Západočeská univerzita v Plzni
Katedra informatiky a výpočetní techniky
Univerzitní 8
306 14 Plzeň

Znalostní nástroje pro analýzu a prohledávání Webu

Odborná práce ke státní doktorské zkoušce

Martin Dostal

Technical Report No. DCSE/TR-2011-01
Květen, 2011

Technical Report No. DCSE/TR-2011-01
Květen, 2011

Znalostní nástroje pro analýzu a prohledávání Webu

Martin Dostal

Abstrakt

Tato práce je věnována problematice vyhledávání a extrakce informací z textových bloků webových stránek. Je zaměřena na nesupervizované metody pro extrakci informací založené zejména na statistických a grafových přístupech. Ze statistických metod se v současnosti nejvíce používá TFIDF a jeho modifikace pro přiřazení klíčových slov k analyzovanému textu. Extrakci klíčových slov lze zdokonalit rozšířením o sémantickou informaci, kterou lze získat z linked data. Klíčová slova tak získávají podobu štítků, které jsou definovány jménem, popisem, URI a vztahem k ostatním URI zdrojům. Mezi hlavní výhody využití linked data patří i jazyková nezávislost. Vyhledávání tak lze relativně snadno rozšířit o jazykově nezávislé vyhledávání. Na základě vztahů mezi URI zdroji lze využít grafové algoritmy pro nalezení skrytých znalostí, což umožňuje další zpřesnění vyhledávání i odvozování znalostí. Může se jednat např. o souvislosti mezi články nebo disambiguaci pojmů. Návrh těchto metod bude cílem disertační práce.

Copies of this report are available on
http://www.kiv.zcu.cz/publications/
or by surface mail on request sent to the following address:

University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitni 8
30614 Pilsen
Czech Republic

Obsah

1. Úvod 2
2. Klasifikace 4
2.1 Naivní Bayesovský klasifikátor 5
2.1.1 Metody výběru vlastností pro klasifikaci s využitím NB 6
2.2 Náš klasifikátor inspirovaný NB 6
2.3 Rocchio klasifikace 7
2.4 Hierarchická klasifikace 8
2.4.1 Vyhodnocení hierarchické klasifikace na základě podobnosti kategorií 10
2.4.2 Vyhodnocení hierarchické klasifikace na základě vzdálenosti kategorií 11
3. Shlukování 13
3.1 Jednoúrovňové shlukování 13
3.1.1 Shlukování založené na cestě 14
3.2 Hierarchické shlukování 15
3.3 Použití shlukování 16
4. Linked Data 18
4.1 Prohlížení Linked Data 19
5. Štítkování 21
5.1 Hlavní problémy se štítkováním 21
5.2 Relevantní systémy využívající štítkování 22
5.3 Naše technika štítkování s využitím Linked Data 23
5.4 Shlukování štítků 24
6. Závěr a budoucí práce 26
6.1 Metody pro zpracování jednoho dokumentu 26
6.2 Metody pro manipulaci s více dokumenty (korpusy) 27
Citovaná literatura 28
Aktivity 33

1. Úvod

Každým dnem se jen v České republice objevuje na Internetu několik desítek tisíc nových dokumentů, které svojí existencí ovlivňují značnou část populace. Schopnost vyhledat informace je zásadní pro úspěch firmy i fyzické osoby. Na základě údajů Českého statistického úřadu [1] z roku 2010 se za poslední 4 roky zdvojnásobil počet firem prezentující své služby a produkty na webových stránkách na současných 76%. 86% podniků v České republice má k dispozici vysokorychlostní Internet, který je v 89% firem používán za účelem obsluhy internetového bankovnictví a v 32% ke vzdělávání a školení zaměstnanců formou e-learningu. Mezi uživatele osobního počítače ve věku 16 – 74 let lze zařadit 69% národa. Internet využívá 66% obyvatel zejména k vyřizování emailů a obsluze internetového bankovnictví. Je poměrně překvapivé, že 57% jednotlivců používá Internet téměř každý den. Přestože se dle ČSÚ mluví o využívání Internetu, ve většině případů je to synonymem pro práci s webovými stránkami.

Česká republika se svými zákony dlouhodobě snaží zajistit občanům přístup k informacím v elektronické podobě zejména z důvodu úspor, které toto řešení problematiky svobodného přístupu k informacím přináší. Všechny obce s rozšířenou působností musí provozovat vlastní webové stránky, na kterých 98,8% z nich zveřejňuje rozvojový plán, 76% informace o pracovních místech a 41% poskytuje kontaktní formulář. Mnohé dokumenty jsou zveřejněny zejména na základě zákona 106/1999 Sb. o svobodném přístupu k informacím, který požaduje zveřejnění informací všemi státními a městskými orgány. Dále se může jednat např. o zákon 111/1998 Sb., o vysokých školách upřesňující podmínky pořádání výběrových řízení. V rámci zákona se však neřeší problematika dosažitelnosti dokumentů. Prakticky je tak velmi náročné požadované informace nalézt.

Obecné vyhledávací nástroje jako např. Google mohou být velkým pomocníkem, avšak díky optimalizaci webových stránek pro vyhledávače (SEO) se tak stále častěji musíme probírat nevhodnými výsledky, než kvalitními daty. Státní správa většinou do správného indexování neinvestuje čas ani prostředky a omezuje se tak pouze na publikaci informací bez možnosti jejich lepšího dosažení. Problematika správného nalezení požadovaných informací se v dnešní době jeví jako kritický problém, jehož i částečné řešení může být velmi prospěšné.

V této práci se budeme zabývat problematikou získávání informací z textových dat. Nejdříve si přiblížíme techniky extrakce informací z lokálního korpusu článků, následně se zaměříme na vyhledávání a analýzu informací v prostředí Webu.

V případě technik nad lokálními daty si nejdříve v kapitole 2 vysvětlíme základní principy ploché i hierarchické klasifikace dokumentů. Stručně si představíme Rocchio algoritmus, který je možné využít pro jednoúrovňovou i hierarchickou klasifikaci dokumentů. Představíme si pokročilé metody evaluace hierarchické klasifikace dokumentů s ohledem na podobnost, nebo vzdálenost kategorií. V kapitole 3 si přiblížíme existující metody shlukování a možnosti tvorby hierarchické struktury shluků. Ukážeme si možná řešení problematiky pojmenování shluků. Shlukování může být použito pro třídění výsledků vyhledávání, nebo

jako nástroj filtrování výsledků. V kapitole 4 si zopakujeme základní principy linked data dle T. B. Leeho a naznačíme si možnosti využití v rámci metod pro zpracování dat. V kapitole 5 si představíme základní myšlenku štítkování a důvod jeho rychlého rozšíření v prostředí Internetu. Neopomeneme ani základní problémy spojené se štítkováním a jejich řešení. V kapitole 6 si představíme cíl další práce zaměřené na vytvoření metod řešící základní problémy klasifikace a shlukování s využitím dodatečné sémantické informace extrahované z linked data.

2. Klasifikace

Klasifikace nebo také známá jako kategorizace je technika zařazení objektu, v našem případě dokumentu, do jedné nebo více klasifikačních tříd. Prakticky je možné klasifikaci používat k automatické detekci spamu, třídění emailů nebo např. detekci názoru dokumentu. Lze tak rozhodnout, zda dokument obsahuje pozitivní, nebo negativní postoj. Mnohé klasifikační úlohy byly tradičně vykonávány manuálně. Zejména v knihovnách je každá kniha manuálně zařazována do vybraných kategorií dle interních směrnic, avšak tato činnost je velmi časově a tedy i finančně nákladná. S rostoucím počtem elektronických dokumentů se zvyšují požadavky na plně automatickou klasifikaci, která je z výkonnostních i finančních důvodů mnohem výhodnější. Mezi hlavní nevýhody automatických metod patří obecně nižší přesnost zvláště v případě komplikovanějších úloh.

Metody klasifikace lze rozdělit do dvou základních kategorií:

- **Plochá klasifikace** – všechny kategorie jsou izolované bez jakýchkoliv vztahů mezi nimi. Tímto typem klasifikace se zabývají např. publikace [2] a [3].
- **Hierarchická klasifikace** – kategorie vytváří stromovou strukturu. Problematikou hierarchické klasifikace se zabývají např. publikace [4] [5] [6] a [7].

Vstupem klasifikačního procesu dle [8] je:

- Popis dokumentu $d \in X$, kde X je prostor dokumentu,
- množina klasifikačních tříd $C = \{c_1, c_2, .. , c_j\}$,
- trénovací množina D obsahující označené dokumenty $(d, c) \in X \times C$.

S využitím trénovacího algoritmu se snažíme najít takovou klasifikační funkci γ, která bude mapovat dokumenty na klasifikační třídy:

$$\gamma : \mathrm{X} \rightarrow C \tag{2.1}$$

Po vytvoření klasifikátoru γ je třeba ověřit jeho přesnost a úplnost. K tomuto účelu využijeme testovací množinu dokumentů, která má prázdný průnik s trénovací množinou.

K vyhodnocení správnosti klasifikace se v dnešní době nejvíce používají míry přesnosti a úplnosti. Tyto míry však nemohou zachycovat výkonnost klasifikace v případě jejich izolace. Z tohoto důvodu je správnost klasifikace vyhodnocována na základě jejich kombinace. Nejčastěji se používá:

- **F-míra** – tato míra byla navržena Rijsbergenem [9]. Její hodnota je určena na základě kombinace přesnosti P a úplnosti R. Zajímavé je, že v rámci výpočtu umožňuje zvolit důležitost přesnosti a úplnosti v podobě koeficientu β. Lze jí určit s využitím vzorce:

$$F_\beta = \frac{(\beta^2+1).P.R}{\beta^2.P+R}; \beta \in [0, \infty). \tag{2.2}$$

- **Break-Even Point (BEP)** – tato míra definuje bod, kde jsou si přesnost a úplnost rovny. Je však třeba vzít v úvahu, že někdy není možné tohoto bodu dosáhnout. Např. v případě správné klasifikace pouze několika dokumentů a špatné klasifikace mnoha dalších. V tomto případě přesnost nebude nikdy schopna dosáhnout shodné hodnoty s úplností. Ve všech ostatních případech je však tato míra velmi dobrá pro porovnání odlišných metod.
- **Průměrná 11-bodová přesnost** – přesnost se určuje v 11 bodech, kde úplnost dosahuje hodnot 0; 0,1 ; .. ; 1.

2.1 Naivní Bayesovský klasifikátor

Mezi nejjednodušší supervizované metody patří pravděpodobnostní metoda označovaná jako Naivní Bayesovský klasifikátor [8], zkráceně NB. Tato metoda byla i přes svoji jednoduchost označena jako velmi efektivní v porovnání s ostatními metodami [3]. Pravděpodobnost zařazení dokumentu *d* do klasifikační třídy *c* lze určit jako:

$$P(c \mid d) = P(c) \prod_{1 \le k \le n_d} P(t_k \mid c), \tag{2.3}$$

kde $P(t_k \mid c)$ je podmíněná pravděpodobnost existence termu t_k v dokumentu náležícího do klasifikační třídy *c*. Cílem klasifikace je výběr nejvhodnější třídy pro dokument. Tuto třídu zvolíme s využitím vzorce:

$$c_{map} = \arg\max_{c \in C} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k \mid c), \tag{2.4}$$

kde $\hat{P}(c)$ a $\hat{P}(t_k \mid c)$ je odhad hodnot na základě trénovací množiny. V průběhu praktického výpočtu se využívá vzorce pro výpočet logaritmu:

$$\log(xy) = \log(x) + \log(y), \tag{2.5}$$

což algoritmus Naivního Bayese zjednoduší na výběr třídy s nejvyšší hodnotou logaritmu a součin ve vzorci nahradí součtem.

Z předchozího vzorce (2.4) vidíme, že na zařazení dokumentu do klasifikační třídy mají vliv všechna slova, což v případě delších dokumentů není zcela efektivní. Pokud by se nám podařilo výrazně snížit počet termů, které použijeme pro výběr klasifikační třídy, dosáhli bychom tím značného algoritmického zlepšení [6]. Tímto problémem se zabývají metody výběru vlastností (feature selection).

2.1.1 Metody výběru vlastností pro klasifikaci s využitím NB

Cílem těchto metod je snížení počtu termů, které použijeme pro výběr klasifikační třídy a zvýšení přesnosti klasifikace díky odstranění rušivých vlastností. Výsledkem je tak výběr pouze termů s nevětším významem pro dokument. Mezi nejznámější techniky výběru nejdůležitějších termů patří:

- Vzájemná informace (mutual information),
- X^2 test,
- frekvence termů,
- Koller & Sahami feature selection [10].

2.2 Náš klasifikátor inspirovaný NB

Nyní si stručně popíšeme naší metodu klasifikace inspirovanou NB [11]. Cílem této úlohy byla klasifikace událostí nalezených v textech emailů. Jednalo se o klasifikaci krátkých popisků do předem stanovených klasifikačních tříd.

Mezi nejvýznamnější rozdíly, oproti ostatním klasifikátorům, patří výběr vlastností pro definici klasifikační třídy. Každá klasifikační třída je definována v podobě klíčových slov, která tuto třídu potvrzují, nebo vyvrací. V případě NB jsou klasifikační třídy určovány s využitím podmíněné pravděpodobnosti. Každé slovo testovaného dokumentu, nebo vybraného vektoru, tedy může klasifikační třídu doporučovat více, či méně. Avšak téměř nikdy ji zcela jasně nevylučuje. V našem případě je tomu jinak. Další důležitou vlastností je využití operace součtu, místo součinu hodnot jako v případě NB. Byla zvolena tato kritéria na klasifikační funkci:

1) Vytvoříme funkci vracející celočíselné ohodnocení textu pro každou klasifikační třídu. Celkové ohodnocení textu je dáno jako součet ohodnocení pro všechna vstupní slova. Každé slovo má určené ohodnocení dle jeho významu ke klasifikační třídě. Toto ohodnocení bylo definováno v rámci definice klasifikačních tříd. Pozitivní ohodnocení slova danou klasifikační třídu potvrzuje, negativní vyvrací.

2) Text zařadíme do třídy c s nejvyšším ohodnocením h určeným, kde T je množina slov vstupního textu, s_i slovo z T a funkce v vrací ohodnocení slova dle jeho definice v rámci klasifikační třídy.

$$h(T \mid c) = \sum_{i=1}^{n} v(s_i \mid c). \tag{2.6}$$

Funkce *v* je definována jako:

- k_1 pro $\forall s_i \in c$,
- k_2 pro $\forall s_i \in E$, kde *E* je množina klíčových slov vylučujících třídu *c*,
- k_3 pro $\forall s_i : \exists w_i \in c$, kde w_i je podřetězec s_i.

Konstanty byly experimentálně nastaveny následovně: k_1=1, k_2=-10, k_3=0.5. Konstanta k_3 charakterizuje pouze nalezení klíčového slova v podobě podřetězce analyzovaného slova.

Výsledný klasifikátor dle [11] splňuje všechna stanovená kritéria:

$$classify(T) = \arg\max_{c} \sum_{i=1}^{n} v(s_i \mid c). \tag{2.7}$$

2.3 Rocchio klasifikace

Rocchio klasifikace [8] [12] lze zařadit do kategorie klasifikací ve vektorovém prostoru. Klasifikační třídy (regiony) jsou odděleny hranicemi. Prvním krokem této klasifikace je určení hranic, které oddělují jednotlivé regiony. Testovaný dokument je zařazen do klasifikační třídy s nejmenším rozdílem jeho vektoru a vektoru klasifikační třídy.

Každý dokument je reprezentován jako vektor $\vec{v}_i$ obsahující váhy pro všechny, nebo vybrané termy. K volbě důležitých termů lze využít stejné metody jako v případě Naivního Bayese. K určení vah jednotlivých termů se v rámci vektoru reprezentujícího dokument často využívá TF-IDF skóre [13]. Jedná se o součin frekvence termu a jeho inverzní frekvence v dokumentech. Frekvenci termu lze určit s využitím vzorce:

$$tf = \frac{n_{ij}}{\sum_k n_{kj}}, \tag{2.8}$$

kde n_{ij} je počet výskytů termu t_i v dokumentu d_j a tuto hodnotu dělíme počtem všech termů v dokumentu. Inverzní frekvenci v dokumentech určíme dle vzorce:

$$idf = \log\left(\frac{|D|}{|j : t_i \in D_j|}\right), \tag{2.9}$$

kde |D| je počet všech dokumentů a ve jmenovateli je počet všech dokumentů obsahující term t_i.

Nyní musíme nalézt hranice mezi regiony. V případě Rocchio klasifikace se k určení hranic regionů používají tzv. „centroidy“. Centroid třídy *c* lze určit jako průměr vektorů nebo střed množiny jeho členů:

$$\vec{u}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d), \tag{2.10}$$

kde D_c je podmnožina všech dokumentů D náležící do klasifikační třídy c. Hranice mezi dvěma regiony je v případě Rocchio klasifikace určena jako množina bodů se shodnou vzdáleností od dvou centroidů. Grafickou reprezentací je přímka. Formálně lze pro výpočet volby regionu použít Eukleidovskou vzdálenost, nebo kosinovou podobnost:

$$classify(d) = argmax_c \cos(\vec{u}\,(c), \vec{v}(d)). \qquad (2.11)$$

Ze vzorce (2.11) je patrné, že klasifikace dokumentů je v případě vhodného výběru vlastností poměrně efektivní. Tuto klasifikaci však lze použít pouze v případě klasifikace právě do jedné klasifikační třídy.

2.4 Hierarchická klasifikace

Jednoúrovňová klasifikace patří mezi efektivní a použitelné techniky, avšak v případě výrazného růstu počtu kategorií je stále složitější procházení a vyhledávání kategorií. Jednou z cest řešící tento problém je organizace kategorií do stromové struktury. Jedním z příkladů může být např. hierarchie vyvinutá společností Yahoo! [14]. Mezi základní techniky hierarchické kategorizace je postupná aplikace ploché kategorizace v každé úrovni, dokud nedosáhneme listu, nebo dokud není možné klasifikovat do dalších podkategorií. Ve většině hierarchických klasifikačních metod [15] [4] [16] jsou kategorie reprezentovány v podobě stromové struktury. Dle [7] [5] [17] se nejčastěji pro hierarchickou klasifikaci využívají následující struktury:

1) **Virtuální strom kategorií** – kategorie jsou organizovány jako strom. Každá kategorie může náležet právě jedné rodičovské kategorii a dokumenty mohou být přiřazeny pouze kategoriím v listech [5].

2) **Strom kategorií** – v tomto případě se jedná o rozšíření virtuálního stromu kategorií, které umožňuje přiřazení dokumentů do kategorií v listech i uvnitř stromu [17].

3) **Virtuální orientovaný acyklický graf kategorií** – kategorie jsou v tomto případě organizovány jako orientovaný acyklický graf. Jedná se o podobné řešení jako v případě virtuálního stromu kategorií a dokumenty mohou být přiřazeny pouze kategoriím v listech. V tomto případě uzlům bez výstupních hran.

4) **Orientovaný acyklický graf kategorií** – toto je pravděpodobně jedna z nejčastěji používaných struktur v rámci katalogů webových stránek jako je Yahoo! [14], Open Directory Project [18], Firmy.cz [19], Centrum.cz [20] a jeho novější verze Najisto.cz [21]. Dokumenty, nebo v tomto případě webové stránky, mohou být zařazeny do uzlů s výstupními hranami i bez nich. Ve většině případů je možná kategorizace zároveň do několika kategorií, neboli výběr několika různých uzlů, které nemusí mít ani stejného předchůdce (rodiče).

V případě ploché kategorizace se velmi často k vyhodnocení kvality metody využívají míry jako je přesnost a úplnost. Vyhodnocení hierarchické klasifikace lze do jisté míry provádět

s využitím stejných evaluačních metod vyvinutých pro plochou klasifikaci. V současné době jsou nejvíce používané tyto základní přístupy k hierarchické klasifikaci [7]:

- **Big-bang** – v případě této metody je použit pouze jeden klasifikátor v průběhu klasifikačního procesu. Dokumentu je přiřazena jedna nebo více kategorií ze stromu kategorií. Tento evaluační přístup byl použit v případě rule-based klasifikátoru [22], obdobě Rocchio klasifikátoru [23], nebo metodě založené na dolování asociačních pravidel [17]. Výkonnostní míry byly v těchto případech postaveny pouze na jednoduchém měření spočívající v určení počtu správně klasifikovaných dokumentů, nebo procentuální chybovosti.

- **Top-down level-based** – v tomto případě je použit jeden, nebo více klasifikátorů pro každou úroveň klasifikačního stromu. Každý klasifikátor je realizován jako plochý na dané úrovni stromu. Klasifikace dokumentu probíhá ze shora dolů. Tato technika byla použita v případě vícenásobného použití Bayesovského klasifikátoru [6], algoritmu prezentovaného [4] a Support Vector klasifikátoru [5]. V těchto případech byly použity základní míry jako přesnost, úplnost a F-míra.

Hlavními problémy v případě **big-bang** přístupu jsou:

- Může využívat pouze informaci uloženou ve stromové struktuře z trénovací fáze.
- Rodičovská kategorie nemůže být diskriminační na úrovni potomků. Problémem je vhodné využití různých vlastností na odlišných úrovních stromu.
- Klasifikátor je třeba opětovně natrénovat v případě změny struktury kategorií.

Problémy v případě **top-down level-based** [7] přístupu:

- V případě chybné kategorizace na rodičovské úrovni může dojít k vyloučení všech kategorií na následující úrovni bez ohledu na to, že dosud tento klasifikátor nebyl použit.
- Metody založené na tomto přístupu vyžadují větší trénovací množinu z důvodu většího množství klasifikátorů, než v případě big-bang přístupu. Každý klasifikátor obvykle vyžaduje jiný trénovací soubor. V případě nedostatečné trénovací sady jednoho klasifikátoru může být negativně ovlivněn celý výsledek, což zvyšuje tlak na kvalitu natrénování každého klasifikátoru zvlášť.

Většina metod hierarchické klasifikace aplikuje **virtuální strom kategorií**, což zajistí, že pro potřeby evaluace lze využít standardní míry z ploché klasifikace. Jenom připomeňme, že v tomto případě jsou všechny dokumenty zařazeny pouze do kategorií v listech a vlastní hierarchii již není třeba pro potřeby evaluace uvažovat, neboť vnitřní uzly (kategorie) nemají přiřazeny žádné dokumenty. Postačí nám pouze standardní míry jako je přesnost, úplnost, F-míra apod.

V případě **stromu kategorií** je však možné přiřazovat i vnitřní uzly, což zvýší náročnost evaluace. Standardní míry pro plochou kategorizaci nemusí být v tomto případě hierarchické

klasifikace dostatečné. Pokud bychom tyto míry přesto použili, nemusí být výkonnost hierarchické klasifikace správně zachycena. Pro potřeby evaluace je třeba vzít v úvahu dvě relace:

- **rodič-potomek** – v případě výběru kategorie rodiče, následuje klasifikace na úrovni jeho potomků.
- **sourozenci** - pokud dvě kategorie sdílí významné množství dokumentů, lze je označit za sourozence bez ohledu na to, zda měli stejného přímého předka, či nikoliv.

Pokud je dokument zařazen do špatné kategorie, je třeba vzít v úvahu, zda mezi „správnou“ a „špatnou“ kategorií neexistuje jeden z těchto vztahů. Pokud ano, lze považovat tuto chybu za menší, než v případě klasifikace do zcela odlišné kategorie.

2.4.1 Vyhodnocení hierarchické klasifikace na základě podobnosti kategorií

Podobnost kategorií (Category Similarity) lze pro dvě kategorie C_i a C_k obecně vyjádřit funkcí $CS(C_i, C_k)$. Dle [7] je možné využít např. upravenou verzi kosinové vzdálenosti mezi vektory příznaků těchto kategorií. Vektor příznaků kategorie by měl být odvozen ze sumarizace vektorů příznaků všech trénovacích dokumentů spadajících do této kategorie. Na základě podobnosti kategorií *CS* lze definovat i průměrnou podobnost kategorií *ACS* vzorcem (2.13) viz [7]:

$$C_i = \{w_1 t_1, w_2 t_2, \dots, w_N t_N\}$$

$$C_k = \{v_1 t_1, v_2 t_2, \dots, v_N t_N\}$$

$$CS(C_i, C_k) = \frac{\sum_{n=1}^{N}(w_n \times v_n)}{\sqrt{\sum_{n=1}^{N} w_n^2 \times \sum_{n=1}^{N} v_n^2}}, \tag{2.12}$$

$$ACS = \frac{2 \times \sum_{i=1}^{m} \sum_{k=i+1}^{m} CS(C_i, C_k)}{m \times (m-1)}, \tag{2.13}$$

kde t_n je term s indexem n, m počet tříd, w_n a v_n jsou odpovídající váhy v kategoriích. Na základě podobnosti kategorií *CS* nyní můžeme měřit stupeň správnosti přiřazených kategorií dokumentu d_j. Z důvodu dalšího výkladu zavedeme následující množiny dokumentů ve vztahu ke kategorii C_i:

- TP_i (true positive) – množina dokumentů správně klasifikovaných do kategorie C_i.
- FP_i (false positive) – množina špatně klasifikovaných dokumentů.
- FP_i (false negative) – množina špatně odmítnutých dokumentů pro klasifikaci do C_i.
- TN_i (true negative) – množina správně odmítnutých dokumentů.

Pokud je dokument d_j správně klasifikován do kategorie C_i, pak $dj \in TP_i$ a hodnota přesnosti a úplnosti pro C_i je rovna 1. Pokud však d_j je špatně klasifikován do C_i, měli bychom zvážit,

zda je přiřazená kategorie podobná se správnou kategorií. Množinu všech vybraných kategorií pro dokument d_j označíme $D^{sel\text{-}j}$. Snažíme se tedy určit nakolik se *dj* může podílet (*contribute* = *Con*) v kategorii C_i, pokud počítáme přesnost a úplnost pro kategorii C_i. Dle [7] značíme *Con*(dj, C_i) a definujeme jako:

$$Con(d_j, C_i) = \frac{\sum_{C' \in D^{sel-j}} (CS(C', C_i) - ACS)}{1 - ACS}, \tag{2.14}$$

kde v čitateli počítáme celkovou sumu podobnosti všech kategorií $D^{sel\text{-}j}$ přiřazených k d_j ke správné kategorii C_i. Přínos dokumentu může být pozitivní, nebo negativní v závislosti na míře, nakolik jsou přiřazené kategorie shodné s průměrnou podobností kategorií *ACS*. Samozřejmě uvažujeme nejobecnější případ, kdy dokument může být zařazen do více kategorií. Abychom zabránili výrazně většímu vlivu jednoho dokumentu na úkor ostatních omezujeme přínos dokumentu d_j pro kategorii C_i pouze na rozpětí [-1, 1]. Proto definujeme omezení *RCon*(d_j, C_i) jako:

$$RCon(d_j, C_i) = \min(1, \max(-1, Con(d_j, C_i))). \tag{2.15}$$

Pro všechny dokumenty náležící FP_i, lze stanovit celkový přínos $FpCon_i$:

$$FpCon_i = \sum_{d_j \in FP_i} RCon(d_j, C_i). \tag{2.16}$$

Na základě těchto úprav můžeme definovat rozšířenou verzi pro přesnost Pr_i^{CS} a úplnost Re_i^{CS} pro kategorii C_i na základě podobnosti kategorií:

$$Pr_i^{CS} = \frac{\max(0, |TPi| + FpCon_i + FnCon_i)}{|TPi| + |FPi| + FnCon_i} \tag{2.17}$$

$$Re_i^{CS} = \frac{\max(0, |TPi| + FpCon_i + FnCon_i)}{|TPi| + |FNi| + FpCon_i} \tag{2.18}$$

Pro celkové vyhodnocení kvality klasifikátoru je pak možné na základě těchto vzorců stanovit BEP případně průměrnou 11-bodovou přesnost (Average 11-point precision).

2.4.2 Vyhodnocení hierarchické klasifikace na základě vzdálenosti kategorií

Nyní se zaměříme na míry založené na vzdálenosti kategorií [7] ve stromu kategorií. Vzdálenost mezi kategoriemi C_i a C_k označíme jako *Dis*(C_i, C_k) a definujeme jako počet linků (relací) mezi C_i a C_k. Intuitivně tak lze konstatovat, že čím kratší cesta mezi kategoriemi, tím jsou si obě kategorie blíže. Vzdálenost mezi kategoriemi byla poprvé zavedena v [24]. Nejdříve musíme definovat přijatelnou vzdálenost označenou jako *Dis*. Tato vzdálenost musí být větší než 0 a vyjadřuje vzdálenost, kdy lze dokument považovat za aspoň částečně správně zařazený. Formálně můžeme přínos dokumentu d_j kategorii C_i na základě vzdálenosti kategorií definovat jako [7]:

Pokud $d_j \in Fp_i$ pak:

$$Con(d_j, C_i) = \sum_{C' \in D^{sel-j}} (1.0 - \frac{Dis\,(C', C_i)}{Dis_\theta}). \quad (2.19)$$

Hodnota přínosu dokumentu je pak následně opět omezena pouze na rozsah [-1, 1] z důvodů vysvětlovaných výše. Na základě tohoto vzorce pak lze definovat přesnost a úplnost obdobně jako v případě podkapitoly „Vyhodnocení hierarchické klasifikace na základě podobnosti kategorií“.

3. Shlukování

V této kapitole si představíme problematiku shlukování, neboť v disertační práci bychom se chtěli věnovat vylepšení některých shlukovacích metod s využitím sémantických informací získaných z linked data. Shlukování je metoda umožňující rozdělení dokumentů do množin, což nám umožňuje rychlejší analýzu, než kdybychom zkoumali každý dokument zvlášť. Vycházíme z předpokladu, že dokumenty budou mít podobný obsah a zaměření. Rozlišujeme následující typy:

- Tvrdé shlukování (hard clustering) – 1 dokument je třeba zařadit právě do 1 shluku,
- Měkké shlukování (soft clustering) – 1 dokument lze zařadit do více shluků.

Dle úrovně zařazení můžeme shlukování dále dělit na:

- **Jednoúrovňové shlukování** – všechny shluky jsou na stejné úrovni, součástí vstupu je většinou požadované množství shluků.

- **Hierarchické shlukování** – shluky se dále spojují a vytváří hierarchickou strukturu zcela automaticky, nebo dle vstupního požadavku na počet shluků, do kterých je třeba dokumenty rozdělit. Ve většině případů však není nutné zadávat požadované množství shluků, neboť algoritmy jsou schopné deterministicky zvolit optimální množství.

Mezi hlavní výhody shlukování patří schopnost nesupervizovaného učení. Jedním z hlavních problémů je pojmenování shluků, které je významné zvláště v případě hierarchického shlukování. Metoda shlukování obecně sdružuje dokumenty s velkým počtem shodných termů.

Shlukovací hypotéza: dokumenty stejného shluku se chovají podobně a mají podobné vlastnosti.

3.1 Jednoúrovňové shlukování

Mezi nejdůležitější metody jednoúrovňového shlukování patří K-průměrů. Jejím cílem je minimalizovat průměr čtverců Eucleidovské vzdálenosti dokumentů od středu shluku. Střed shluku je definován jako centroid $\vec{\mu}$ v shluku *w*:

$$\vec{\mu}(w) = \frac{1}{|w|}\sum_{\vec{x}\in w}\vec{x}. \qquad (3.1)$$

Dokumenty jsou reprezentovány jako normalizované vektory v reálném prostoru. Obdobným způsobem byly použity centroidy v případě Rocchio klasifikace. Hlavním rozdílem oproti Rocchio klasifikaci je fakt, že nemáme k dispozici žádnou trénovací množinu dokumentů, u kterých bychom věděli, do kterého shluku náleží.

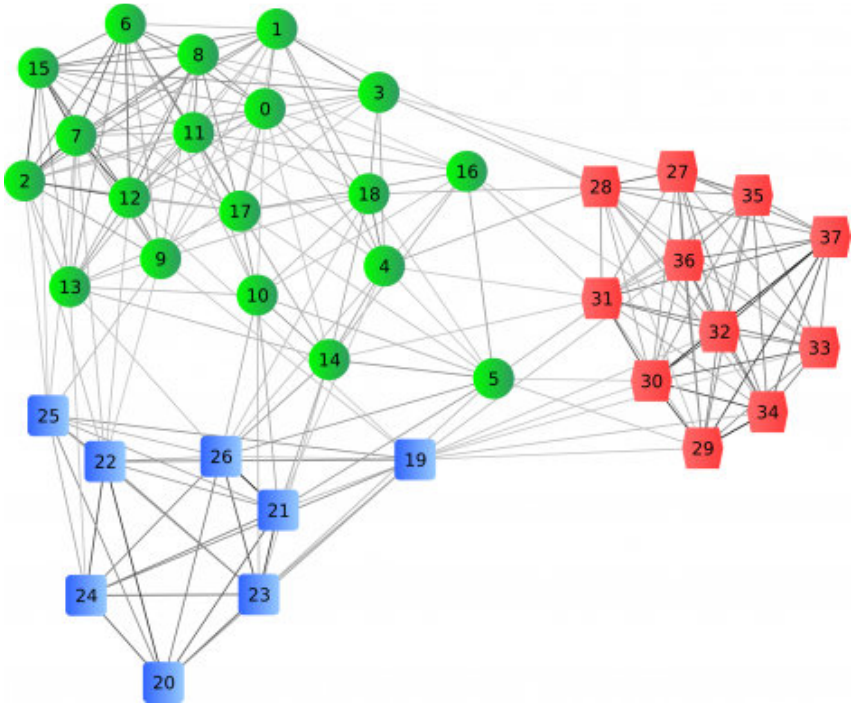
Míra určující, jak dobře centroid reprezentuje dokumenty obsažené v shluku se nazývá reziduální suma čtverců (residual sum of squares = RSS). Jedná se o sumu čtverců vzdálenosti každého vektoru od centroidu. Suma čtverců vzdálenosti vektoru *k* od centroidu se určí s využitím vzorce:

$$RSS_k = \sum_{\vec{x} \in w_k} |\vec{x} - \vec{\mu}(w_k)|^2. \quad (3.2)$$

Výsledné RSS se vypočte s využitím vzorce:

$$RSS = \sum_{k=1}^{K} RSS_k. \quad (3.3)$$

Cílem k-průměrů je minimalizace RSS funkce. Minimalizace RSS je ekvivalentní k minimalizaci průměru čtverců vzdálenosti, neboli míry, jak dobře centroidy reprezentují jejich dokumenty. Celý popis algoritmu zde nebudeme z důvodu jeho rozsahu a obecné známosti uvádět. Schéma jednoúrovňového shlukování je zobrazeno na obrázku 3.1.



Obrázek 3.1 – schéma jednoúrovňového shlukování [25]

3.1.1 Shlukování založené na cestě

Nyní si představíme základní principy shlukování založeného na cestě. Shlukování založené na cestě (Path-based clustering) [26] je technika shlukování, která zařadí dva dokumenty do jednoho shluku, pokud jsou spojeny cestou s vysokou podobností objektů na cestě. Hlavní výhodou této metody je její robustnost při extrakci protáhlých struktur a řešení problematiky „odpadlíků“ (outliners). Předpokládejme, že dokumenty jsou popsány na základě rozdílů mezi dvojicemi dokumentů (obdoba complete-link). Matici rozdílů označíme *D* a množinu objektů $O = \{o_1, o_2, .. ,o_n\}$. Mapovací funkce $c: O \rightarrow \{1, .. ,k\}$ mapuje každý objekt na jeden

z *k* popisků (štítků). Ohodnocovací funkce h: $C \rightarrow R_+$ ohodnocuje každou mapovací funkci *c* kladnou reálnou hodnotou tak, aby minimum *h(c)* bylo řešením shlukovacího problému. *C* je množina všech možných mapovacích funkcí.

Jádrem každé shlukovací metody je rozhodnutí, kdy dva objekty patří do stejného shluku. V případě párového (pairwise) shlukování jsou dva podobné objekty $o_{i,}$ o_j zařazeny do stejného shluku D_{ij}, pokud *c(i) = c(j)*. Tento princip může být zobecněn s využitím předpokladu tranzitivního chování. Definujme množinu všech cest $P_{ij}(c)$ z objektu o_i do objektu o_j, kde všechny ostatní objekty na cestě náleží do stejného shluku jako objekt o_i, nebo objekt o_j. Rozdílnost způsobená podcestou $p \in P_{ij}(c)$ je definována jako maximální rozdílnost pro tuto cestu. Nejmenší vzdálenost mezi dvěma objekty je určena jako minimální délka cesty mezi objekty o_i a o_j :

$$D_{ij}^{min} = min_{p \in P_{ij}(c)} \{ max_{1 \leq h \leq |p|-1} D_{p[h]p[h+1]} \}. \quad (3.4)$$

S využitím definice této minimální rozdílnosti můžeme zavést ohodnocovací funkci pro Path Based clustering. Ohodnocení pro každý shluk bude vyjádřeno jako aritmetický průměr minimálních délek cest mezi objekty. Tuto funkci definujeme jako [26]:

$$h^{pbc}(c; D) = \sum_{v \in \{1,..,k\}} \frac{1}{|O_v(c)|} \sum_{o_i \in O_v} \sum_{o_j \in O_v} D_{ij}^{min}(c; D). \quad (3.5)$$

Praktické využití Path Based shlukování se většinou orientuje na zpracování obrazu, avšak základní myšlenka je použitelná i pro zpracování textových dat.

3.2 Hierarchické shlukování

Výstupem hierarchického shlukování je hierarchická struktura shluků. Tato struktura je mnohem více informativní, než v případě nestrukturované množiny shluků z plochého shlukování. Základními rozdíly oproti plochému shlukování jsou:

- hierarchické shlukování nevyžaduje zadání počtu shluků,
- většina algoritmů se chová deterministicky,
- efektivnost hierarchického shlukování je nižší, než v případě plochého shlukování.

Metody hierarchického shlukování je možné dělit dle způsobu tvorby hierarchické struktury na:

- **Aglomerativní** – jedná se např. o algoritmy single-link, complete-link, group-average a centroid-similarity. V tomto případě se jedná o spojování shluků ze zdola nahoru od listů směrem ke kořenu. Listy jsou tvořeny individuálními dokumenty a kořen obsahuje množinu všech dokumentů.

- **Divisivní metody** – postupujeme ze shora dolů, neboli od kořene k listům.

Nyní se zaměříme na aglomerativní metody shlukování, neboť se jedná o nejčastěji používané metody:

- **Single-link** – počítáme podobnost dvou nejvíce podobných členů. Spojovací kritérium je lokální. Nevýhodou tohoto algoritmu je, že produkuje roztroušené shluky, neboť spojovací kritérium je striktně lokální.

- **Complete-link** – počítáme podobnost dvou nejméně podobných členů. Spojovací kritérium je nelokální. Nevýhodou tohoto algoritmu je, že se příliš zaměřuje na vzdálené dokumenty, které velmi špatně zapadají do globální struktury shluku.

- **Group-average** – počítáme podobnost shluků na základě všech podobností mezi dokumenty. Touto metodou se vyhneme problémům nastíněným v případě single-link a complete-link algoritmů.

- **Centroid similarity** – v tomto případě počítáme podobnost shluků na základě podobnosti jejich centroidů.

3.3 Použití shlukování

V mnohých aplikacích využívajících ploché, nebo hierarchické shlukování je třeba interakce s uživatelem většinou s využitím grafického uživatelského prostředí. V tomto případě je třeba výstižné pojmenování shluků. Existují dvě základní techniky značení shluků:

- **Differential cluster labeling** – označení shluku je vytvořeno na základě porovnání důležitých termů jednoho shluku se všemi ostatními shluky. Porovnání se provádí s využitím vhodné volby vlastností – např. metodami X^2 test, nebo s využitím vzájemné informace (mutual information).

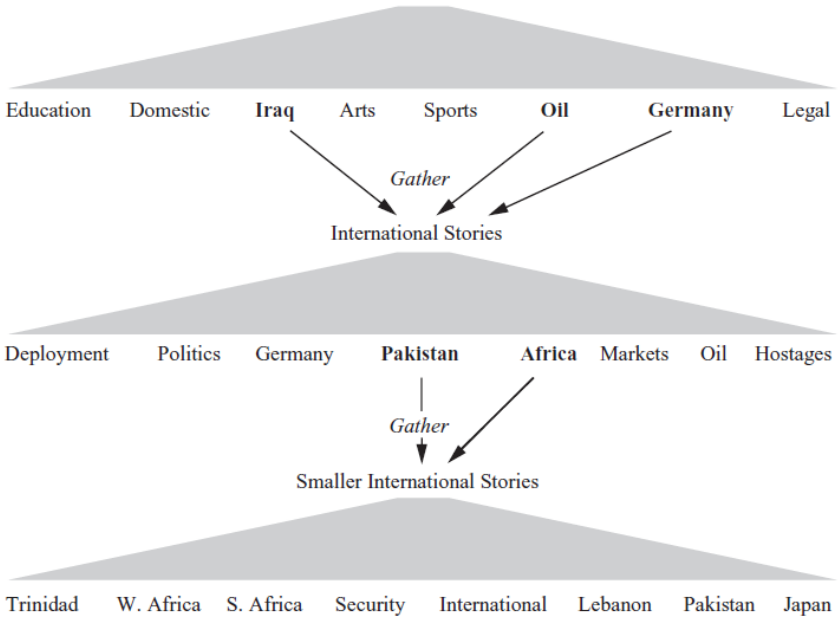
- **Cluster internal labeling** – v tomto případě se označení shluku volí na základě obsahu pouze daného shluku bez ohledu na ostatní. Jednou z metod je např. označení shluku s využitím dokumentu, který je nejblíže centroidu. Titulek dokumentu použitý k označení shluku je tak lépe čitelný, než seznam termů. Další možností je využití seznamu termů, které v rámci shluku hrají nejvýznamnější roli. Tyto metody pojmenování shluků jsou velmi efektivní, avšak nemusí poskytovat dostatečné odlišení od ostatních shluků.

Praktické aplikace využívající shlukování je většinou možné zařadit do jedné ze dvou základních kategorií:

- **search result clustering** - shlukování výsledků vyhledávání. Uživatel nemusí individuálně procházet všechny výsledky vyhledávání, ale může je procházet po shlucích. Jestliže jeden nebo více shluků označí jako nevhodný, další dokumenty z tohoto shluku mu již v daném případě nebudou nabízeny. V průběhu vyhledávání

nejdříve najde požadovaný shluk a až následně prochází individuální dokumenty. V tomto případě se jedná o využití jednoúrovňového shlukování.

- **scatter-gather** – tato technika hierarchického shlukování umožňuje uživateli zvolit množinu shluků, jejichž dokumenty jsou vybrány a následně opětovně shlukovány. Témata obsažená ve shlucích jsou stále více konkrétní a přibližují se k hledanému tématu. Takto uživatel postupuje až k požadovanému výsledku. Tato technika je zobrazena na obrázku 3.2.



Obrázek 3.2 – technika scatter-gather [8]

4. Linked Data

V této kapitole si stručně vysvětlíme, co to jsou linked data a jejich základní principy a možnosti využití. Představíme si i nejvýznamnější nástroje umožňující jejich prohlížení. Nejdříve bychom však měli začít pojmem Sémantický Web. Tim-Berners Lee definuje Sémantický Web [27] jako přístup umožňující vyjádření informací ve strojově čitelné podobě. Jeho hlavní myšlenka je provázání dat s využitím odkazů, což umožní lidem i strojům procházet související data (linked data). T. B. Lee formálně definoval 4 základní pravidla linked data [28] [29]:

1. Používat URI jako identifikátory věcí.
2. Používat http URI, aby si lidé mohli tyto věci prohlédnout.
3. Pokud se někdo podívá na URI, je třeba poskytnout užitečné informace s využitím standardů (RDF*, SPARQL [30]).
4. Používat odkazy na další URI, což umožní vyhledávat související věci.

První a druhé pravidlo vyžaduje používání URI jako identifikátorů. Jinak řečeno se jedná o jednoznačná jména, nikoliv adresy, jak by mohlo být mylně interpretováno. Třetí pravidlo vyžaduje popis objektu s využitím standardů v případě přístupu na toto URI. Člověk i stroj by měl získat základní informace o objektu, jeho formálním popisu a navíc i odkazy na další objekty, což je základní myšlenkou linked data.

Pro publikaci informací s využitím linked data můžeme použít jednu z následujících metod [31]:

- Pokud na Webu nemáme žádné informace, je vhodnější je publikovat v čisté sémantické podobě např. s využitím standardu RDF [32].
- Pokud na Webu máme existující informace, je vhodnější jim pouze přidat sémantický význam. K tomuto účelu je možné vkládat tato metadata přímo do webové stránky s využitím RDFa [28], nebo mikroformátů [33].

Hlavní rozdíl mezi RDFa a mikroformáty je ve způsobu použití XHTML atributů pro uložení metadat. Mikroformáty využívají zejména atribut *class*, zatímco RDFa používá více popisných metod pro uložení sémantických informací. Mikroformáty využívají mnoho různých slovníkově specifických syntaxí, oproti tomu RDFa je založeno na obecné metodě vkládání sémantického obsahu, které je slovníkově nezávislé. RDFa používá např. XHTML atributy: *about*, *resource*, *instanceof*, *property*, *content*. Zajímavostí je, že atributy jako *rel* a *href* mohou být použity pro všechny html elementy, nikoliv pouze pro odkazy.

Nyní si ukážeme příklad demonstrující použití RDFa ke strojově čitelnému označení události. Tento příklad je vytvořen dle podpory Yahoo! [14] pro RDFa:

```
<div typeof="vcal:Vevent"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

```
        xmlns:vcal="http://www.w3.org/2002/12/cal/icaltzd#"
        xmlns:vcard="http://www.w3.org/2006/vcard/ns#"
        xmlns:review="http://purl.org/stuff/rev#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
        xmlns:commerce="http://search.yahoo.com/searchmonkey/commerce/">

    <span property="rdfs:label vcal:summary">ICSS 2010</span>
    <span property="vcal:description rdfs:comment">The 7th International
        Conference…</span>
    <span property="v:summary">ICSS 2010</span>
    <span property="v:description">The 7th International Conference </span>

    <span property="vcal:dtstart" datatype="xsd:dateTime"
        content="2010-08-17">August 17</span> —
    <span property="vcal:dtend" datatype="xsd:dateTime"
        content="2010-08-20">August 20</span>
```

Z příkladu je zřejmé, že použití RDFa místo mikroformátů je mnohem vhodnější zejména z důvodu popisnějšího sémantického značení. V případě mikroformátů se XHTML atribut *class* k sémantickému značení spíše zneužíval, než aby sloužil k jeho původnímu účelu formátovat s využitím CSS vzhled webové stránky.

Linked data je možné analyzovat s využitím statistických metod [34], nebo s využitím metod navržených pro analýzu citačních sítí [35].

4.1 Prohlížení Linked Data

Prohlížení linked data je většinou možné s využitím běžného webového prohlížeče. DBPedia [36] [37] např. detekuje přístup s využitím webového prohlížeče a výstup pro něj upraví. V případě prohlížení čistých linked data bez standardu xhtml je však vhodné použití specializovaného prohlížeče sémantického webu. Tyto nástroje můžeme rozdělit do následujících kategorií:

- **Obecné sémantické prohlížeče** - jedním z obecně použitelných sémantických prohlížečů je Tabulator [38], na kterém se podílel T. B. Lee. Tento prohlížeč si klade za cíl být obecným nástrojem pro prohlížení sémantického webu. Na druhou stranu však podporuje i doménově specifické aplikace. Jedná se např. o adresář, manipulaci s penězi, kalendář i speciální funkcionalitu jako je manipulace s formuláři. Tento prohlížeč lze používat jako webovou aplikaci, nebo si celou jeho funkcionalitu můžeme nainstalovat v podobě rozšíření např. do Firefoxu. Další sémantické prohlížeče je třeba stáhnout a nainstalovat na pc. Jedná se např. o nástroje: Magpie [39], Haystack [40], Piggybank [41] a Longwell [42].

- **Obecné prohlížeče s omezenou funkcionalitou** – z těchto nástrojů můžeme jmenovat např. Palm-DAML [43], RDF Author [44] a IsaViz [45]. Hlavní nevýhoda spočívá v omezené schopnosti zobrazit data v očekávané podobě, jako jsou např. tabulky a grafy.

- **Specializované prohlížeče** – do této kategorie patří nástroje CS-Aktive [46] a mSpace [47]. CS-Aktive se zaměřuje na informace o lidech, jejich projektech, výzkumu a umí vyhledávat např. dle geografické polohy. mSpace je orientován na hudbu a informace o ní. Oba tyto prohlížeče jsou poměrně komfortní, avšak orientované pouze na vybranou oblast zájmu. Tím je jejich použitelnost velmi omezená.

5. Štítkování

Štítkování je technika přiřazení popisku (label, tag, štítek) např. k textovému dokumentu. Tento dokument může být uložen lokálně, nebo kdekoliv na Internetu. V případě vzdáleného umístění je třeba tento dokument (zdroj) identifikovat. K tomuto účelu se nejlépe hodí využít technik linked data [28] a označit ho s využitím http URI. Štítkování se velmi rychle rozšiřuje zvláště v prostředí Internetu a umožňuje uživatelům označovat dokumenty dle jejich vlastního uvážení. Mezi největší výhody a zároveň i nevýhody patří možnost nechat uživatele štítkovat dle jeho osobního názoru. Výhodou je, že uživatel nemusí nad tímto úkolem přemýšlet a může volně vyjádřit své názory a pocity ohledně dokumentu.

Uživatel může štítkovat dokumenty pro sebe, nebo i pro ostatní. V každém případě však dojde k přidání metadat k dokumentu, což nám umožní lepší vyhledávání, prohlížení a organizování dat. I když v nejhorším možném případě dokument označí prostým štítkem „To se mi líbí“, další uživatele může tento obsah také zaujmout. Na tomto jednoduchém principu funguje např. populární sociální síť Facebook [48].

5.1 Hlavní problémy se štítkováním

Ve většině systémů umožňujícím štítkování si uživatel může vybrat štítek ze sdíleného slovníku, nebo přidat vlastní. Z možnosti definice štítku dle vlastních potřeb však vznikají problémy. Dle [49] a našeho článku [50] patří mezi hlavní problémy zejména:

- **Synonyma** – slova se stejným významem. Tato různá slova se stejným význam představují jeden z hlavních důvodů nekonzistentnosti ve štítkovacích systémech. V případě vyhledávání dochází k nalezení neúplné množiny vyhovujících záznamů. Část záznamů může být z důvodu špatné volby synonyma zcela nedostupná. Autor článku se tak musí spoléhat zejména na svoji vlastní expertízu a schopnost vybrat správnou variantu štítku.
- **Homonyma** – slova, která stejně znějí, ale mají různý význam.
- **Polysemická slova** – polysemické slovo má několik souvisejících významů. Odstranění polysemického slova můžeme provést s využitím dalších štítků, nebo doplňujícího vysvětlení [50].
- **Různá úroveň popisu** – štítky mohou být příliš detailní, nebo obecné. Problémem je najít optimální a stále stejnou úroveň popisu. Může se jednat např. o popis objektu na obrázku. Např. auto vs. jaguár vs. jaguár model xk.
- **Zkratky** – obecně známé zkratky mohou být používány bez komentářů, avšak ostatní je třeba důsledně specifikovat. Např. zkratky RDF a OWL. Obě jsou velmi dobře známé v této podobě a mnoho odborníků je pravidelně používá, avšak jejich celé znění se používá spíše minimálně.

5.2 Relevantní systémy využívající štítkování

Současné relevantní systémy můžeme rozdělit do následujících kategorií:

- **Systémy pro automatické štítkování** – tyto metody jsou většinou založené na trénovacích korpusech, nebo statistice [50]. Oba typy metod většinou využívají vzorce na principu TFIDF. Statisticky orientované metody ke zjištění nejdůležitějších termů a korpusově orientované metody k mapování testovaného dokumentu na objekt z trénovacího korpusu.

- **Systémy pro automatické shlukování na základě štítků** – shlukování poskytuje rozdělení jedné velké množiny štítků do menších skupin s podobnými vlastnostmi s ohledem na stejnou úroveň popisu. Shlukovací algoritmus může být postaven na základě sdílených štítků [51]. Další přístup [52] využívá štítky pouze k vytipování dokumentů, které mohou být podobné. Tyto dokumenty jsou následně porovnávány s využitím TFIDF a dle této míry podobnosti dále shlukovány. Tento přístup je efektivnější, než porovnávání náhodně zvolených dokumentů.

- **Systémy pro návrh štítků** – tyto systémy jsou vytvořeny za účelem návrhu štítků uživateli. Např. TagAssist [53] je založen na anotovaném korpusu a testované dokumenty s tímto korpusem pouze porovnává s využitím TFIDF. Tato metoda umožňuje učení, avšak její hlavní nevýhodou je potřeba korpusu obsahující podobné dokumenty. Další systémy fungují jako webové služby umožňující uživatelům sdílení štítků i oštítkovaných zdrojů. Jedná se např. o služby Delicious.com [54] a Technorati [55]. Další možností je např. implementace jako v případě YouTube. Uživatelé mohou sdílet štítky a označovat videa, avšak pouze v rámci serveru YouTube.

- **Systémy pro manuální štítkování s využitím technik Sémantického webu** – existuje několik projektů, které se specializují na sdílení manuálně zadaných štítků. Projekt Int.ere.st [56] využívá pro manuální štítkování ontologii SCOT [57] (Social Semantic Cloud of Tags). Tento projekt poskytuje otevřenou štítkovací platformu s veřejným API, což umožňuje její využití v rámci dalších aplikací. Další projekt je postaven na frameworku MOAT [58] (Meaning Of A Tag), který umožňuje autorům používat sdílenou množinu štítků a pomáhá jim zvolit optimální verzi štítku dle jeho významu. Význam štítku je vyjádřen jako trojice v ontologii:
  - **Subjekt:** zvolený štítek,
  - **Predikát:** *moat:meaningURI*,
  - **Objekt:** zdroj v rámci linked data identifikovaný s využitím http URI.

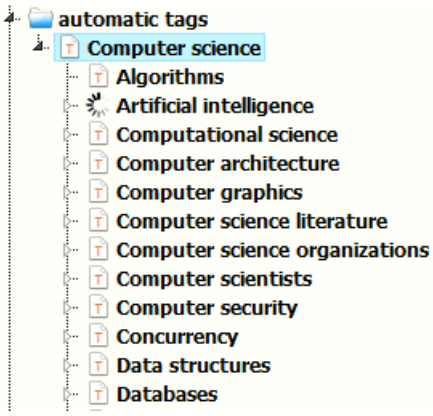
5.3 Naše technika štítkování s využitím Linked Data

V rámci článku [50] jsme publikovali techniku štítkování s využitím linked data. Tato technika je založena spíše na principech a teorii, než konkrétní implementaci, nebo přesné definici algoritmu. Štítkování s využitím linked data by mohlo probíhat s využitím následujících principů:

- Metody pro štítkování s využitím linked data by měly být implementačně jednoduché a pokud možno bez potřeby specializované webové služby. Podobných služeb bylo již publikováno několik [53], avšak velice rychle a často zanikají, což v potenciálních uživatelích vzbuzuje nedůvěru. Obecnější webové aplikace jako Delicious [54] nebo Technorati [55] jsou sice trvalé, avšak API je pro napovídání štítků relativně nepoužitelné. Už jenom z důvodu absence dodatečné sémantické informace. Tyto služby Vám pomohou napovědět pojem, avšak jeho význam a související odkazy již nezískáte.

- Každý štítek bude identifikován s využitím URI a bude obsahovat odkazy na zdroje popisující jeho význam i na související objekty. Z tohoto důvodu je nejvhodnější využít dostatečně rozsáhlé portály zabývající se publikací dat ve strojově čitelné podobě. Zvolili jsme portály: DBPedia [37], Geonames [59] a Freebase [60]. Všechny tři portály poskytují přístup k datům s využitím API a zároveň data poskytují včetně sémantické informace. Z důvodu nejobecnějšího zaměření se v rámci dalšího popisu budeme věnovat zejména DBPedii, která zpřístupňuje data např. i s využitím SPARQL [30]. Jedna webová stránka tak bude představovat jeden štítek.

- DBPedie mimo jiné publikuje informace i ve formátu RDF. V případě, kdy dokumentu přiřadíme štítek, můžeme se podívat i na související témata a tento štítek tak dále upřesnit. DBPedia využívá ontologii SCOT [57], což nám umožňuje vytvořit mezi štítky hierarchické vazby. Prakticky nám tak vzniká univerzální strom štítků viz. obr. 5.1, který je možné využívat např. k odvozování dalších znalostí.



Obrázek 5.1 – strom štítků

V rámci dalšího výzkumu bychom se chtěli mimo jiné zaměřit i na vytvoření metod umožňující štítkování s využitím linked data. Principiálně by mělo probíhat takto:

1. V článku [50] jsme k vytipování klíčových slov používali pouze jednoduchou detekci pojmenovaných entit a vybraná slova jsme vyhledávali v rámci linked data. Později jsme však publikovali techniku [61] automatické extrakce klíčových slov a klíčových frází pouze s využitím statistických metod a NLP vzorů. Mapování těchto klíčových slov na linked data by bylo mnohem efektivnější.

2. Odvozování dle [50] probíhalo na principu vyhledávání štítků na úrovni sourozence, nebo rodiče. Ke štítkům by však mohlo být přistupováno jako k orientovanému grafu, místo stromu štítků. Tento přístup by umožňoval využití dalších metod pro prohledávání a analýzu grafů.

3. Štítky je možné dále využívat ke klasifikaci a shlukování dokumentů.

5.4 Shlukování štítků

Begelman et. al. [62] popisuje techniku shlukování štítků. Tato technika nám umožňuje vytvořit hierarchickou strukturu, která zefektivní vyhledávání a třídění štítků. Následně je pak možné tyto štítky využít pro lepší dosažitelnost jednotlivých dokumentů. Nyní si stručně popíšeme tuto techniku:

1. **Data** - data jsou získána z RSS kanálu služby Delicious [54], který obsahuje poslední štítkované stránky. RSS kanál obsahuje u každého odkazu množinu štítků v podobě jejich URI. Tato URI tvoří u každého z odkazů neorientovaný graf provázaný hranami technikou každý s každým.

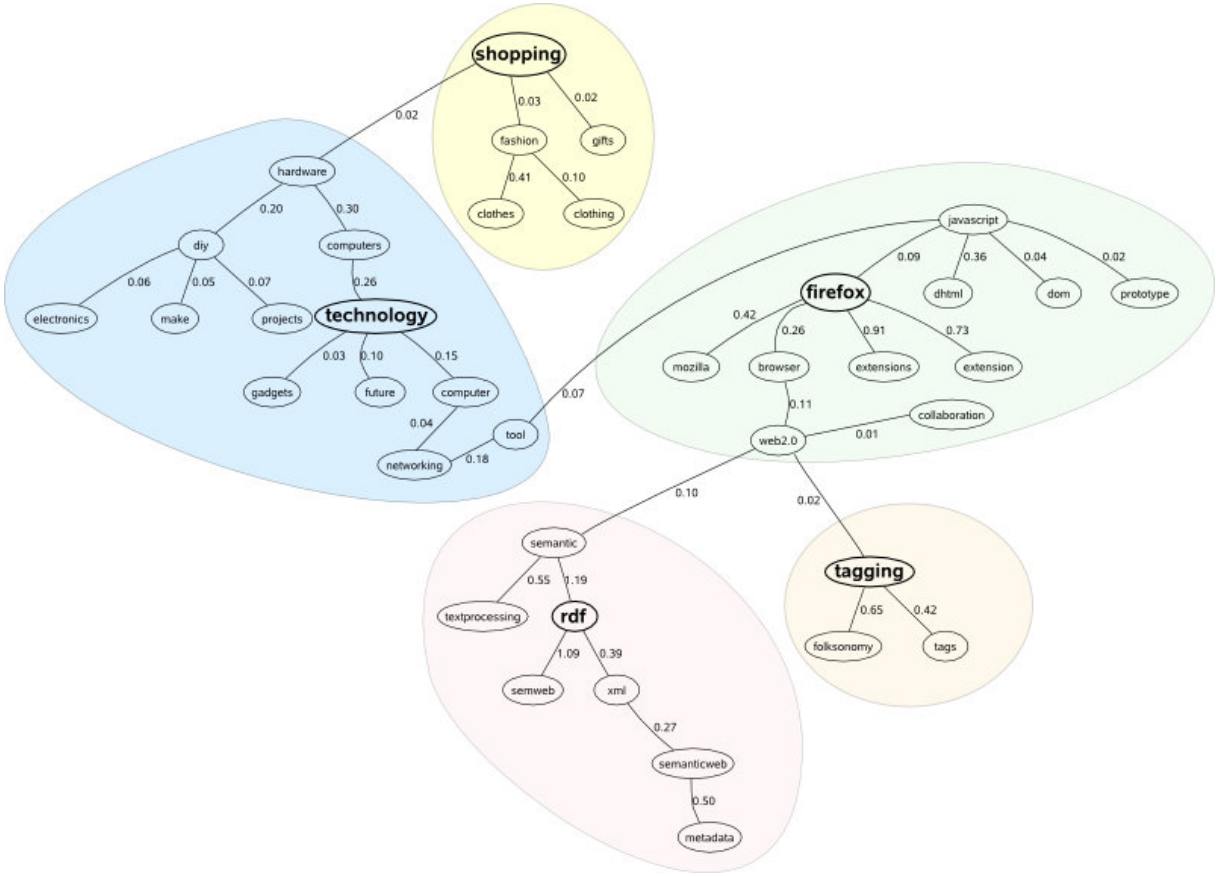
2. **Volba podobnostní míry** – grafy získané dle předchozího kroku jsou dále shlukovány dle podobnostní míry. Použitá podobnostní míra však nesmí být příliš ovlivněna populárností daného štítku. Např. štítek „Web 2.0“ se dle služby Delicious vyskytuje téměř s čímkoliv [62]. Z tohoto důvodu základní míra založená na společném výskytu dvou štítků není příliš efektivní. V [62] byla nakonec použita míra podobná spektrálnímu shlukování.

3. **Shlukování** – shlukovací algoritmus je založen na spektrální bisekci [63]. Nejdříve vytvoříme Laplaceovu matici L_G grafu G. Tato matice je symetrická a definovaná:
   - $L_G(i, i)$ je rovno stupni uzlu v_i (počet vstupních a výstupních hran)
   - $L_G(i, j) = -1$ pokud existuje hrana mezi uzly v_i a v_j.
   - $L_G(i, j) = 0$ v ostatních případech

   Následně počítáme vlastní vektor v_2 matice L_G odpovídající druhému největšímu pozitivnímu vlastnímu číslu matice L_G. Uzly grafu jsou rozděleny na dvě části na

základě znaménka odpovídající komponenty v_2. Takto je pokračováno, dokud není dosaženo optimálního množství shluků.

Výsledek shlukování štítků je zobrazen na následujícím obrázku:



Obrázek 5.2 – shlukování štítků [62]

6. Závěr a budoucí práce

V této práci jsme si představili základní techniky pro zpracování a analýzu dokumentů. Vysvětlili jsme si principy štítkování a důvody jeho expanze v prostředí Internetu. Linked data se jeví jako zajímavá technologie umožňující strojové zpracování dat. Tento nový přístup ke strojově čitelným datům nám umožní jejich další využití pro analýzu zdrojů bez sémantického významu. Linked data se přes velký potenciál v současné době k těmto účelům příliš nevyužívá, což je naší hlavní motivací pro budoucí výzkum.

Cílem disertační práce bude vytvoření metod využívajících zejména linked data k odstranění některých významných problémů jako např. pojmenování shluků a stavbu klasifikačního stromu. Tyto metody můžeme rozdělit do dvou skupin:

- metody pro zpracování jednoho dokumentu,
- metody pro manipulaci s více dokumenty.

6.1 Metody pro zpracování jednoho dokumentu

Dokument se většinou analyzuje v rámci kolekce podobně zaměřených dokumentů, avšak bez odpovídající trénovací množiny. Typickým příkladem může být analýza předem neznámého korpusu. V tomto případě není možné využít natrénovaných pravidel a je třeba využít jiný přístup. Cílem disertační práce bude vytvoření metod řešících následující úlohy:

a) **Automatická extrakce klíčových slov** – v případě čistě statistického přístupu se nejčastěji používá obdoba TFIDF skóre. S využitím linked data je možné vybraná slova testovat a určit, zda se jedná o důležitý pojem či nikoliv. Pokud ano, s využitím linked data je možné rovnou testovat související pojmy a velmi rychle určit tématické zaměření dokumentu. Výhodou je naprosto automatická funkčnost bez nutnosti trénování nástroje na podobné datové kolekci.

b) **Detekce pojmenovaných entit** – většinou probíhá s využitím vzorů, nebo slovníků. V případě využití linked data je možné potenciální pojmenované entity testovat a zároveň zjistit i typ nalezené entity. V případě nalezení entity „Plzeň“ je možné určit, že se jedná město a s využitím linked data přednostně testovat, zda se článek týká pouze tohoto města, regionu, nebo celé České republiky.

6.2 Metody pro manipulaci s více dokumenty (korpusy)

Tyto metody jsou zaměřené na zpracování množiny dokumentů a jejich analýzu. V tomto případě nás nezajímají jednotlivé dokumenty, ale zajímá nás korpus jako celek. Může se jednat např. o následující problematiku:

a) **Hierarchická klasifikace** – v tomto případě mezi zásadní problémy patří nutnost vybudovat hierarchickou strukturu kategorií a natrénovat klasifikátor. Tato práce může být velmi časově náročná a s využitím linked data je možné ji částečně automatizovat. Např. je možné vybrat entitu z linked data a stanovit pravidlo pro automatickou klasifikaci do podkategorií této entity. Linked data lze také využít pro napovídání kategorií, což usnadní manuální stavbu klasifikačního stromu. Některé zdroje mohou obsahovat vícejazyčné názvy, které lze využít k automatickému překladu klasifikačního stromu do jiného jazyka.

b) **Hierarchické shlukování** – v tomto případě je možné přímo využít strukturu zdrojů z linked data, které již hierarchickou strukturu většinou obsahují. Druhou možností může být pokus mapovat automatické shluky na linked data a získat tak vhodná pojmenování shluků. ´Takto je možné zcela vyřešit problematiku pojmenování shluků. Zcela novou možností je shlukovat dokumenty dle typu. Může se jednat např. o lidi, města a další objekty. Pokud jsou tyto vlastnosti s využitím nějaké existující ontologie definovány, je možné dokumenty dle těchto pravidel shlukovat a poskytnout uživateli zcela nový pohled na dokumenty. Linked data lze využít i pro vyvažování velikosti shluků. Pokud je jeden shluk příliš velký, můžeme z linked data získat jeho potomky a dokumenty k těmto potomkům přidělit. Vlastní mapování dokumentu na linked data je možné s využitím běžných statistických metod jako např. TFIDF, nebo s využitím Naivního Bayesovského klasifikátoru, kdy je možné porovnat text článku s popisem zdroje v linked data.

c) **Štítkování dokumentů** – dokumenty je možné štítkovat s využitím linked data [50]. Klíčová slova je možné extrahovat s využitím statistických metod [61] a následně je mapovat na linked data. Díky tomuto přístupu je možné štítky jednoznačně identifikovat s využitím URI a odstranit téměř všechny existující problémy. Synonyma je možné odstranit díky existující vazbě „*sameAs*“ mezi zdroji, polysemická slova budou mít vlastní URI s dostatečnou definicí každého objektu a různou úroveň popisu můžeme sjednotit s využitím hierarchických vazeb potomek-rodič. V kapitole o hierarchické klasifikaci jsme si představili techniky evaluace založené na podobnosti kategorií [7] a vzdálenosti kategorií [7] ve stromu. Tyto techniky můžeme relativně snadno aplikovat i na linked data.

Citovaná literatura

[1] *Český statistický úřad.* [Online] 10. dubna 2011. [Citace: 10. dubna 2011.] http://www.czso.cz.

[2] **Yang, Y.** An evaluation of statistical approaches to text categorization. *Information Retrieval Journal 1.* Pittsburgh : Kluwer Academic Publishers, 1999, stránky 69–90.

[3] *Inductive Learning Algorithms and Representations for Text Categorization.* **Dumais, S., a další.** 1998. Proc. of the 7th Int. Conf. on Information and Knowledge Management. stránky 148–155.

[4] *The effect of using hierarchical classifiers in text categorization.* **D'Alessio, S., a další.** Paris : RIAO, 2000. Proc. of the 6th Int. Conf. Recherche d'Information Assistee par Ordinateur. stránky 302–313.

[5] *Hierarchical classification of Web content.* **Dumais, S. a and Chen, H.** Athens : ACM, 2000. Proc. of the 23rd ACM Int. Conf. on Research. stránky 256–263. ISBN: 1-58113-226-3.

[6] *Hierarchically classifying documents using very few words.* **Koller, D. a and Sahami, M.** San Francisco : Morgan Kaufmann Publishers Inc., 1997. Proc. of the 14th Int. Conf. on Machine Learning. ISBN: 1-55860-486-3.

[7] *Hierarchical Text Classification and Evaluation.* **Sun, A. a Lim, E.** Washington : IEEE Computer Society, 2001. ICDM 2001. stránky 521-528. ISBN:0-7695-1119-8.

[8] **Manning, Ch. D., Raghavan, P. a Schütze, H.** *Introduction to Information Retrieval.* Cambridge : Cambridge University Press, 2008. stránky 234-264. ISBN 0521865719.

[9] **Rijsbergen, C. J. V.** *Information Retrieval.* London : Butterworth-Heinemann, 1979. ISBN: 0408709294.

[10] *Toward optimal feature selection.* **Koller, D. a M., and Sahami.** 1996. Proc. ICML-96. stránky 284-292.

[11] *Zborník príspevkov prezentovaných na konferencii ITAT.* **Dostal, M., Ježek, K. a Krčmář, L.** Bratislava : University of P.J. Šafárik, 2010. ITAT 2010. ISBN 978-80-970179-3-4.

[12] *Boosting and Rocchio applied to text filtering.* **Schapiere, R. E., Singer, Y., and Singhal, A.** Melbourne : ACM, 1998. 21st ACM International Conference on Research and Development in Information Retrieval. stránky 215–223.

[13] *A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization.* **Joachims, T.** Nashville : Morgan Kaufmann Publishers Inc., 1997. 14th International Conference on Machine Learning. stránky 143–151.

[14] *Yahoo!* [Online] 10. března 2011. http://www.yahoo.com.

[15] *Hierarchical classification of Web content.* **Dumais, S. a and Chen, H.** Athens : ACM, 2000. Proc. of the 23rd ACM Int. Conf. on Research and Development in Information Retrieval. stránky 256-263.

[16] **Weigend, A. S., Wiener, E. D. a and Pedersen, J. O.** Exploiting hierarchy in text categorization. *Information Retrieval 3.* 1999, stránky 193-216.

[17] *Hierarchical classification of real life documents.* **Wang, K., Zhou, S. a and He, Y.** Chicago : Society for industrial and applied mathematics, 2001. Proc. of the 1st SIAM Int. Conf. on Data Mining.

[18] *ODP - Open Directory Project.* [Online] 19. února 2011. http://dmoz.org/.

[19] *Firmy.cz - Seznam.* [Online] Seznam.cz, a.s., 10. března 2011. http://firmy.cz.

[20] *Centrum.* [Online] 10. března 2011. http://centrum.cz.

[21] *Najisto.cz.* [Online] 5. února 2011. http://najisto.cz.

[22] *Rule-based text categorization using hierarchical categories.* **Sasaki, M. a and Kita, K.** La Jolla : IEEE, 1998. Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics. stránky 2827–2830.

[23] *Yahoo! as an ontology: Using Yahoo! categories to describe documents.* **Finin, T. W. a and Labrou, Y.** Kansas City : ACM, 1999. Proc. of the 8th Int. Conf. on Information Knowledge Management. stránky 180–187.

[24] *Building hierarchical classifiers using class proximity.* **Wang, K., Zhou, S. a and Liew, S. C.** Edinburgh : Morgan Kaufmann Publishers Inc., 1999. Proc. of the 25th Int. Conf. on Very Large Data Bases. stránky 363–374.

[25] Clustering to Improve Merchandise Allocation, Testing, and Forecasting: An Application of the K-Medians Algorithm. *Quantitative and Applied Economics.* [Online] 10. května 2011. [Citace: 10. května 2011.] http://espin086.wordpress.com/2011/02/27/clustering-to-improve-merchandise-allocation-testing-and-forecasting-an-application-of-the-k-medians-algorithm/.

[26] *Path based pairwise data clustering with application to texture segmentation.* **Fischer, B., Zoller, T. a and Buhmann, J.** 2001. Energy Minimization Methods in Computer Vision and Pattern Recognition. stránky 235–250.

[27] Semantic Web Frequently Asked Questions. *W3C – Semantic Web.* [Online] 9. března 2011. http://www.w3.org/RDF/FAQ.

[28] **Lee, T. B.** Linked Data – Design Issues. *W3C.* [Online] 18. června 2009. http://www.w3.org/DesignIssues/LinkedData.html.

[29] —. Linked Data. *International Journal on Semantic Web and Information Systems.* 2006, Sv. II, 4.

[30] **Seaborne, A. a Prud'hommeaux, E.** SPARQL Query Language for RDF. [Online] 2006. http://www.w3.org/TR/2006/CR-rdf-sparql-query-20060406/.

[31] **Bizer, Ch., Cyganiak, R. a Heath, T.** How to Publish Linked Data on the Web. [Online] 2007. http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/.

[32] **Klyne, G. a Carroll, J.J.** Resource Description Framework (RDF): Concepts and Abstract Syntax. *W3C Recommendation.* [Online] 2004. http://www.w3.org/TR/rdf-concepts/.

[33] *Microformats, A Pragmatic Path to the Semantic Web.* **R., Khare a T., Çelik.** Edinburgh : ACM Press, 2006. Proceedings of the 15th International Conference on World Wide Web. stránky 865-866. DOI: http://doi.acm.org/10.1145/1135777.1135917.

[34] *Statistical Challenges to Inductive Inference in Linked Data.* **Jensen, David.** 1999. Accepted to Uncertainty99: The Workshop on AI and Statistics.

[35] *Link Mining Applications: Progress and Challenges.* **Senator, T. E.** New York : ACM, 2005. stránky 76-83.

[36] **Soren, A. et. al.** DBpedia: A Nucleus for a Web of Open Data. *Lecture Notes in Computer Science.* 2007, 4825/2007, stránky 722-735.

[37] DBPedia. *DBPedia.org.* [Online] 12. dubna 2011. http://dbpedia.org.

[38] *Tabulator: Exploring and Analyzing linked data on the Semantic Web.* **Lee, T. B. et al.** 2006. Proceedings of the 3rd International Semantic Web User Interaction.

[39] *Magpie: Towards a Semantic Web Browser.* **Dzbor, M., Domingue, J. a Motta, E.** 2003. Proc. of the 2nd Intl. Semantic Web Conf.

[40] *Haystack: A Platform for Authoring End User Semantic Web Applications.* **Quan, D., Huynh, D. a Karger, D.R.** 2003. International Semantic Web Conference.

[41] *Piggy Bank: Experience the Semantic Web Inside Your Web Browser.* **Huynh, D., Mazzocchi, S. a Karger, D.** 2005. International Semantic Web Conference.

[42] **Butler, M., a další.** Longwell project page. [Online] 2006.

[43] **Dean, M. a Margerison, J.** PalmDAML. [Online] http://www.daml.org/PalmDAML/.

[44] **Steer, D.** RDFAuthor. [Online] 18. ledna 2011. http://rdfweb.org/people/damian/RDFAuthor/.

[45] **Pietriga, E.** IsaViz. [Online] 17. ledna 2011. http://www.w3.org/2001/11/IsaViz/.

[46] *CS AKTive Space: Building a Semantic Web Application.* **Glaser, H., Alani, H., Carr, L., Chapman, S., Ciravegna, F., Dingli, A., Gibbins, N., Harris, S., m. c. schraefel, Shadbolt, N.** Heraklion : Springer Verlag, 2004. stránky 417–432.

[47] *mSpace: interaction design for userdetermined, adaptable domain exploration in hypermedia.* **Schraefel, M. C., Karam, M. a Zhao, S.** Nottingham : University of Southampton, 2003. AH 2003: Workshop on Adaptive Hypermedia and Adaptive Web Based Systems. stránky 217–235.

[48] Facebook - Developer Docs. *Facebook.* [Online] Facebook, 10. března 2011. http://developers.facebook.com/docs/.

[49] *Usage Patterns of Collaborative Tagging Systems.* **Golder, S. a and Huberman, B.** 2006. Journal of Information Science. stránky 198-208.

[50] *Automatic tagging based on Linked Data.* **Dostal, M. a and Ježek, K.** Perth : Curton univerzity in Perth, 2010. IEEE International Conference on Service-Oriented Computing and Applications. ISBN: 978-1-4244-9802-4.

[51] *Automated Tag Clustering: Improving search and exploration in the tag space.* **Begelman, G., Keller, P. a and Smadja, F.** Edinburgh : ACM Press, 2006. Proceedings of the Fifteenth International World Wide Web Conference.

[52] *Improved annotation of the blogosphere via autotagging and hierarchical clustering.* **Brooks, C. H. a and Montanez, N.** Edinburgh : ACM, 2006. Proceedings of the 15th International World Wide Web Conference. stránky 625-632.

[53] *TagAssist: Automatic Tag Suggestion for Blog Posts.* **Sood, S. et al.** Colorado : AAAI Press, 2007. Proceedings of the International Conference on Weblogs and Social Media.

[54] Delicious. *Delicious.* [Online] 3. března 2011. http://www.delicious.com/.

[55] Technorati. [Online] 5. dubna 2011. http://technorati.com/.

[56] *Combining Tags and the SemanticWeb for Linked Tagging Data.* **Haklae, K., a další.** Karlsruhe : Elsevier, 2008. Semantic Web Conference 2008.

[57] *Social semantic cloud of tag: semantic model for social tagging.* **Kim, H. L., a další.** Berlin : Springer-Verlag, 2008. Proceedings of the 2nd KES International conference on Agent and multi-agent systems: technologies and applications. ISBN: 3-540-78581-7 978-3-540-78581-1.

[58] *Meaning Of A Tag: A Collaborative Approach to Bridge the Gap Between Tagging and Linked Data.* **Passant, A. a and Laublet, P.** 2008. Proceedings of the Linked Data on the Web workshop at WWW2008.

[59] Geonames. [Online] [Citace: 10. března 2011.] http://www.geonames.org .

[60] Freebase. [Online] 10. března 2011. http://www.freebase.com.

[61] *Automatic keyphrase extraction based on NLP and statistical methods.* **Dostal, M. a Ježek, K.** Proceedings of the Dateso 2011. stránky 140-145. ISBN 978-80-248-2391-1.

[62] Automated Tag Clustering: Improving search and exploration in the tag space. *Philip und Irene.* [Online] 20. dubna 2011. [Citace: 20. dubna 2011.] http://www.pui.ch/phred/automated_tag_clustering/.

[63] *Partitioning sparse matrices with eigenvectors of graphs.* **Pothen, A., Simon, H. D. a a Liou, K.-P.** 1990. Matrix Anal. Appl. stránky 430-452.

Aktivity

Recenzované publikace

- Martin Dostal a Karel Ježek. Automatic keyphrase extraction based on NLP and statistical methods. *Proceedings of the Dateso 2011*, April 20-22, pp. 140-145, ISBN 978-80-248-2391-1.

- Martin Dostal a Karel Ježek. Automatic tagging based on Linked Data. *IEEE International Conference on Service-Oriented Computing and Applications* (SOCA 2010), 13-15 Dec. 2010, Perth, WA; ISBN: 978-1-4244-9802-4.

- Martin Dostal, Karel Ježek a Lubomír Krčmář. Extrakce informací z emailů typu Call for papers. *Zborník príspevkov prezentovaných na konferencii ITAT*, ITAT 2010, Informačné technológie – Aplikácie a Teória, september 2010, ISBN 978-80-970179-3-4.

Studentské publikace

- Martin Dostal a Karel Ježek. Automatická extrakce klíčových slov s využitím statistických metod. *Studentská vědecká konference*, 2011. Plzeň, Česká republika.

- Martin Dostal. Zodpovídání dotazu v prostředí sémantického webu. *Diplomová práce*, vedoucí Karel Ježek, Západočeská univerzita v Plzni, Univerzitní 22, Plzeň, Česká republika 2009.

- Martin Dostal. Sémantické vyhledávání katastrof. *Studentská vědecká konference*, 2009. Plzeň, Česká republika.