností, jak počítat „blízkost“ termínů, je použít podobnost založenou na teorii informací, jak ji definuje D. Lin v [5]. Jedná se o obecnou definici vycházející z několika intuitivních předpokladů, která říká, že podobnost objektů A a B je podíl množství informace potřebné pro popis toho, co mají A a B společné, a informace potřebné pro popis A a B. Množství informace v nějakém sdělení je definováno jako záporný logaritmus pravděpodobnosti tohoto sdělení. Symbolicky je tedy podobnost definována jako:

$$\mathrm{sim}(A, B) = \frac{\log P(\mathrm{common}(A, B))}{\log P(\mathrm{description}(A, B))}$$

Lin dále specializuje tuto definici konkrétně pro podobnost slov, a to na základě kontextů, ve kterých se dané slovo vyskytuje. Používá parser, který každou sémantickou vazbu v textu reprezentuje jako trojici hlavního členu, třídy vazby a vedlejšího členu. Např. větu “I have a brown dog” parser převede na tyto trojice:

(have subj I), (have obj dog), (dog adj-mod brown), (dog det a)

kde „subj“ reprezentuje vztah mezi podnětem a přísudkem, „obj“ vztah mezi slovesem a předmětem, „adj-mod“ je vztah přívlastkový a „det“ vztah mezi podstatným jménem a jeho členem.

Slova jsou charakterizována vazbami, ve kterých se vyskytují. Například slovo „dog“ má vlastnost „adj-mod brown“, a naopak „brown“ má vlastnost „adj-mod-of dog“. Z pohledu teorie informací je pak slovo w možné popsat množinou všech takových vlastností $F(w)$. To, co mají dvě slova společné, popíšeme průnikem jejich množin vlastností.

Podobnost slov je pak definována jako

$$\mathrm{sim}(w_1, w_2) = \frac{2 \times I(F(w_1) \cap F(w_2))}{I(F(w_1)) + I(F(w_2))} \tag{2.1}$$

kde $I(S)$ je množství informace obsažené v množině vlastností S. Za předpokladu, že jsou tyto množiny nezávislé, platí $I(S) = -\sum_{f \in S} \log P(f)$, kde $P(f)$ je pravděpodobnost vlastnosti f. Ta se dá odhadnout jako podíl slov, které mají danou vlastnost, a všech slov v textu, které mají stejný slovní druh.

2.2 Model vektorového prostoru

Tradiční metoda pro porovnávání významů textů je použití tzv. modelu vektorového prostoru (*vector-space model*), kdy je každý dokument (nebo věta, dotaz) reprezentován vektorem termínů (*term vector*) [7]. Tradičně se používá vektor všech slov, které se vyskytují v předem daném korpusu. Hodnota přiřazená každému slovu je tzv. váha, a na způsobu jejího výpočtu závisí výsledky metody.

Výpočet váhy většinou kombinuje počet výskytů slova v daném dokumentu (*term frequency,* označuje se $\mathrm{tf}_{i,j}$ pro slovo i a dokument j), a počet všech dokumentů, v nichž se dané slovo vyskytuje (*document frequency,* df_i). Čím častěji se slovo v daném dokumentu vyskytuje, tím větší je jeho váha. Naopak pokud se slovo vyskytuje v mnoha různých dokumentech, tím je pravděpodobnější, že nemá vztah ke konkrétním informacím v dokumentu. Jeho váha je proto nižší.

Metody výpočtu váhy založené na těchto dvou veličinách se označují *tf.idf* a liší se od sebe konkrétním způsobem použití frekvencí a normalizace hodnot. Jeden ze způsobů, uvedený v [7], je následující:

$$\mathrm{weight}(i,j) = \begin{cases} (1+\log(\mathrm{tf}_{i,j}))\log\frac{N}{\mathrm{df}_i} & \text{pro } \mathrm{tf}_{i,j} \geq 1 \\ 0 & \text{pro } \mathrm{tf}_{i,j} = 0 \end{cases}$$

kde N je celkový počet dokumentů. Tento způsob se označuje „ltn“, což je zkratka pro logaritmické vážení počtu výskytů slova v dokumentu, logaritmické vážení frekvence slova ve všech dokumentech a žádnou normalizaci.

Podobnost dvou textů, nebo textu a dotazu, potom odpovídá kosinu úhlu mezi vektory vah slov v nich obsažených:

$$\mathrm{podobnost}(A,B) = cos(\theta) = \frac{A \cdot B}{|A|\,|B|}$$

Problém při použití vector-space modelu spočívá v tom, že počítá s vektory vah prakticky všech slov, které se vyskytují v daném jazyce. Tyto vektory jsou pak nutně velké, což ztěžuje práci s nimi. Navíc tato metoda nezohledňuje fakt, že jeden význam může být vyjádřen různými synonymy. Tento problém je výraznější u porovnávání krátkých frází, kde je větší pravděpodobnost, že se konkrétní slovní zásoba nebude shodovat.

Byly proto vyvinuty metody, které v term vektorech slučují slova podobných významů a tím tento problém řeší. Klasická metoda tohoto typu je latentní sémantická analýza (LSI), která využívá matematické metody SVD (Singular Value Decomposition). Vektory jsou při jejím použití promítnuty do prostoru nižší dimenze, který zachovává osy s nejvyšší variací (axes of greatest variation).

Podrobnější informace lze nalézt v [7].

2.3 Explicitní sémantická analýza

Variantou LSI je tzv. explicitní sémantická analýza (ESA), kterou popisují její autoři E. Gabrilovich a S. Markovitch v [3]. Metoda ESA využívá encyklopedických článků, které jsou napsány o konkrétních tématech, a sdružují tak související termíny. Autoři této metody zvolili za zdroj článků Wikipedii (`http://en.wikipedia.org`), největší existující encyklopedii.

Následuje stručný popis této metody podle [3].

Pro použití metody ESA nejdříve vytvoříme *inverzní index*, který přiřazuje jednotlivým slovům seznamy článků, v nichž se vyskytují. Následně každý vstupní text zpracujeme *sémantickým interpretrem*, který mu přiřadí vektor souvisejících článků podle následujícího postupu:

Mějme vstupní text $T = W_i$. $\langle v_i \rangle$ je jeho TFIDF vektor, kde v_i je váha slova w_i. k_i je záznam reverzního indexu pro slovo w_i, kde k_j určuje významnost slova W_i z pohledu článku $c_j, c_j \in c_1, \ldots, c_N$ (N je počet všech článků). Potom je sémantickou interpretací textu T je vektor V o délce N, ve kterém váha každého konceptu c_j je definována jako $\sum_{w_i \in T} v_i \cdot k_j$. Prvky tohoto vektoru vyjadřují vztah textu T k příslušnému článku. Porovnáním vektorů pomocí kosínové metriky je pak možné odhadnout souvislost mezi dvěma texty.

Podle zmíněného článku není metoda ESA vhodná jen pro porovnávání delších textů, ale i pro srovnání významů jednotlivých slov či frází.

2.4 Výpočet nejdelší cesty v orientovaném acyklickém grafu

Máme-li orientovaný acyklický graf s ohodnocenými hranami, lze podle [2] nalézt nejdelší cestu mezi dvěma jeho vrcholy pomocí následujícího algoritmu:

Pro každý uzel x si pamatujeme hodnotu $U(x)$, což délka nejdelší zatím nalezené cesty, a odkud(x), což je uzel, který x v nejdelší cestě předchází.

1. Pro každý uzel x nastavíme $U(x) := -\infty$
2. Pro počáteční uzel s nastavíme $U(s) := 0$
3. Pro všechny hrany $x \to y$, v topologickém uspořádání:
   - Pokud $U(y) < U(x) + a(x, y)$, kde $a(x, y)$ je hodnota hrany $x \to y$:
     - položíme $U(y) := U(x) + a(x, y)$
     - položíme odkud$(y) := x$

Algoritmus 2.1: Výpočet nejdelší cesty v orientovaném acyklickém grafu

Kapitola 3

Zpracování programu konference

3.1 Úvod k implementaci

Jako plánovaná součást projektu Generátoru vědeckých webových portálů je průvodce konferencí napsán v programovacím jazyku Python, stejně jako většina ostatních komponentů GVP.

Python je vysokoúrovňový skriptovací jazyk, jehož hlavní výhody jsou rychlost tvorby aplikací a možnost propojení s knihovnami napsanými v jiných jazycích, např. v jazyce C.

Aby byl systém co nejpružnější, je rozčleněn do několika modulů, a to:

1. modul pro převod programů přednášek z HTML a jiných formátů do databáze,
2. modul pro ohodnocení jednotlivých přednášek,
3. modul pro sestavení výsledného programu, a nakonec
4. modul pro testování úspěšnosti hodnocení přednášek.

Tato kapitola se zabývá prvním z nich.

3.2 Schéma databáze

Jeden ze základních problémů projektu je navržení schématu databáze, kterou tento modul plní. V databázi musí být uloženy informace o jednotlivých přednáškách a konferencích.

V rámci průvodce konferencí je navržena jen část databáze, která bude společná pro celý GVP. Bylo tedy důležité zohlednit nejen potřeby návrhu individuálního plánu konference, ale i propojení se zbytkem vznikajícího systému.

Konceptuální schéma databáze je navrženo jako ER (*Entity-Relationship*) model.

Přednáška

Klíčovou entitou je *přednáška*, což je jediná entita potřebná pro návrh individuálního plánu. Tato entita reprezentuje bod programu konference – přednášku, seminář nebo plakát. Pro samotné plánování jsou důležité atributy *začátek* a *konec*, které obsahují čas začátku, resp. konce přednášky. Kromě času je nutné uchovávat i datum, kdy se přednáška koná. Bylo by možné pro datum vyhradit další atribut, ale toto řešení by neumožňovalo reprezentovat přednášky, které začínají a končí v různých dnech. Taková přednášky se může objevit například pokud jsou časy ukládány v UTC. Proto je informace o datu ukládána společně

s časem. Toto řešení má navíc tu výhodu, že k řazení přednášek podle času začátku či konce potřebujeme pouze jeden atribut.

Z hlediska hodnocení přednášek je důležitý *název* přednášky. Podle něj můžeme zjistit, o čem přednáška je, a tedy jestli je doporučit uživateli.

Poslední modelovaný atribut přednášky jsou *klíčová slova*. Ta většinou v programech konferencí nebývají, ale pokud jsou k dispozici, mohou se podobně jako název použít ke zjištění tématu přednášky.

Konference

Další modelovaná entita je *konference*. Ta má atributy *název*, *číslo* a URL. Číslo je nepovinné, ale většina konferencí – a zvláště těch s paralelními sekcemi – se každoročně opakuje, a v názvu má uvedeno buď číslo nebo letopočet. Celý název konkrétní konference tedy dostaneme textovým spojením atributů název a číslo. URL je domovská stránka konference, případně adresa, ze které byl stažen program.

Člověk

Entita *člověk* je v průvodci konferencí použita pro přednášející. Člověk má atributy *jméno* a *příjmení*. Atribut jméno obsahuje vedle křestního také případná další jména, častá v anglicky mluvících zemích. Celé jméno tedy získáme textovým spojením těchto dvou atributů.

Další entity

Dále modeluji entitu *místo* jako budovu, ve které přednášky probíhají. Její atributy jsou *město* a *stát*, ve kterém se nachází, a její *jméno*.

Poslední modelovaná entita je *místnost*, konkrétní určení, kde se koná přednáška. V případě přednášek s paralelními sekcemi se všechny přednášky jedné sekce odehrávají v jedné místnosti.

Tyto dvě entity nejsou v průvodci konferencí použity, protože není způsob, jak je spolehlivě získat z programu konference.

Vztahy mezi modelovanými entitami jsou zobrazeny na obr. 3.2.

3.3 Vstupní data

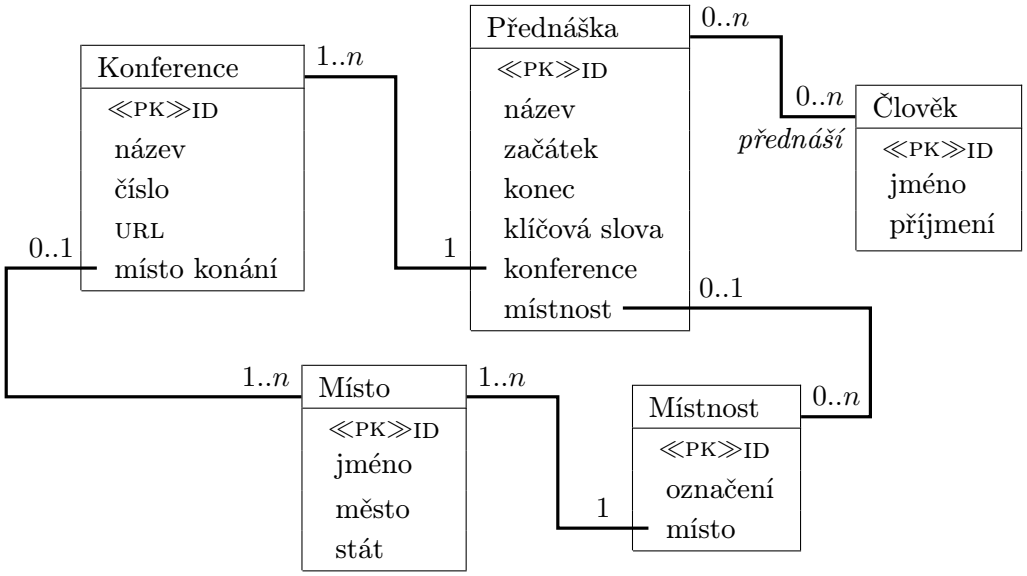
Jako vývojová data pro modul extrakce informací z programu byly použity přednášky týkající se zpracování přirozeného jazyka z roku 2008. Jejich seznam je k nalezení na adrese `http://aclweb.org/aclwiki/index.php?title=Conferences_in_2008`.

Z šestnácti stránek konferencí uvedených na této stránce jsou dvě (IJCNLP a GoTAL) již nedostupné. Zbylé nabízejí programy ve formátech HTML a PDF. V jiných formátech se programy přednášek prakticky neobjevují.

Existují nástroje na převod dokumentů ve formátu PDF na HTML. Jedním z nich je `pdf2xhtml`, který vytvořil Radek Bartoň. Jeho nevýhodou je, že neumí vždy správně převádět tabulky a text vysázený do více sloupců. Jednoduché dokumenty však převádí uspokojivě. Skript pro extrakci informací se proto zabývá jen formátem HTML.

Následuje stručný popis jednotlivých typů zdrojových programů.

Pro konferenci AAAI je program zveřejněn ve formátu PDF se třemi sloupci textu. Kromě samotného časového plánu program obsahuje anotace přednášek, plánky areálu, obrázky a jiné materiály. Po převodu do HTML dostupnými nástroji většina obsahu nebyla zachována.



Obrázek 3.1: Návrh schématu databáze

Program konference ACL je v čistém HTML. Záznam každé přednášky obsahuje čas začátku a konce, název a seznam autorů. Tyto údaje jsou vyznačeny HTML značkami. Jediná nepříjemnost v tomto programu je blok plakátů a demonstrací, kde je čas začátku a konce uveden před celým blokem místo u jednotlivých přednášek.

Stejné schéma je použito v programu EMNLP, jen s jiným výběrem značek pro jednotlivé položky. Konference CONNL, CICLing, FSNLP, HPSG a IWSLT používají také velice podobný formát, jen s tabulkami pro grafickou úpravu.

Konference AMTA a COLING mají více paralelních sekcí. Jejich program je ve formátu HTML a používá tabulky se sloupcem pro každou sekci a řádkem pro jednotlivé časové bloky.

Konference ICGL předkládá program ve formátu PDF. Jednotlivé konference jsou v něm graficky vyznačeny jako řádky tabulky. Jde o konferenci bez paralelních sekcí, takže obsahuje pouze jednu přednášku na řádek. Po převodu do HTML se sice ztratila struktura tabulky, ale požadované údaje o přednáškách jsou stále zřejmé.

Program konference LREC je zajímavý tím, že se skládá ze tří dokumentů PDF, jednoho pro každý den konference. Před převodem do HTML byly tyto dokumenty spojeny do jednoho.

Program poslední konference v seznamu ACL, je ve formátu PDF. Jednotlivé přednášky jsou aranžovány do tabulky o dvou sloupcích, které odpovídají dvěma paralelním sekcím. Převod do HTML bohužel strukturu dokumetu poruší.

Programy se tedy dají rozdělit na tři kategorie podle toho, jak používají tabulky:

- Bez tabulek.
- Se strukturou přednášky reprezentovanou tabulkou.
- S tabulkami použitými jen pro vizuální formátování.

Dále je možné konference rozdělit na ty s paralelními sekcemi, na kterých probíhá několik přednášek najednou, a na ty bez paralelních sekcí. Pro samotný průvodce konferencí je důležitý jen první typ – optimální individuální program konference bez paralelních sekcí

zahrnuje všechny přednášky – ale dává smysl, aby extrakce informací z programů fungovala i pro jednodušší konference. Získané informace tak budou použitelné i pro jiné nástroje.

V tabulce 3.1 jsou počty programů, které se v testovacích datech objevují, pro přehlednost shrnuty.

	Počet programů		
Formát	S paralelními sekcemi	Bez paralelních sekcí	Celkem
HTML, tabulek	3		3
HTML, strukturovací tabulky	2		2
HTML, prezentační tabulky	1	4	5
PDF bez tabulek		1	4
PDF, strukturovací tabulky	2		2
PDF, prezentační tabulky		1	5
Celkem	8	6	14

Tabulka 3.1: Přehled formátů programů konferencí použitých při vývoji programu

Je zřejmé, že extrakční skript by neměl brát ohled na konkrétní strukturu a formát dokumentu, protože ta je u každého programu jiná.

3.4 Návrh extrakčního skriptu

Systém se při určování přednášek může dopustit tří typů chyb:

1. Najde přednášku tam, kde se nejedná o přednášku,
2. Nenajde přednášku, ačkoli v programu je uvedena, nebo
3. Špatně přiřadí informace (čas, název nebo autory) ke správně nalezené přednášce.

Pokud budeme považovat určení, zda je daná část textu popis přednášky či nikoli, za klasifikační problém, zjistíme, že první dva zdroje chyb odpovídají chybám typu I a II, popsaným např. v [9]. Platí, že zvýšením pravděpodobnosti výskytu jednoho z těchto druhů chyb snížíme pravděpodobnost výskytu druhého druhu a naopak.

Extrahované informace budou součástí vědeckého portálu, který bude uživatelům umožňovat změnu jakýchkoli informací v něm obsažených. V takovém případě je mnohem jednodušší špatně určenou přednášku vymazat než znovu opisovat informace z programu. Špatně určené přednášky také pravděpodobně nebudou dobře hodnoceny, takže se neobjeví v doporučeném individuálním programu. Proto je důležité se hlavně zaměřit na odstranění chyb typu II, tj. snažit se nalézt všechny přednášky, a určitý počet chybně určených přednášek v programu tolerovat.

Třetí druh chyb je například záměna názvu přednášky za seznam autorů. I tyto chyby bude snadné opravit v interaktivním editoru, při generovaném individuálním programu však nebudou adekvátně hodnoceny.

Extrakce informací z HTML

Extrakční skript využívá hierarchickou strukturu dokumentu, ale nespoléhá na to, že je přesná.

Pro práci s HTML dokumenty je použita knihovna BeautifulSoup, která reprezentuje dokument jako strom jednotlivých prvků. BeautifulSoup umožňuje zpracování i nekorektního HTML. Chyby většinou opravuje tak, jako to dělají grafické prohlížeče.

Skript prochází jednotlivé uzly dokumentového stromu v pořadí, v jakém jsou uvedeny v programu. V první řadě hledá data (ve smyslu určení dnů, kdy se konference koná). Pokud narazí na rok, měsíc nebo datum, příslušný údaj si zapamatuje. Měsíce a dny mohou být určeny dvěma způsoby – buď číslem, nebo slovně. Pokud skript narazí na údaj určený slovně, předpokládá, že tak budou určeny i ostatní údaje, a dále ignoruje čísla, která by mohla vyjadřovat stejnou informaci.

Druhý údaj, který skript hledá, jsou časové údaje. Předpokládá – a vývojová data potvrzují – že časový údaj se objevuje vždy před přednáškou, ke které se vztahuje, nebo případně v rámci stejného uzlu. Při průchodu dokumentem je proto zaznamenán poslední objevený čas. Tyty údaje mohou být ve tvaru rozmezí (např. 10:00–10:30), nebo jen jednoho času (např. 10.00), který značí začátek přednášky. Takové přednášky jsou „uzavřeny“ nastavením času konce na další časový údaj, který se v dokumentu objeví.

Souběžně program kontroluje obsah textových uzlů, na které narazí. Pokud uzel obsahuje alespoň dva textové poduzly, jsou tyto zpracovány jako názvy a autoři přednášek a v nadřazených uzlech se už nezpracovávají. Pokud ne, je případný obsah dále zpracováván nadřazeným uzlem.

Informace o názvu a autorech přednášky se zpracovávají heuristickým algoritmem, který zjišťuje, zda jde o seznam jmen. Text je nejdříve rozdělen na čárkách, střednících a slově „and“ na jednotlivé kandidáty na jména. Poté je každá část ohodnocena podle toho, kolik obsahuje slov (jména mají 2–3, výjimečně 4 nebo 5 slov) a jestli všechny začínají velkým písmenem (kromě výjimek jako „von“). Výsledné ohodnocení textového uzlu je pak průměr ohodnocení jednotlivých takovýchto částí.

Všechny zkoumané texty jsou pak rozděleny podle průměrné hodnoty ohodnocení na seznamy jmen a části názvů přednášek. Sousední texty stejného druhu jsou sloučeny dohromady. Každému páru název–autoři je pak přiřazen poslední nalezený časový údaj a zaznamená se jako přednáška.

hCalendar

Kromě výše uvedené metody skript zpracovává i programy vyznačené pomocí mikroformátu hCalendar.

Mikroformáty jsou způsob, jak přidat sémantické informace do HTML souborů. Toho je dosaženo použitím XML atributu `class`, jehož hodnota označuje typ obsahu příslušné značky. Jejich výhoda spočívá v tom, že jsou tyto informace obsaženy přímo v dokumentu určeném pro lidské čtenáře.

Aktuální politika pořadatelů konferencí je zveřejňovat program pouze ve verzi pro lidské publikum. Takové dokumenty se, jak ukazuje předchozí sekce, dají automaticky zpracovat jen složitě. Na druhou stranu HTML stránka s vloženým mikroformátem poskytuje informace ve standardizované formě a se zachováním původní vizuální podoby.

Další informace jsou k dispozici v [4] a [11].

Pro účely vyznačení programu konference je vhodný formát hCalendar (`http://microformats.org/wiki/hcalendar`). Tento formát bohužel neumožňuje standardně vyznačit autory přednášek, ty proto používají nestandardní třídu `speaker`.

3.5 Použití jiných extrakčních skriptů

Pro extrakci informací lze samozřejmě použít i jiné programy než ten, který je součástí průvodce konferencí. V rámci GVP například probíhá projekt *Extrakce a zpracování informací ze seznamu publikací nebo programu konference*, který má také za cíl získávat informace o přednáškách. Tento projekt je však zaměřen spíše na nalezení odkazu na program přednášky na domovské stránce konference. Stejně jako průvodce konferencí však aktualizuje databázi, kterou je možno po drobných úpravách použít jako vstup pro plánování individuálního programu. Nutné úpravy spočívají v převodu časových informací: extrakční program ukládá datum a čas přednášky, zatímco průvodce konferencí vyžaduje začátek a konec přednášky. Byl proto napsán skript, který z data a času doplní začátek a konec všech přednášek v databázi, pokud chybí. Tím sice vznikají přednášky o nulové délce, ale průvodce konferencí byl navržen tak, aby je uměl zpracovat.

Mimo to lze samozřejmě využít i jiné programy, nebo například napsat zvláštní skript na extrakci jednoho konkrétního formátu. Pokud bude výsledek kompatibilní s požadovaným schématem databáze, není problém ho použít.

Kapitola 4

Navržení individuálního programu

Tato část práce se zabývá vlastním návrhem individuálního programu pro konkrétního uživatele. Nejdříve je popsán matematický model a algoritmus pro návrh takového programu, jsou-li všechny přednášky ohodnoceny. Poté se teprve text zaměří na způsoby, kterými lze vyhodnotit, jak přednáška odpovídá zájmům určitého uživatele. Nakonec je stručně popsán výstup programu.

4.1 Matematický model

Progam konference je možné chápat jako orientovaný acyklický graf, v němž vrcholy představují jednotlivé přednášky. Hrany grafu představují možné přesuny mezi místnostmi nebo setrvání v jedné místnosti mezi navazujícími přednáškami. Pro dva uzly A a B tudíž existuje hrana $A \to B$ tehdy a pouze tehdy, pokud přednáška, která odpovídá uzlu A, končí předtím, než začíná přednáška, která odpovídá uzlu B.

Pro účely formálních definicí se ke grafu přidá uzel „začátek konference“ Z, a „konec konference“ K, které jsou v této relaci minimální, resp. maximální.

Každá hrana grafu je ohodnocena podle hodnocení přednášky, která přísluší uzlu, do kterého vede.

Každá cesta v tomto grafu koresponduje s možným individuálnímu programem, tj. sekvencí přednášek, které může jeden člověk navštívit. Délka cesty odpovídá součtu hodnocení navštívených přednášek. Optimální program je pak nejdelší cesta grafem z uzlu Z do K.

Bylo by možné rozšířit hodnocení hran např. o náročnost přechodů mezi jednotlivými místnostmi, ale tato informace se obtížně získává automaticky, proto se jí zde nezabývám.

Je teoreticky možné, že by určitá přednáška byla ohodnocena zápornou hodnotou. To odpovídá případu, že by na danou přednášku bylo lepší vůbec nejít. Pokud je ale na výběr z několika záporně hodnocených přednášek, jedna z nich se přesto zahrne do programu – volbu, zda ji absolvovat nebo například počkat na chodbě, nechává skript na uživateli. Proto pokud je hodnocení přednášky záporné, nastaví se hodnota příslušných hran na 0.

To, že každá hrana má nezáporné hodnocení, znamená, že optimální program nemůže žádnou přednášku „přeskočit“, tj. nejdelší cesta z uzlu Z do uzlu K musí být obsažena v tranzitivní redukci grafu. Úloha pak spočívá v nalezení této nejdelší cesty.

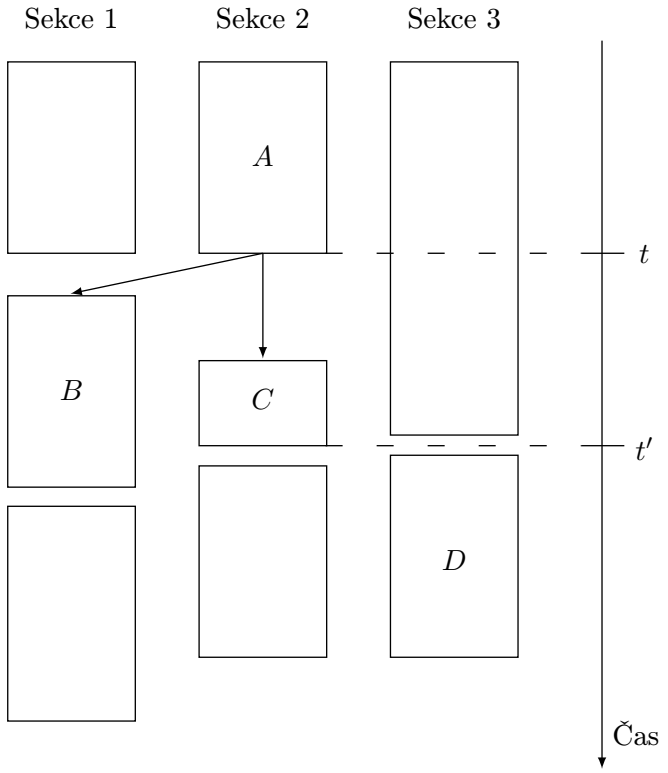
4.2 Algoritmus

Skript pro navržení optimálního individuálního programu používá algoritmus, který z databáze vybere tranzitivní redukci grafu a nalezne v něm nejdelší cestu v jednom průchodu.

Algoritmus využívá funkci další(t), která danému času přiřazuje množinu přednášek, mezi kterými se v daný čas může účastník konference rozhodovat. Jistě nemá cenu se zabývat přednáškami, které již proběhly, nebo právě probíhají – přijít na přednášku pozdě je z hlediska optimálního programu nepřípustné. Přednášky, které začínají právě v čase t, však do množiny zahrnuty jsou, protože v programech konferencí většinou nejsou uvedeny krátké přestávky mezi navazujícími přednáškami. Reálně tak přednášky začínají o něco později a končí o něco dříve, než je uvedeno v programu. Není však nutné zabývat přednáškami, které jsou příliš daleko v budoucnosti – přesněji těmi, před jejichž začátkem proběhne ještě jiná přednáška, kterou by účastník mohl stihnout.

Z hlediska matematického modelu jsou hodnoty této funkce uzly, do kterých vedou hrany z uzlu odpovídajícího přednášce, která končí v daném čase t. Pro čas $t = -\infty$ jsou to pak uzly, do kterých vedou hrany z počátečního uzlu.

Obrázek 4.2 ukazuje diagram konference se třemi paralelními sekcemi, na kterém je znázorněn způsob výpočtu hodnot této funkce. Přednášky jsou reprezentovány obdélníky. Chceme-li vypočítat další(t) pro $t = k_A$, konec přednášky A, nejdříve zjistíme nejdřívější čas konce přednášky, která začíná po nebo v čase t, a označíme jej t'. Hodnota další(t) je pak množina všech přednášek, které začínají v časovém intervalu $\langle t, t')$ – v našem příkladu $\{B, C\}$. Přednáška D již do další(t) nepatří, protože před ní může návštěvník stihnout přednášku C.



Obrázek 4.1: Funkce další

Samotný algoritmus používá seznam hran ke zpracování H a množinu závěrečných přednášek F.

Podobně jako v algoritmu uvedeného v sekci 2.4 si pro každý uzel x pamatujeme hodnotu $U(x)$, která je rovna nejdelší zatím nalezené cestě do tohoto uzlu, a odkud(x), což je uzel, který x v nejdelší cestě předchází.

1. Pro každé $x \in$ další$(-\infty)$ přidej do seznamu H hranu (Z, x).
2. Pro všechny $p \in P$ nastav hodnotu $U(p) = 0$.
3. Nastav $K = \emptyset$, a $U(Z) = 0$.
4. Dokud není H prázdný:
   - Vyber z H hranu (x, y) s nejnižším časem začátku přednášky y.
   - Pokud $U(y) < U(x) + h(y)$:
     - $U(y) := U(x) + h(y)$.
     - odkud$(y) := x$.
   - Pro každé $z \in$ další(k_y) přidej do H hranu (y, z).
   - Pokud další$(\mathrm{k}_y) = \emptyset$, přidej y do množiny F.

Algoritmus 4.1: Hledání optimálního individuálního programu

Optimální cesta pak končí v uzlu s nejvyšší hodnotou $U(x)$, který je v F, a zbytek cesty se dá najít sledováním hodnot odkud podobně jako u algoritmu 2.4.

Protože další(t) vždy obsahuje první přednášku, která začíná po čase t, a ze seznamu hran je vždy vybrána ta s nejnižším časem začátku přednášky, je zaručeno že se přednášky budou zpracovávat v pořadí podle času začátku, což je topologické uspořádání.

Protože v další(k_x) jsou obsaženy všechny uzly, do nichž vede hrana z uzlu x, postupným sledováním těchto hran navštíví všechny uzly dostupné z uzlu Z. Pracujeme s tranzitivní redukcí grafu, v němž jsou ze Z přímo dostupné všechny uzly, algoritmus tedy zpracuje všechny uzly grafu.

Jsou tedy v topologickém uspořádání zpracovány všechny hrany podle algoritmu 2.4, čímž zaručeno, že nalezená cesta bude opravdu nejdelší.

Pokud existuje nejdelších cest více, najde algoritmus jen jednu z nich. To v praxi není problém, protože není pravděpodobné, že dvě přednášky (a tedy dvě různé cesty) budou mít stejné hodnocení.

4.3 Ohodnocení jednotlivých přednášek

Důležitá součást průvodce konferencí je hodnocení přednášek. Vzhledem k tomu, že zatím nejsou k dispozici podrobné informace o jednotlivých osobách, hodnotí program přednášky podle dotazu. Dotaz slouží jako profil uživatele a měl by obsahovat klíčová slova či frázi vyjadřující jeho zájmy.

Problém vyhledávání relevantních informací je v oblasti zpracování přirozeného jazyka často řešený. Jeho aplikace jsou nejviditelnější ve webových vyhledávačích. Hledání relevantních přednášek na programu konference je však specifické tím, že jsou k dispozici jen názvy přednášek, které obsahují pouze několik málo slov.

Klasické techniky zjišťování podobnosti dokumentů založené na modelu vektorového prostoru využívají toho, že dokumenty o podobných tématech používají přesně stejnou slovní zásobu. U krátkých textů hrozí při použití těchto metod, že dotaz a název přednášky nebudou používat stejná slova. Musíme se proto zaměřit na podobnost významů jednotlivých termínů.

Pro hodnocení přednášek jsem využil metody podobnosti podle D. Lina a ESA, zmíněné v kapitole 2.

Trénovací data

Obě zvolené metody pro svou funkci potřebují učební texty, na kterých zakládají svůj model podobnosti mezi jednotlivými slovy.

Pro tento účel by měly být vybrány texty ze stejného oboru, z jakého jsou zpracovávané přednášky. Tím se zajistí, že výsledky budou odpovídat zvolenému tématu. Při použití obecného korpusu bychom se vystavovali nebezpečí, že použité metody sloučí konkrétní termíny používané v daném oboru dohromady, neboť jsou používány v jedné konkrétní skupině dokumentů.

Konkrétně byl použit archiv článků z oblasti zpracování přirozeného jazyka, který je k dispozici na FIT VUT. Vybrány byly jen dokumenty ve formátu čistého textu, celkem 1939 článků. Tabulka 4.3 ukazuje zdroje článků – většinou jsou ze sborníků ke konferencím.

Počet	Zdroj
291	LREC 2000
175	COLING 2000
41	GWN 2002
180	AAAI 2002
201	COLING 2002
144	COLING 2002 WS
372	LREC 2002
464	arxiv.org
71	Ostatní
1939	Celkem

Tabulka 4.1: Zdroje článků použitých jako vstupní korpus

Kvalita článků je ovlivněna tím, že jde většinou o texty převedené nedokonalými nástroji z formátu PDF. Zvláště články původně vysázené do dvou sloupců zachovávají tuto formu i po převodu do textu, takže věty, které začínají a končí na různých řádcích, jsou přerušeny obsahem sousedního sloupce. Další problém se vyskytuje v matematických výrazech, které po automatickém převodu z PDF do textu ztratí svoji strukturu a změní se v nepřehlednou změť čísel a znaků. Přesto je ale naprostá většina textů použitelná.

Podobnost podle D. Lina

První zvolená metoda zjišťování podobnosti názvů je podobnost podle D. Lina.

Jako základ implementace jsem použil skripty, které napsal Ing. Marek Schmidt a jsou popsány na `https://merlin.fit.vutbr.cz/nlp-wiki/index.php/HOWTO_Lin_Similarity`.

Tato metoda je založena na podobnosti jednotlivých slov.

Ze vstupních článků byla vybrána slova, která se ve všech textech vyskytovala více než pětkrát. Výskyty méně častých slov byly považovány za statisticky nevýznamné. Poté byla slova očíslována pomocí knihovny CMPH minimální perfektní hašovací funkcí vypočítanou pomocí algoritmu CHM. Knihovna a algoritmus jsou popsány v [10], resp. [1].

Texty článků z vývojové sady byly poté analyzovány programem MINIPAR, což je rychlý parser textů se širokým zaměřením, který extrahuje z částí textů závislostní stromy [6]. Bylo zjištěno a uloženo množství informace společné každému páru slov a vlastní každému slovu.

V rámci práce byly vytvořeny rozšiřující moduly pro jazyk Python, které umožňují MINIPAR a CMPH používat přímo, bez nutnosti spouštět externí proces.

Vzhledem k tomu, že tato metoda rozlišuje různé slovní druhy, je nutné použít na dotaz a jednotlivé názvy přednášek také použít parser, podobně jako na dokumenty z korpusu. K tomuto účelu je opět využit MINIPAR. Po označení slovních druhů je podle vztahu 2.1 spočtena podobnost pro každou kombinaci slova z dotazu a slova z názvu přednášky.

Zbývá ještě jednotlivé podobnosti sloučit do výsledného hodnocení. Zde jsem vyzkoušel tři různé metody: Lin_1 je součet hodnocení všech hodnocených párů dělený počtem slov názvu přednášky, Lin_2 je součet čtverců hodnocení všech hodnocených párů dělený počtem slov názvu přednášky, a Lin_∞ je průměrná hodnota maximálních hodnocení pro každé slovo názvu přednášky. Obecně:

$$\mathrm{Lin}_n(V, W) = \frac{1}{|V|} \sum_{v \in V} \sum_{w \in W} \mathrm{sim}(v, w)^n$$

$$\mathrm{Lin}_\infty(V, W) = \frac{1}{|V|} \sum_{v \in V} \max_{w \in W} (\mathrm{sim}(v, w))$$

kde W je vektor zadaných klíčových slov je a V vektor slov názvu přednášky.

Čím je hodnota n vyšší, tím vyšší váha je přidělena slovům s vyšší podobností. Varianta Lin_∞ bere pro každé slovo dotazu v úvahu jen nejpodobnější slovo z názvu přednášky.

Stojí za zmínku, že výsledky hodnocení lze porovnávat pouze pokud odpovídají stejnému dotazu a použité metodě, protože nejsou normalizovány.

Explicitní sémantická analýza

Pro hodnocení přednášek pomocí ESA jsem místo Wikipedie, jak je uvedeno v [3], použil výše uvedené články z oboru zpracování přirozeného jazyka.

Vědecké články mají s Wikipedií společné to, že jsou specializované a poměrně specifické. Na rozdíl od Wikipedie ale může existovat více článků na stejné téma, takže neplatí, že každému konceptu je přiřazen právě jeden článek. Na druhou stranu jsou ale články zaměřené konkrétně na oblast, ze které jsou i zpracovávané konference, takže mají potenciál poskytovat relevantnější výsledky.

Články byly nejdříve indexovány pomocí knihovny Xapian. Tím byl vytvořen reverzní index, na jehož základě pracuje sémantický interpretr podle [3]. Výstupem sémantického interpretru je kosinus úhlu, který svírají vektor příslušící zadaným klíčovým slovům (profilu uživatele) a názvu přednášky. Tato hodnota je použita jako hodnocení přednášky.

Na rozdíl od Linovy podobnosti pracuje ESA na úrovni celého názvu přednášky a všech klíčových slov. Výsledky jsou hodnoty v rozmezí $\langle 0, 1 \rangle$, které mezi sebou lze bez omezení porovnávat.

4.4 Výstup programu

Standardně poskytuje skript na výstupu individuální program jako seznam doporučených přednášek v chronologickém pořadí. Je ale možné generovat program konference ve formě HTML tabulky, ve kterém jsou přednášky barevně vyznačeny podle jejich hodnocení a doporučené přednášky – individuální plán konference – jsou dále zvýrazněny. Tento výstup používá mikroformát hCalendar, takže je jej možné použít i jako exportní formát pro samotnou konferenci.

Kapitola 5

Vyhodnocení výsledků

5.1 Vyhodnocení extrakce programu konference

Testovací data

Pro testování extrakce informací z programu konference byly použity programy konferencí v roce 2007, jejichž seznam je k dispozici na http://aclweb.org/aclwiki/index.php?title=Conferences_in_2007. Programy generované z výsledků extrakce informací byly porovnány s programy původními. U každého z nich byly spočteny následující údaje:

- Celkový počet přednášek v programu
  Za „přednášku“ zde počítám jak klasické přednášky, tak plakáty a semináře, které mají v programu uvedený název, čas a jméno nebo jména autorů.
- Počet extrahovaných přednášek
- Počet přednášek, u kterých byl správně rozpoznán čas začátku a konce
  Čas začátku je nejdůležitější údaj pro sestrojení modelu konference. Pokud se nedaří správně určit časy začátků přednášek, je obtížná i manuální kontrola výsledků extrakce. Některé programy přednášek uvádějí u přednášek pouze čas začátku. U těchto přednášek předpokládá extrakční skript, že končí se začátkem následující přednášky. Tento odhad však muže být mylný, zvláště u poslední přednášky dne.
- Počet přednášek, u kterých byl správně rozpoznán název a autoři
  Název přednášky je samozřejmě klíčový pro její hodnocení. Systém však někdy udělá chybu při určování, který z údajů uvedených v programu je název a který seznam autorů. Pokud se tato chyba vyskytne u přednášky, která má správně určený čas začátku a konce, nemělo by být obtížné ji objevit a opravit při manuální kontrole výsledků extrakce.

Pro úspěšnost rozpoznání času začátku a konce a názvu a autorů byly vypočítány hodnoty přesnosti a pokrytí. Přesnost (*precision*) je počet správně extrahovaných přednášek dělený celkovým počtem extrahovaných přednášek. Je to měřítko chyb typu I. Pokrytí (*recall*) je počet správně extrahovaných přednášek dělený celkovým počtem všech přednášek – měřítko chyb typu II.

Výsledky testování

Výsledky jsou uvedeny v tabulce 5.1.

Konference	Formát	Celkem přednášek	Extrahovaných přednášek	Správný začátek a konec	Přesnost [%]	Pokrytí [%]	Správný název a autoři	Přesnost [%]	Pokrytí [%]
AAAI-07	PDF	162	255	0	0	0	0	0	0
ACL 2007	HTML	139	184	134	72,8	96,4	129	70,1	92,8
EMNLP-CoNLL 2007	HTML	133	241	122	50,6	91,7	117	48,5	88,0
NAACL-HLT 2007	HTML	152	159	146	91,8	96,1	142	89,3	93,4
ICASSP 2007	HTML	†	192	0	—	—	0	—	—
Interspeech 2007	HTML	10†	26	6	—	—	3	—	—
MT Summit XI	PDF	59	106	50	47,2	84,7	31	29,2	52,5
PACLING 2007	HTML	40	43	40	93,0	100,0	36	83,7	90,0
RANLP 2007	PDF	18	0	0	0	0	0	0	0
RIAO-07	HTML	⋆	0	0	—	—	0	—	—
TIME 2007	HTML	†	0	0	—	—	0	—	—
TMI-07	HTML	31	0	0	0	0	0	0	0

Tabulka 5.1: Úspěšnost hodnocení přednášek

Protože se téměř u každého zpracovávaného programu objevily jiné problémy, má smysl rozepsat výsledky jednotlivých testů.

Konference označené † mají program ve formě HTML obsahu s odkazy na jednotlivé sekce. S tímto formátem extrakční skript nepočítá, a proto nebyl schopen informace získat. V případě konference Interspeech jsou některé klíčové přednášky uvedeny přímo v obsahu, vzhledem k počtu ostatních přednášek jsou však tyto zanedbatelné.

Konference RIAO, označená ⋆, neuvádí u přednášek časy začátku ani konce. Skript proto přednášky nedetekoval. Pro účely průvodce konferencí je takový program nepoužitelný.

Program AAAI 2007 vypadá podobně jako u ročníku 2008, diskutovaném v sekci 3.3. Je to PDF dokument vysázený do tří sloupců, obsahující kromě programu i další informace, jehož struktura se převodem do HTML ztratila. Přednášky v něm proto skript nemohl identifikovat.

U konference MT Summit XI se při extrakci několikrát stalo, že skript sloučil několik přednášek do jedné. To bylo způsobeno hlavně nedostatečně zachovanou strukturou dokumentu. Vzhledem k tomu, že jde o program vysázený do tří sloupců (které odpovídají třem sekcím), považuji však dosažené výsledky za překvapivě dobré.

Program konference EMNLP-CoNLL je zvláštní v tom, že obsahuje nejdřív časový rozvrh se všemi požadovanými informacemi, a poté následuje detailní seznam přednášek s abstrakty, kde jsou údaje zopakovány. Skript proto většinu přednášek extrahoval dvakrát, což odpovídá faktu, že hodnota přesnosti je zhruba polovina pokrytí.

Tři konference s nejlepšími dosaženými výsledky – ACL, NAACL-HLT a PACLING – poskytují programy ve formátu HTML tabulek, se kterými nemá extrakční skript větší problémy. Za zmínku stojí fakt, že tyto tři konference jsou organizovány v paralelních sekcích.

U poslední konference, TMI, skript nerozeznal formát dat (ve smyslu určení dnů). Tato chyba, která by mohla být velice snadno opravena, ukazuje na obecnou nevýhodu *ad hoc* metod extrakce informací – pokud se objeví jiná data, než podle jakých byl vytvořen, nemusí systém fungovat správně.

5.2 Úspěšnost hodnocení přednášek

Největší problém hodnocení optimálního individuálního programu, vygenerovaného podle profilu uživatele reprezentovaného klíčovými slovy, spočívá v získání testovacích dat, podle kterých by bylo možné úspěšnost automaticky vyhodnotit.

Testovací data

Některé konference, např. COLING, mají přednášky rozděleny do pojmenovaných sekcí, např. „Parsing“ nebo „Information Retrieval“. Tyto názvy sekcí mají stejnou funkci jako klíčová slova v dotazech – popisují sadu přednášek, které se týkají dané problematiky, stejně jako dotazy popisují problematiku, která zajímá daného uživatele. Problém s použitím názvů sekcí spočívá v tom, že je uvádí jen minimum konferencí, a navíc každá v jiném formátu, což komplikuje jejich automatickou extrakci. Proto jsem pro testování úspěšnosti hodnocení vybral seznam seminářů CEUR (http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/), který obsahuje informace o více než 6000 seminářích seřazených podle témat.

Jednotlivé semináře jsou z hlediska tohoto projektu velice podobné přednáškám – mají svůj název a autory a většinou jsou založeny na nějaké vědecké práci. Hlavní rozdíl mezi tématy seminářů a konferencemi je ten, že semináře bývají menší (typicky do dvaceti akcí). Z programů na stránkách CEUR lze také relativně snadno automaticky získat názvy jednotlivých seminářů, protože jejich název je uveden jako odkaz na odpovídající publikaci ve formátu PDF nebo PS.

Drobnou nevýhodou seminářů ze seznamu CEUR je to, že některá témata jsou v němčině. Témata obsahující některá častá německá slova byla proto vyřazena.

Při vlastním testování bylo každé z témat bráno jako dotaz a na jeho základě byly ohodnoceny všechny semináře. Výsledkem byly dva seznamy hodnocení, jeden pro semináře patřící do testovaného tématu, druhý pro ostatní.

Metoda vyhodnocení

Hodnocení přednášky je úspěšné, pokud jsou doporučeny takové přednášky, které doporučeny být mají. Systém doporučí vždy přednášku s nejvyšším hodnocením. Jeden způsob vyčíslení úspěšnosti hodnocení je tedy zjistit, jak často systém doporučí správnou přednášku – v našem případě seminář, který patří do tématu daného dotazem. Pro dané téma proto rozdělíme všechny semináře do dvou skupin: na ty, které do daného tématu patří, a ty ostatní. Poté porovnáme hodnocení každého semináře z první skupiny s hodnocením každého semináře z druhé skupiny. Pokud je hodnocení z první skupiny vyšší, rozhodl se systém správně. Pokud je nižší, doporučil by systém špatnou přednášku. Pokud jsou hodnocení stejná, systém se nemůže rozhodnout. Takové výsledky z hodnocení vyřadíme (v praxi se příliš často nevyskytují). Úspěšnost hodnocení se pak pro dané téma dá vyhodnotit jako:

$$\frac{\text{počet správně doporučených seminářů}}{\text{počet správně doporučených seminářů} + \text{počet špatně doporučených seminářů}}$$

Výsledky vyhodnocení

Tabulka 5.2 ukazuje střední hodnotu a směrodatnou odchylku úspěšnosti jednotlivých metod. Navíc je zde pro porovnání uvedena úspěšnost náhodného hodnocení, které každé přednášce přiřadí náhodné číslo s rovnoměrným rozložením.

Metoda	Střední hodnota úspěšnosti	Směrodatná odchylka
Lin_1	54,8%	30,3%
Lin_2	57,9%	27,2%
Lin_∞	58,2%	25,1%
ESA	61,1%	26,8%
Náhodné hodnocení	51,0%	9,1%

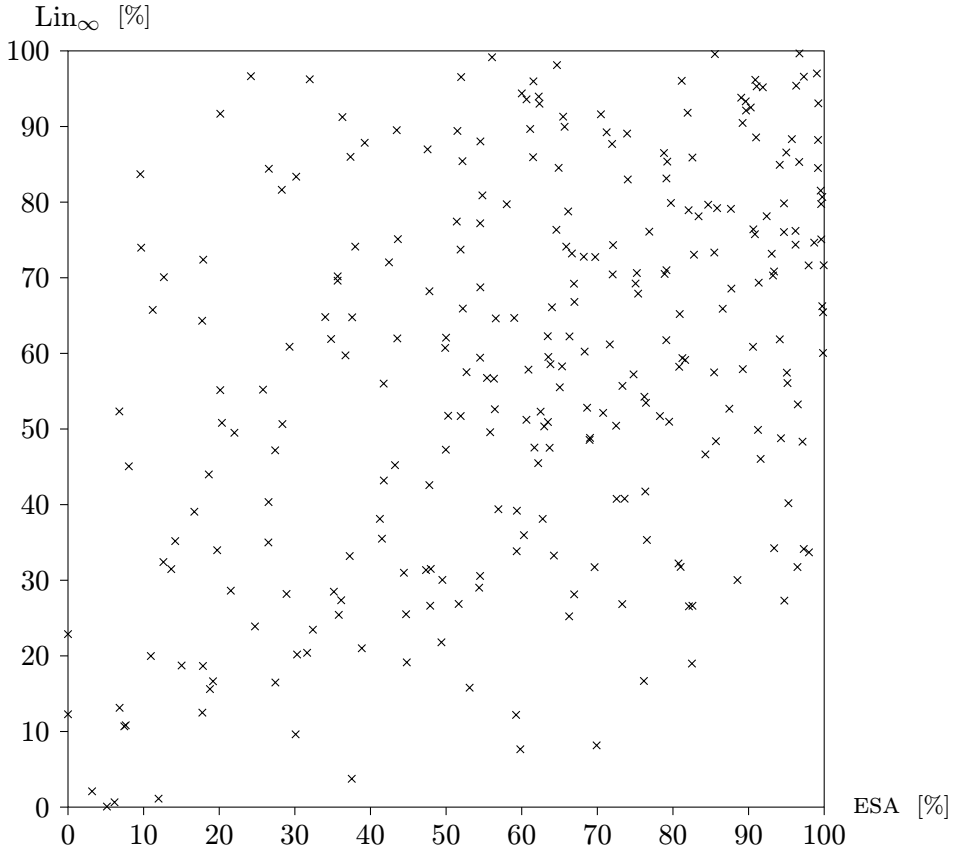
Tabulka 5.2: Úspěšnost hodnocení přednášek

Z tabulky je patrné, že průměrná úspěšnost jednotlivých metod je srovnatelná s náhodným hodnocením. Vysoká hodnota směrodatné odchylky však ukazuje, že je úspěšnost hodnocení silně závislá na konkrétním tématu. Některé názvy témat jsou jako klíčová slova vhodnější než jiné. Poznamenám, že témata, která jsou úspěšně hodnocena metodou podobnosti podle Lina, bývají úspěšnější i při použití metody ESA, i když tento trend není nijak výrazný. Korelace úspěšností těchto metod na jednotlivých tématech je rovna 0,425 a graficky je znázorněna na obr. 5.2.

V tabulkách v příloze B jsou uvedeny názvy témat, pro která byly jednotlivé použité metody nejméně úspěšné, resp. nejúspěšnější.

Ukazuje se, že názvy seminářů s nejnižší úspěšností nepopisují jejich obsah. Typickým příkladem jsou zkratky, např. „NIR'96“ nebo „VLDB PhD Workshop 2003“. Naopak úspěšnější jsou témata s velmi specifickými názvy. Toto chování je očekávatelné a žádané – pokud uživatel zadá konkrétní klíčová slova, může systém přesněji najít odpovídající přednášky. Naopak je-li zadán obecný dotaz, výsledky systému s ním nemůžou korespondovat přesně.

Jako nejúspěšnější vyšla ze srovnání metoda ESA. Z metod založených na Linově podobnosti je pak nejúspěšnější Lin_∞. Rozdíly úspěšností však nejsou významné.



Obrázek 5.1: Souvislost mezi úspěšnostmi hodnotících metod Lin_∞ a ESA pro jednotlivá testovaná témata

Kapitola 6

Závěr

Byly zkoumány dva přístupy získávání sémantických informací z textu – *ad hoc* systém pro extrakci strukturovaných údajů a dvě formální metody odhadu podobnosti významů textů, založené na podobnosti podle D. Lina a explicitní sémantické analýze. Tyto metody byly použity pro plánování individuálního programu konference v rámci projektu Generátoru vědeckých portálů.

Úspěšnost extrakce programu přednášek z HTML tabulek, převládajícího formátu pro menší konference s paralelními sekcemi, přesahuje 90%. U ostatních formátů je však úspěšnost citelně nižší – někdy systém správně nezjistí ani jednu přednášku.

Pokud je program správně extrahován, záleží úspěšnost systému na volbě klíčových slov v profilu uživatele. I při dobré volbě je však nutné navržený program brát jen jako doporučení, protože úspěšnost hodnotících metod je relativně malá.

Jádro systému – sestavení individuálního plánu podle známého hodnocení – však funguje dobře. V budoucnu bude tedy možné použít sofistikovanější metody extrakce programu a hodnocení přednášek, které mohou úspěšnost zvýšit. Jedno z možných vylepšení je pro získávání informací z programů použít dvouprůchodový systém, který nejdřív zjistí formát dat a poté je teprve získá. Úspěšnost uvedených hodnotících metod se pravděpodobně dá zvýšit výběrem kvalitnějších trénovacích dat. Největším přínosem pro systém by však bylo jeho reálné nasazení a zkoušení na lidech.

Jako součást GVP je průvodce konferencí ovlivněn ostatními částmi tohoto systému. Související projekty jsou *Zjednotenie návrhu databázy a dotazovací systém* Petera Bartoše a *Extrakce a zpracování informací ze seznamu publikací nebo programu konference* Petra Drastila a Magdaléné Krygielové.

Osobně mi projekt přinesl mnoho nových poznatků z oblasti zpracování přirozeného jazyka a hlubší seznámení s programovacím jazykem Python.

Literatura

[1] Czech, Z. J.; Havas, G.; Majewski, B. S.: An Optimal Algorithm for Generating Minimal Perfect Hash Functions. *Information Processing Letters*, 1992.

[2] Demel, J.: *Grafy a jejich aplikace*. Academia, 2002, s. 85–108, ISBN 80-200-0990-6.

[3] Gabrilovich, E.; Markovitch, S.: Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis [online]. http://www.cs.technion.ac.il/~gabr/papers/ijcai-2007-sim.pdf, [cit. 2009-04-23].

[4] Khare, R.; Çelik, T.: Microformats: a pragmatic path to the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, New York, NY, USA: ACM, 2006, ISBN 1-59593-323-9, s. 865–866, doi:http://doi.acm.org/10.1145/1135777.1135917.

[5] Lin, D.: An Information-Theoretic Definition of Similarity. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, ISBN 1-55860-556-8, s. 296–304.

[6] Lin, D.: LaTaT: language and text analysis tools. In *HLT '01: Proceedings of the first international conference on Human language technology research*, Morristown, NJ, USA: Association for Computational Linguistics, 2001, s. 1–6, doi:http://dx.doi.org/10.3115/1072133.1072198.

[7] Manning, C. D.; Schütze, H.: *Foundations of Statistical Natural Language Processing*. MIT Press, 1999, ISBN 0-262-13360-1.

[8] Wiki, N.: Generátor vědeckých webových portálů. https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Generátor_vědeckých_webových_portálů, [cit. 2009-05-02].

[9] Wikipedia: Type I and type II errors [online]. http://en.wikipedia.org/wiki/Type_I_and_type_II_errors, [cit. 2009-04-03].

[10] WWW stránky: CMPH – C Minimal Perfect Hashing Library. http://cmph.sourceforge.net/.

[11] WWW stránky: Microformats. http://microformats.org/about/.

Dodatek A

Obsah CD

Na přiloženém CD je kompletní implementace a dokumentace k ní, skripty a data použité k testování, elektronická podoba tohoto textu se zdrojovými soubory, a plakát.

Kompletní obsah CD je uveden v souboru Readme.

Dodatek B

Další tabulky

Úspěšnost je uváděna v procentech.

Úspěšnost	Téma
0,00	NIR'96.
0,00	Föderierte Datenbanken.
3,28	Workshop on Applications of Description Logics 2002.
5,15	LSTKM 2004.
6,15	MBUI 2004.
6,79	Mastering the Gap 2006.
6,82	BPM Demonstration Program 2007.
7,44	Verification Workshop 2007.
7,64	CAiSE Forum 2003.
8,04	Behaviour Monitoring and Interpretation 2007.
⋮	
99,18	Semantic and Dynamic Web Processes 2005.
99,22	Knowledge Markup and Semantic Annotation 2004.
99,54	Model Driven Development of Advanced User Interfaces 2007.
99,60	Description Logics 1998.
99,61	Information Systems Implementation and Modelling 2006.
99,74	Electronic Personal Health Records 2006.
99,75	Model Driven Development of Advanced User Interfaces 2006.
99,84	Semantic Web Workshop 2001.
99,85	Web Information Systems Modeling 2006.
99,93	Intelligent Access to Heterogeneous Information.

Tabulka B.1: Témata, pro která byla a nejméně a nejvíce úspěšná metoda ESA

Úspěšnost	Téma
0,12	LSTKM 2004.
0,25	MBUI 2004.
0,48	Föderierte Datenbanken.
0,94	MIXER 2004.
1,68	Knowledge Web PhD Symposium 2007.
2,91	Virtual Campus 2006.
3,08	CAiSE Workshop Proceedings 2003.
3,48	Knowledge Representation meets Databases 2002.
3,56	CAiSE Forum 2005.
4,16	Knowledge Representation meets Databases 2001.
⋮	
98,81	Modeling and Retrieval of Context 2005.
98,99	Design and Management of Data Warehouses 2001.
99,08	Social Information Retrieval for Technology-Enhanced Learning & Exchange 2007.
99,47	Ontology, Conceptualizations and Epistemology for Software and Systems Engineering 2005.
99,48	Enterprise Modelling and Ontologies for Interoperability 2004.
99,86	Semantic Intelligent Middleware for the Web and the Grid 2004.
99,89	Representation and Analysis of Web Space 2005.
99,94	Emerging Research Opportunities for Web Data Management 2007.
99,96	Uncertainty Reasoning for the Semantic Web 2006.
99,97	New Forms of Reasoning for the Semantic Web: Scalable, Tolerant and Dynamic 2007.

Tabulka B.2: Témata, pro která byla a nejméně a nejvíce úspěšná metoda Lin_1

Úspěšnost	Téma
0,27	LSTKM 2004.
0,44	MBUI 2004.
0,50	MIXER 2004.
1,43	Knowledge Web PhD Symposium 2007.
1,96	Workshop on Applications of Description Logics 2002.
2,23	CAiSE Workshop Proceedings 2003.
4,93	Virtual Campus 2006.
5,07	Verification Workshop 2007.
6,26	NIR'96.
6,40	Ontologies and Information Sharing 2001.
⋮	
97,45	Desarrollo Dirigido por Modelos - MDA y Aplicaciones 2006
98,56	Modeling and Retrieval of Context 2005.
98,73	Enterprise Modelling and Ontologies for Interoperability 2004.
99,16	Uncertainty Reasoning for the Semantic Web 2006.
99,20	Emerging Research Opportunities for Web Data Management 2007.
99,51	Metadata Management in Grid and P2P Systems 2003.
99,61	Representation and Analysis of Web Space 2005.
99,69	Semantic Web Case Studies and Best Practices for eBusiness 2005.
99,80	New Forms of Reasoning for the Semantic Web: Scalable, Tolerant and Dynamic 2007.
99,93	Semantic Intelligent Middleware for the Web and the Grid 2004.

Tabulka B.3: Témata, pro která byla a nejméně a nejvíce úspěšná metoda Lin_2

Úspěšnost	Téma
0,08	LSTKM 2004.
0,64	MBUI 2004.
1,12	MIXER 2004.
2,09	Workshop on Applications of Description Logics 2002.
3,75	CAiSE Workshop Proceedings 2003.
7,65	Knowledge Web PhD Symposium 2007.
8,17	Ontologies and Information Sharing 2001.
9,63	Enterprise Application Integration 2006.
10,68	Verification Workshop 2007.
10,82	CAiSE Forum 2003.
⋮	
96,17	Semantic Business Process and Product Lifecycle Management 2007.
96,24	OWL: Experiences and Directions 2006.
96,54	Scripting for the Semantic Web 2005.
96,59	Enterprise Modelling and Ontologies for Interoperability 2004.
96,65	Infrastructure for Dynamic Business-to-Business Service Outsourcing.
97,02	Interoperability of Web-Based Educational Systems 2005.
98,12	Modeling and Retrieval of Context 2005.
99,16	Metadata Management in Grid and P2P Systems 2003.
99,58	New Forms of Reasoning for the Semantic Web: Scalable, Tolerant and Dynamic 2007.
99,66	Semantic Intelligent Middleware for the Web and the Grid 2004.

Tabulka B.4: Témata, pro která byla a nejméně a nejvíce úspěšná metoda Lin_∞