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

AUTOMATICKÁ TVORBA KORPUSŮ

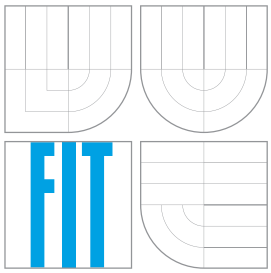
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK ŠANTAVÝ

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

AUTOMATICKÁ TVORBA KORPUSŮ

AUTOMATIC CREATION OF CORPORA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK ŠANTAVÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2009

Abstrakt

Obsahem práce je představení způsobu formátování a značkování textových dat korpusu. Nad vhodně reprezentovanými dokumenty vytváří vrstvu pro jejich vzájemné porovnání s cílem určení míry podobnosti mezi nimi. Nástroje, které výpočty podobnosti zajišťují, jsou základem automatizovaného systému pro vytváření a doplňování existujícího korpusu dat. Mezi dvěma základními přístupy je možno volit podle požadavku výpovědní hodnoty výsledku. Prostředkem pro získávání dat nových je nástroj stahování obsahu webu.

Abstract

This work is a presentation of tagging and formatting of text-data corpus. It creates a layer above suitable represented documents for their mutual comparison in order to determine the similarity among them. Tools that provide near-duplicate calculations are the basis for an automated system for creation and expansion of the existing text-data corpus. There is an option to choose between two basic approaches according to the significance of the outcome. Means of new text-data acquiring is the tool for web crawling.

Klíčová slova

korpus, duplicity, Rabin otisk, redundance, podobnost textových dat, stahování obsahu webu, vertikální text, SHA-384

Keywords

corpus, near-duplicate, Rabin fingerprint, redundancy, text-data similarity, web crawl, vertical format, SHA-384

Citace

Marek Šantavý: Automatická tvorba korpusů, bakalářská práce, Brno, FIT VUT v Brně, 2009

Automatická tvorba korpusů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. Pavla Smrže

.....
Marek Šantavý
18. května 2009

Poděkování

Chtěl bych poděkovat všem, kteří se na mé práci podíleli a především Doc. Pavlu Smržovi a Ing. Marku Schmidtovi za cennou odbornou pomoc a rady.

Obsah

Kapitola 1

Úvod

1.1 Volba tématu zpracování

Téma automatické tvorby korpusů se zaměřením na kontrolu duplicit bylo zvoleno především pro jeho širokou paletu využitelnosti se stále rozsáhlými možnostmi dalšího rozvoje známých technik a tvorby technik nových. Hlavní oblastí zájmu je zpracování elektronických textů a metodik, které jsou přitom využívány. Stejně jako i diametrální rozdílnosti ve výkonu a kvalitě výsledných dat v závislosti na využitých nástrojích.

Jedním z cílů tématu je úkol vytvořit systém zpracovávající textová data prostřednictvím automatizované činnosti a rozšíření, případně vytvoření požadovaného korpusu dat. Užším zaměřením se tato práce zabývá kontrolou duplicit textových dat, které už jsou součástí korpusu, nebo jsou kandidáty na přidání do něj. Očekávaným výstupem systému je značení duplicitních dokumentů společně s návrhem efektivního korpusu dat.

V této práci bude nastíněna metodika zpracování textových dat korpusu s využitím technik pro určování podobnosti dokumentů. Budou představeny rozličné principy způsobů určování podobnosti dokumentů stejně jako i rozličné způsoby značení dokumentů pro účel určení jejich podobnosti.

1.2 Motivace

Motivace pro vytvoření systému automatizovaného udržování konzistence a sestavování korpusu z dat nových je rozdělení na jednotlivé dílčí pod úkoly, které umožní využití výhod modulární výstavby systémů. Samotný celek (pod úkol) tvoří určování podobnosti (shody) dokumentů, který sestává z několika významných obsáhlých oblastí, které budou uvedeny v následujícím výčtu.

Pro systémy provádějící správu korpusů je nalezení a odhalení podobných (shodných) dokumentů významnou součástí. U některých dokonce míra významnosti stoupá na hranici kritického požadavku pro určení možné duplicity jednoho prvku, nebo i celé množiny prvků. Takový výstup je nadále systémem zpracováván a brán v úvahu.

Textové korpusy dat. Velké textové korpusy složené ze zpravodajských článků, knih i přepisů mluvené řeči jsou jedním z možných uplatnění systému pro automatizovanou kontrolu a určení duplicit. Korpus

Redundance dat je v podobných systémech významnou sledovanou položkou, protože v případě uchovávání článků v korpuse, v rozsahu milionů jednotek, mohou duplicitní dokumenty zabírat signifikantní množství diskového prostoru, popřípadě operační paměti.

Lingvistické účely jsou dalším faktorem, který u korpusových dat je nutno brát v úvahu. Data jsou často využívána pro korpusovou lingvistiku, korpusem podporovanou výuku jazyků a ostatní obory. Podle [9] korpusová lingvistika využívá korpusových dat ke studování jazykových fenoménů. Korpusem podporovaná výuka jazyků využívá frází a vět jazyka k efektivnímu přístupu učení daného jazyka. Ostatními obory, ve kterých jsou data použita, jsou například sledování četnosti slov pro systémy doplňování diakritiky textu. U všech těchto příkladů vede násobná přítomnost určitého dokumentu ke zkreslení dat a tím i ke znehodnocování relevantnosti výstupu systému.

Stahování obsahu webu

Web crawling, stahování obsahu webu. Na základě jednoho z výzkumů bylo odhaleno, že 30% a více obsahu je na webu duplicitních, jak uvádí [8]. Mezi tento obsah jsou řazeny i šablony stránek, které představují obecně využívanou strukturu nabídky a samotného obsahu webové stránky. Zpravodajské servery doplňují své publikované články o hlavičky a zápatí, které se vyskytují ve všech dokumentech, a to vůbec nebo pouze částečně modifikované. Společně s rozvojem textových reklamních nabídek je nutno brát v potaz i tuto kategorii. Na základě těchto údajů je nutno množství dat určených k archivaci, která jsou stažena z webu, významnou měrou redukovat. Na základě [8] jsou dokumenty při stahování webu brány v úvahu jako samostatné, i když se jedná pouze o modifikaci URL, příklad `http://www.cs.umd.edu/~pugh` versus `http://www.cs.umd.edu/users/pugh`. Spjatou oblastí jsou i webové vyhledávače, u kterých je záhodno odstranění duplicit z výsledku vyhledávání klíčových slov.

Plagiátorství

Plagiátorství. V souvislosti s rozvojem internetu a usnadnění přístupu k velkému množství informací v elektronické podobě nabývá na významu systém pro kontrolu plagiátorství zdrojových kódů a textových dat. Kontrola plagiátorství slouží nejenom pro účely dozorování respektování autorského zákona, ale i zachování studijní práce na školské úrovni. V souvislosti s autorským zákonem přichází v úvahu i nutnost uvádění správných citací při využití děl dalších autorů.

Kapitola 2

Teoretická část práce

Následující text je zaměřen na výstavbu nástrojů pro systém zpracování textových dat a jejich zařazení do existujícího korpusu. Rozsah zaměření pojme celou oblast zpracování, formátování a reprezentace dat. Budou představeny základní způsoby určování podobnosti dokumentů, počínaje zpracováním původních textových dat (před příprava) pro další zpracování i samotné určení podobností mezi dokumenty. Fáze přípravy textových dat není nezbytně nutná, ale většina pokročilých technik tuto fázi zařazuje, vysvětlení jejího významu bude podáno dále.

Každé metodice bude věnována samostatná část, ve které bude vyjádřen její princip a podrobněji popsána bližší specifika, požadavky té které metody.

2.1 Úroveň podobnosti dokumentů

Před pokračováním v další části textu je nezbytně nutné vymezit základní pojmy shody (duplicity), podobnosti, úplné neshody. Tyto pojmy pomohou pochopit význam určování podobnosti dokumentů a vyjádří základní kategorie, do kterých můžeme dokumenty členit [12]. Tyto kategorie jsou uvedeny v tabulce 2.1, kdy zdrojem těchto dat byl [11].

Na základě této tabulky je zřejmé, že míra podobnosti dvou dokumentů se může měnit od naprosté neshody až po úplnou shodu. Kontrola prováděná lidskými zdroji by ve velkých počtech dokumentů byla neúměrně nákladná a časově náročná.

Systém pro automatizované určování podobnosti dokumentů je možno rozdělit do třech fází, jejichž nástroje je možno modulárně skládat a tím pádem výstup jedné fáze může být zpracován *1..n* dalšími nástroji.

Fáze určování podobnosti

1. Před přípravná fáze, příprava textových dat dokumentu
2. Přípravná fáze, zpracování textových dat dokumentu a tvorba značení dokumentu
3. Finální fáze, výpočet podobností na základě značení dokumentů

2.2 Triviální metody

První možnou a nejjednodušší metodou porovnávání textových dat je jejich prosté srovnání. Avšak podle tabulky 2.1 je zřejmé, že tímto způsobem nebudeme sto postihnout celou škálu variací podobnosti dokumentů. Navíc je tento způsob přístupu velmi neefektivní z hlediska výpočetního výkonu při použití nad kolekcí dat, kdy dosahuje kvadratické složitosti.

Popis podobnosti	Text prvního dokumentu	Text druhého dokumentu
Úplná shoda	**A právě na jeden takový stroj jsem se zaměřil v dnešním testu. Jmenuje se Lenovo ThinkPad W700ds, i když tenký tedy rozhodně není. Je to přesně ten notebook, o kterém se před časem mluvilo téměř všude. Má totiž dva LCD panely, což se u notebooku jen tak nevidí. To ale rozhodně není jediná přednost tohoto téměř 100 000 korun stojícího zázraku.**	**A právě na jeden takový stroj jsem se zaměřil v dnešním testu . Jmenuje se Lenovo ThinkPad W700ds, i když tenký tedy rozhodně není. Je to přesně ten notebook, o kterém se před časem mluvilo téměř všude. Má totiž dva LCD panely, což se u notebooku jen tak nevidí. To ale rozhodně není jediná přednost tohoto téměř 100 000 korun stojícího zázraku.**
Shoda odstavce	**Zamrzí však například jen jednořádkový Enter nebo prohozený Fn a Ctrl, což může někomu dělat problém. Také šipky by mohly být vydělené. Na druhou stranu u prodejní verze je vpravo pod klávesnicí ještě malý tablet, takže si nejsem jist, jestli je tam dostatek prostoru.** S klávesnicí jsem byl celkově spokojen, ale jako obvykle se najdou i chyby. Nejsou však nijak zásadního charakteru, protože jak osazení klávesnice, tak ergonomie kláves je výborná a na klávesnici se pohodlně píše. Samozřejmostí je u takto velkého notebooku také numerický blok, který je navíc prakticky oddělný.	**Zamrzí však například jen jednořádkový Enter nebo prohozený Fn a Ctrl, což může někomu dělat problém. Také šipky by mohly být vydělené. Na druhou stranu u prodejní verze je vpravo pod klávesnicí ještě malý tablet, takže si nejsem jist, jestli je tam dostatek prostoru.** Nad klávesnicí jsou ještě tři tlačítka pro ovládání hlasitosti a tlačítko ThinkVantage. To zapíná tzv. Productivity Cetner, které by se dalo označit jako soubor záložek s odkazy na ty nejdůležitější programy či nastavení. Jejich složení je možné samozřejmě změnit.
Přeskupení odstavců	**Na levém boku je první část vývodu chlazení, FireWire, dvojice USB portů a dvojice ExpressCard slotů.** Zepředu je přepínač k WiFi, čtečka paměťových a na pravé straně ještě audio konektory. **Na pravé straně jsou další tři USB porty, zásuvka pro modem, optická mechanika, a další vývody chlazení.**	Také zezadu jsou otvory pro vývod horkého vzduchu. Dále jsou uprostřed výstupy na monitor, ethernet a napájení notebooku. **Na pravé straně jsou další tři USB porty, zásuvka pro modem, optická mechanika, a další vývody chlazení.** **Na levém boku je první část vývodu chlazení, FireWire, dvojice USB portů a dvojice ExpressCard slotů.**
Úplná neshoda	Sehnat notebook s dobrým displejem, to je obrovská věda. Troufnu si tvrdit, že 99 % z vás má notebook i TN+ panelem, který by si někteří z vás kvůli horším barvám i pozorovacím úhlům nejspíš nepostavili na stůl. Jenže co dělat, když potřebujete dobrý displej i na cestách (například pro úpravu fotek či videa)? Od toho je tu segment profesionálních notebooků. Ty stojí sice často neuvěřitelné peníze, ale pokud potřebujete zákazníkovy ukázat návrh v té nejlepší možné kvalitě či jen potřebujete výkonný notebook, jsou často jedinou alternativou.	A právě na jeden takový stroj jsem se zaměřil v dnešním testu . Jmenuje se Lenovo ThinkPad W700ds, i když tenký tedy rozhodně není. Je to přesně ten notebook, o kterém se před časem mluvilo téměř všude. Má totiž dva LCD panely, což se u notebooku jen tak nevidí. To ale rozhodně není jediná přednost tohoto téměř 100 000 korun stojícího zázraku.

Tabulka 2.1: Kategorie podobnosti dokumentů

2.3 Zpracování vstupních textových dat, před přípravná fáze

Potřeba zefektivnění procesu určení podobnosti dokumentů vedla k zavedení fáze před přípravy textových dat i s následnou přípravnou fází pro snadnější určení podobností dokumentů. Nad těmito daty jsou dopočítány další pomocné údaje, které hrají významnou roli v samotné konečné fázi činnosti systému [1]. Toto nám dovoluje vykonat zpracování textových dat dokumentu pouze jednou a posléze již pracovat pouze s pomocnými daty, která bývají často méně náročné na uchování a s obsaženou příslušnou vypovídací hodnotou.

Formátovací značky

V rámci před přípravné fáze dochází ke zpracování dokumentů v jejich původní podobě se všemi formátovacími i lingvistickými značkami. Pro potřeby následujících fází mohou být tyto značky kompletně odstraněny a to tak, že výstupem budou čistá textová data. Alternativní možností je zachování potřebných formátovacích značek, například uvození a závěr odstavce. Toho je využíváno pro metody využívající hranic odstavců jako hranice celků, které jsou brány v úvahu. Význam této fáze spočívá ve schopnosti odstranění netisknutelných znaků, znaků cizí národní abecedy, nebo i odstranění celých nežádoucích textových

bloků s jasným vyznačením obsahu, například reklamy.

2.4 Pokročilé značení dokumentů, fáze přípravy dat

Ve fázi přípravy dat dochází k ohodnocení a označení dokumentů za účelem jejich specifického a jednoznačného rozlišení. V dalším textu budou představeny metody jednoprůchodové, které nevyužívají žádných závislostí mezi jednotlivými dokumenty. Stejně jako i metody víceprůchodové, kdy dokumentu jsou mezi sebou spjaty určitou vazbou, například procentuálním vyjádření četnosti slov v rámci celého korpusu dat.

2.4.1 Metoda, Rabin fingerprint odstavců

První představenou metodou pro značení dokumentů obsahujících textová data je metoda využívající transformační funkci (*Rabin fingerprint*) ke značení uvažovaných elementů [5]. Přípravná fáze tohoto přístupu bere v úvahu dokument jako textová data, se kterými dále pracuje. Dokument je rozdělen na jednotlivé dílčí jednotky, v uvažovaném případě slova, která jsou nazývána *pozicemi* (případně anglickým výrazem *token*). Podmnožiny těchto výsledných *pozic* jsou transformovány pomocí hashovací funkce v případě této metodiky funkce *Rabin fingerprint*. Tyto *pozice* funkce transformuje do elementů zvaných *zákryt*, který reprezentuje anglické slovo *shingle*. Celkový počet těchto *zákrytů* je určen jako *n-k+1*, kde počtu *pozic* dokumentu odpovídá *n* a velikost podmnožin *k*. V případě zvolení velmi malé velikosti podmnožin, limitně až 1, je vytvořeno velké množství *zákrytů* a benefity plynoucí z této transformace jsou mizivé. Proto zavádíme pojem *supershingle* (jako vhodný český ekvivalent bude pro něj použit *super-zákryt*), který slučuje podmnožinu nepřekrývajících se *zákrytů* a provádí na touto podmnožinou další transformaci pomocí stejné transformační funkce. Výsledný vektor *super-zákrytů* reprezentuje daný dokument a slouží pro potřeby porovnání podobnosti mezi jednotlivými dokumenty. Takový výsledný vektor se však může lišit ve velikosti v závislosti na tom, zda-li je uvažován konstantní počet *super-zákrytů* nebo počet závislý například na velikosti dokumentu. U této metody je ve výsledku zahrnuta posloupnost jednotlivých *pozic* se zachováním jejich pořadí. To znamená, že při změně pořadí a zachování stejných *pozic*, bude výsledná množina vektorů rozdílná.

Transformace pozic

2.4.2 Metoda, 384 bitový otisk dokumentu

Další představenou metodou využívající k reprezentaci dokumentu vektor je metoda zvaná *simhash* [6]. Ve fázi přípravy textu je dokument rozdělen pomocí *tokenizace* (proces operující nad textovými daty, jehož výsledkem je vektor váhovaných *pozic*). Souběžně s vytvořením množiny *pozic* dojde i k vytvoření vektoru dokumentu předem dané velikosti f (f bit), vektor V, všechny jeho dimenze jsou nastaveny na počátku na 0. Pomocí hashovací funkce, *SHA-384* funkce, jsou jednotlivé dimenze vektoru váhovaných *pozic* transformovány do podoby *otisků*. V případě využití metodiky *simhash* provádíme převod mnoho-dimenzionálního vektoru dokumentu vzniklého transformací do předem vytvořeného výsledného vektoru. Jednotlivé transformované *pozice* jsou převedeny do bitové reprezentace, kdy bit 0 způsobí snížení příslušné dimenze výsledného vektoru o hodnotu váhy této dimenze *pozice* a bit 1 způsobí zvýšení příslušné dimenze výsledného vektoru o hodnotu váhy dimenze *pozice*. Na konci přípravné fáze a po zpracování všech *pozic* dokumentu je výsledný vektor normován a to tak, že pozitivní hodnotě dimenze přísluší 1 a negativní hodnotě 0. Výsledkem zpracování dokumentu uvedenou metodikou je vektor předem daného počtu dimenzí,

simhash

jejichž hodnoty nabývají hodnot $\{0, 1\}$. Oproti předchozí metodě neuvažuje tato metoda pořadí slov, ale pouze jejich výskyty v rámci zpracovávaného dokumentu. Výsledný vektor je tedy nezávislý na pořadí *pozic*, ale je závislý na četnosti jejich výskytů.

2.4.3 Collection statistics, statistická metodika

Při popisu této metodiky značení a prací s dokumenty bylo vycházeno ze zdroje [12]. V přípravné fázi zpracování dokumentu jsou jeho textová data rozdělena do jednotlivých *shluků*, tento název vznikl z překladu anglického slova *chunk*. Velikost těchto *shluků* závisí na velikosti dokumentu a to tak, že čím je dokument (z pohledu velikosti textových dat) větší, tím je velikost *shluku* a množství textových dat, které obsahuje, větší. Při rozdělování dokumentu do jednotlivých *shluků* je nutno brát v potaz i logické dělení dokumentu na odstavce a nadpisy, kdy *shluk* nesmí přesahovat odstavec a zasahovat do odstavce nebo nadpisu jiného. Z každého *shluku* je vybrána jediná *pozice*, kterou označíme jako *člen* t^*, překlad anglického *term*. Tento *člen* splňuje podmínku minimálního výskytu vůči ostatním *pozicím shluku*. Vycházíme ze vztahu 2.1, kde $df(t_j)$ je počet dokumentů obsahujících tento *člen* (t_j) a pak $\arg\min$ je *člen* s minimální $df()$ hodnotou. Jedná se tedy o výběr *členu* s minimálním výskytem nad celou kolekcí dat v rámci *shluku*. Kolem takto zvoleného *členu* je vytvořeno rozpětí textových dat (podmnožina *pozic shluku*), pro které jsou voleny hodnoty rozsahu povětšinou 3 nebo 5. Výsledkem je množina vektorů *pozic*, v jejichž středu je obsažen *člen*.

Tvorba shluků

$$t^* = \arg_j \min df(t_j) \tag{2.1}$$

Ke shrnutí dalších alternativních postupů, které jsou uvedeny v [12], patří například vytváření *otisků* z podmnožin znaků obsažených v textových datech. Tímto způsobem lze redukovat výsledné množství *otisků* připadajících na dokument až na polovinu z původního množství.

Další metodou je využití *členů* s nejčastějším výskytem ve *shluku* textu, čímž se tyto *členy* stávají nejvhodnějšími reprezentanty pro daný dokument. Tato metoda využívá filtru pro odstranění nevhodných (raritních, chybně hláskovaných slov) *členů* limitní konstantní hodnotu, která *členy* pod touto úrovní automaticky vyřazuje.

2.5 Určování podobnosti dokumentů, finální fáze

V návaznosti na předchozí přípravné fázi je třeba provést určení vzájemné podobnosti textových dat kolekce dokumentů. Výstupy výše uvedených metod jsou množiny vektorů čítající *1..n* vektorů reprezentujících dokument. Při hledání průniku mezi vektory dokumentů hraje roli rychlost zpracování i velikost a počet vektorů připadajících na dokument. Na základě porovnání množin vektorů dvou dokumentů jsme schopni určit vzájemnou shodu, neshodu, ale i mezi stupeň, podobnost (míru podobnosti).

2.5.1 Rabin fingerprint odstavců

Pro tuto metodu je charakteristická množina vektorů, která je určená k porovnání s dalšími dokumenty. Výsledná hodnota určená pro podobnost dvou dokumentů jakožto dvou množin vektorů je definována vztahem [5]:

$$sim(x, y) = \frac{|S(x) \cap S(y)|}{|S(x) \cup (y)|} \tag{2.2}$$

Metoda splňuje následující axiómy, podle [7]:

1. nezápornost $\forall x, y \in D, sim(x, y) \geq 0$
2. symetrie $\forall x, y \in D, sim(x, y) = sim(y, x)$

2.5.2 384 bitový otisk dokumentu

Kosinova vzdálenost

Jelikož výstupem této metody při přípravné fázi je jediný vektor ohodnocující celý dokument, odpovídá určení podobnosti výpočtu *Kosinovy vzdálenosti* mezi vektory obou dokumentů [1]:

$$dot(x, y) = \sum_i x[i] \cdot y[i] \tag{2.3}$$

Metoda splňuje následující axiómy, podle [7]:

1. nezápornost $\forall x, y \in D, dot(x, y) \geq 0$
2. symetrie $\forall x, y \in D, dot(x, y) = dot(y, x)$
3. reflexivita $\forall x \in D, dot(x, x) = 0$
4. pozitivita $\forall x, y \in D, x \neq y, dot(x, y) > 0$
5. trojúhelníková nerovnost $\forall x, y, z \in D, dot(x, y) \leq dot(x, z) + dot(z, y)$

2.5.3 M-Tree strom, 384 bitový otisk

Podobnostní vyhledávání

Tradiční způsoby vyhledávání vycházejí z určení přesné shody (nebo neshody) mezi dvěma entitami určité množiny. Tento způsob vyhledávání je spjat se základními datovými typy, jakými jsou řetězce a celá čísla. Samotný průběh vyhledávání není tedy žádným způsobem řízen a pro každé porovnání je určena konečná výsledná hodnota. Jednou z nových metod vyhledávání na základě nikoliv přesné shody, ale podobností je metoda M-Tree [7, 13]. Příčiny jejího vzniku jsou především v potřebě efektivního vyhledávání složitějších datových struktur. S potřebou provádět podobností vyhledávání, ale i realizování dotazů nad takovými datovými strukturami. Významnou součástí M-Tree je i koncept metrického prostoru, který definuje jak vzdálenostní funkci mezi objekty, tak i podobu samotných objektů, mezi kterými bude podobnost určována.

Mezi jedny z hlavních výhod využití M-Tree patří i vzdálenostně řazené vyhledávání. Tento způsob vyhledání umožňuje provést vyhledávání nejbližších, ale i nejvzdálenějších prvků od prvku zadaného. V případech takto řazeného způsobu vyhledávání je využito metrik vzdáleností mezi jednotlivými prvky. Při tvorbě algoritmu M-Tree byl kladen důraz především na dynamiku sestavování stromu, ale i efektivitu jeho samotného vyvážení.

Metoda splňuje následující axiómy, podle [7]:

1. nezápornost $\forall x, y \in D, d(x, y) \geq 0$
2. symetrie $\forall x, y \in D, d(x, y) = d(y, x)$

3. reflexivita $\forall x \in D, d(x,x) = 0$
4. pozitivita $\forall x, y \in D, x \neq y, d(x,y) > 0$
5. trojúhelníková nerovnost $\forall x, y, z \in D, d(x,y) \leq d(x,z) + d(z,y)$

2.6 Data mining, web crawling

Další potřebnou částí při tvorbě systému pro automatizovanou tvorbu korpusu je i zajištění pravidelného rozšiřování obsahu tohoto korpusu dat. Zdrojů s obsahem dat v elektronické podobě je celá řada, pro přehlednost budou uvedeny některé z nich. Seřazení bylo voleno tak, že odpovídá stoupající tendenci v množství dat a klesající tendenci v pravidelnosti dostupných aktualizací:

Elektronická korespondence, do této skupiny jsou řazeny emaily, elektronické konference [12] a diskuzní skupiny. K aktualizaci těchto zdrojů dochází ve špičkách (časové období největší aktivity) velmi často, avšak množství takto dostupných dat je relativně velmi malé.

Zpravodajské servery, míra aktuálnosti těchto zdrojů se pohybuje v řádu jednotek až desítek minut. Pravidelnost aktualizací má větší tendenci se soustřeďovat do jistého vymezeného časového úseku. Množství textových dat připadající na jeden dokument je však řádově větší, než u předchozí kategorie.

Obsahy knih, e-booky, pravidelnost aktualizace tohoto zdroje se přesouvá do dnů, týdnů. Na druhou stranu kvantum takto dostupných dat je v rámci jednoho dokumentu největší ze všech uvedených. Pro příklad bude uvedeno množství slov v knize (Harry Potter, Kámen Mudrců od J. K. Rowling), přesahuje hranici 70 000.

Ve fázi doplňování dat, která jsou získána tímto způsobem je nutno brát v úvahu, jaké kvalitativní požadavky jsou definovány na podobu výsledného korpusu dat [9]. Jedná se jak o zaměření té které kategorie, kdy zpravodajský server může být zaměřený na sport, obsahy knih být hlavně vědeckofantastickými díly a elektronická korespondence probíhat v rámci technického oddělení specializované firmy. Podle takovéhoto nástinu by po spojení všech těchto dostupných dat vznikl korpus značně nevyrovnaný a jeho vypovídací hodnota by byla velmi malá. Proto je při tvorbě korpusu nutno předem určit z jakých, že dat se má korpus skládat a k čemu má být použit. Dále je nutno brát v úvahu, že v rámci stejného směru zaměření dochází k rozdílnostem jazykového projevu, například mezi seriózním zpravodajským serverem a přátelskou komunikací dvou lidí (odborníků).

Významný zdroj dat

Jak již bylo uvedeno výše, jedním z významných zdrojů dat je internet, který svým rozsahem informací v elektronické podobě představuje velmi výhodný zdroj. Pro internet je typický i další fakt, že množství dostupných informací má tendenci nabývat na kvantitě, případně měnit svůj obsah oproti původnímu stavu. V závislosti na posledním uvedeném faktu přichází v potaz potřeba systému, který při získávání informací z již jednou použitého zdroje, musí provést kontrolu na podobnost potenciálně staženého dokumentu s obsahem již sestaveného korpusu dat.

2.7 Podobnost v krátkých a dlouhých odstavcích

Pro potřeby dalšího výkladu budou rozděleny odstavce z pohledu jejich velikosti (počtu slov) na dvě kategorie, dlouhé a krátké. Krátké odstavce představují množiny čítající řádově jed-

notky až desítky slov. Velikosti odstavců dokumentů se mohou značně lišit, což může mít za následek, že při porovnání dokumentů na úrovni odstavců bez úvahy velikosti odstavců budou dokumenty označeny jako nepodobné. I když v takovém případě se dokumenty shodují ve velkých odstavcích, kde je obsažena většina textových dat dokumentu, mohou zde být krátké odstavce obsahující například řádkovou reklamu nebo titulek. V takovém případě krátké odstavce svým počtem výskytů způsobí onen jev nepodobnosti, i když je na první pohled u obou dokumentů jasná podobnost.

Obsah dlouhých odstavců je zřejmý a představuje samotné sdělení zpravodajského článku, či obsahu díla u textu knihy. Na druhou stranu obsahy krátkých odstavců je možno rozdělit do několika kategorií, které jsou úzce spjaty se zdrojem, odkud jsou dokumenty získávány. Tyto krátké odstavce většinou sestávají z jednoznačného sdělení s vyjádřenou vlastní sémantikou, popřípadě tato sémantika vychází z kontextu článku (dokumentu). Mezi několik základních kategorií obsahu krátkých odstavců spadajících z velké části mezi součásti zpravodajských článků jsou řazeny informace o:

Obsah krátkých odstavců

Počasí. V závislosti na konečném slovním výčtu stavů, ve kterých může být předpověď počasí určena (jasno, polojasno, zataženo) je pravděpodobnost podobnosti mezi odstavci obsahující informace o počasí velmi vysoká. V případě, že předpovědi počasí tvoří samostatné oddělené rubriky, které je možno nějakým způsobem vymezit, je záhodno tyto rubriky při sestavování korpusu dat vůbec nebrat v potaz. Situace se ovšem komplikuje v případě informací o počasí, které jsou součásti článku a jsou tedy potenciálně brány v úvahu jako informativní sdělení týkající se konkrétního článku (dokumentu).

Autorech. Především jako doprovodná informace zpravodajských článků o autorství toho kterého článku. Přítomnost odstavců obsahujících informace o autorech může způsobit snížení kvality výsledného korpusu dat. Příkladem je vyhledávání nejčastěji se vyskytujících českých jmen a příjmení, kdy textová data získaná z článků doplněných o jméno autora tuto hodnotu zkreslovala. Pro vyjádření autora článku jsou používány i na první pohled nesrozumitelné zkratky, které jsou slepením a zkrácením jména i příjmení.

Sportovním sdělení. Relevantní informace o aktuálním průběhu sportovních utkání nebo výsledkový servis zápasů představuje pouze strohé vyjádření zúčastněných stran a dosaženého stavu v rámci utkání. Jako doplňující údaje, které jsou k zápasu uvedeny, mohou být střelci branek nebo průběžné pořadí v turnaji. Drtivá většina dat získaných z podobně zaměřených zdrojů jsou jména sportovců, kluby a číselné hodnoty výsledků.

Kontextu části dokumentu. Pro vyjádření stavby dokumentů a článků se kromě optického rozdělení do odstavců i barevně oddělených celků využívá slovní vyjádření kontextu částí textu. Příkladem budou uvedeny některé často se vyskytující označení kontextu části článku: “Převzato z”, “Diskuze k článku”, “Kapitola”, “Fotogalerie”. Několik zde uvedených příkladů je možno naleznout jak v elektronických zdrojích, například zpravodajský server, tak i v knižní podobě, kdy jsou slovem “Kapitola” odděleny jednotlivé kapitoly.

Zaměření korpusu určuje závislost na významu míry užití uvedených kategorií krátkých odstavců. Pro korpusy spisovného jazyka jsou však textová data těchto odstavců příliš úzce zaměřená, že jejich použití způsobuje zkreslení ostatních dat v korpusu dat. V takovém případě je nutno použití specializovaného nástroje, který je schopen takovéto odstavce označit a navrhnout k odstranění.

Obsah dlouhých odstavců

Výskyt podobnosti je možný i v dlouhých odstavcích. Příkladem mohou být elektronické zpravodajské články, u kterých dochází často ke korekturám i po jejich zveřejnění. Jedná se o rozšíření zprávy o další odstavce nebo modifikace titulku článku pro zvýšení jeho vypovídací hodnoty. Při modifikaci titulku článku však dochází k zachování jeho obsahu, pokud systém dva články mezi sebou rozlišuje pouze na základě jejich titulku, projeví se

zjištění modifikovaného titulku se zachovaným obsahem až při zpracování systémem pro určení podobnosti dokumentů.

Citace představují kategorii podobností, kterou není možno přesně zařadit mezi podobnosti krátkých nebo dlouhých odstavců. Svým potenciálně variabilním rozsahem od krátkých víceslovných až větných citací se může jednat o citace celých pasáží textu. Není možno tedy citace striktně zařadit do jakékoliv z obou hlavních kategorií.

2.8 Rozptýlení dokumentů v jiných dokumentech

Pojem rozptýlení dokumentu označuje stav, kdy je dokument rozptýlen v $1..n$, $n \in \mathbb{N}$ dokumentech. Hraniční stav, kdy je dokument rozptýlen v *1* dokumentu, označuje stav shody obou dokumentů, kdy jsou všechny části rozptýleny pouze v jediném dokumentu. Množství dokumentů, ve kterých k rozptýlení dochází, závisí na uvažovaných jednotkách, u kterých rozptýlení sledujeme. Těmito jednotkami mohou být všechna textová data dokumentu, odstavce, věty a další. Při velmi malých jednotkách, které jsou v nástroji pro rozptýlení kontrolovány, je nutno brát v potaz větší počet výskytů rozptýlených dokumentů a řádově vyšší výpočetní nároky na jejich odhalení. Tento jev je však nežádoucí především z důvodu zachování sémantiky spojení těchto jednotek, které vytváří dokument.

Nástroj pro kontrolu rozptýlení dokumentů, který je součástí automatizovaného systému pro udržování a tvorbu korpusu dat, slouží k zajištění označení dokumentů, které jsou rozptýleny v jiných. Tento nástroj může provádět svou činnost jak nad již sestaveným korpusem, ale i nad daty, která jsou kandidáty na vložení do tohoto korpusu. Rozptýlený dokument se stává kandidátem na odstranění z korpusu dat z důvodu duplicity obsahu jednotek, které jsou již v korpusu přítomny. Takový dokument je možno označit za sestavu ("slepenec") z částí dokumentů jiných. Tento nástroj představuje další z kolekce nástrojů, které slouží především pro zachování kvality korpusu dat.

Příklad textových dat, ve kterých může docházet k vzájemnému rozptýlení obsahů dokumentů, je následující:

Výskyt rozptýlení

Souborná vydání textů představují sestavení několika článků, dokumentů nebo knih. Tento soubor může sestávat z dokumentů, které jsou již v korpusu obsaženy. V takovém případě má větší význam uchovat jednotlivá samostatná díla a nikoliv jejich souborné vydání, protože samostatná díla jsou opatřena vlastní identifikací a jednoznačným rozdělením.

2.9 Formátování textových korpusů

Pro značkování textových korpusů je vhodným adeptem na použití vertikální reprezentace textu. Vertikální značkování je rozšířená a modifikovaná podoba textové reprezentace dat. Text ve vertikální reprezentaci je rozdělen na jednotlivé *pozice*, kdy každá taková *pozice* je umístěna na samostatném řádku [9]. Dokument je tedy reprezentován v podobě vertikální. Výhoda vertikální reprezentace spočívá především ve snadném zpracování GNU utilitami (sed, cat, grep) a programovým řešením, kdy stačí tímto způsobem formátovaný soubor číst po řádcích. Časté je užití této reprezentace v případě rozsáhlých textových korpusů dat. Rozšíření vertikální reprezentace textu spočívá v doplnění o syntaktické a lingvistické značky, krátký přehled nejvýznamnějších značek i s příklady je uveden v tabulce 2.2.

Vertikální text

Mezi značkami formátování dokumentu mohou být i doplňující značky, například výstupu morfologického analyzátoru, které náleží ke slovu bezprostředně následujícímu. Příkladem dokumentu formátovaného do vertikální reprezentace je úryvek z článku, který je

Vertikál	Popis	Příklad použití
<doc>	Identifikační řádek dokumentu	<doc id=”mf/2008/8/4/10/43”lang=”cs”... >
<head>	Titulek článku	<head> Vylepšete si zahradu </head>
	Odstavec textu	 Dnes bude jasno
<g>	Spojení předcházejícího a následujícího vertikálu	10 <g> .

Tabulka 2.2: Vertikálních formátovací značky

uveden v příloze technické zprávy.

2.10 Značkování textu, morfologická analýza

Morfologický analyzátor slouží ke zjištění morfologických kategorií slova. Tyto kategorie mohou být doplněny do formátování textu, ale jejich přítomnost není striktně vyžadována ani není plně zajištěna z důvodu, který bude nastíněn dále. Nástroj morfologického analyzátoru ke své činnosti využívá slovníku a určení je tak závislé od jazykových výrazů, které jsou v něm obsaženy. Příkladem vhodného morfologického analyzátoru je *libma* [4] nebo *ajka* [10].

Pro případ, že je potřeba provést určení morfologické kategorie slova i přes nepřítomnost jazykového výrazu ve slovníku, který morfologický analyzátor využívá, může být využit například nástroj *fsa_guess* (Finite State Automata) [3]. Tento nástroj se snaží o přiřazení požadovaného slova ke slovu obsaženému ve slovníku, které je zvoleno jako nejvhodnější adept.

Finite State Automata

2.11 Významné pojmy

Pro snadnější pochopení budou uvedeny a vysvětleny základní pojmy, které se vyskytují v textu. Ke každému pojmu bude připojen jeho stručný popis a vysvětlení významu jeho použití v této technické zprávě.

2.11.1 Hashovací funkce

Představuje převod vstupních dat do mapovaného prostoru [2], tím je rozuměno:

$$h : \sum^{*} \rightarrow \sum^{n}, n \in N \tag{2.4}$$

$$D = \sum^{*} \tag{2.5}$$

Funkce h je hashovací funkcí. Z výrazu plyne, že řetězec textových dat libovolné délky, je mapován do řetězce délky omezené. To způsobí, že mapování není nikdy injektivní [2].

Společně s hashovací funkcí může být použita i kompresní funkce, která provádí mapování podle

$$h : \sum^{m} \rightarrow \sum^{n}; n, m \in N, m > n \tag{2.6}$$

$$D = \sum^{m} \tag{2.7}$$

Funkce h je kompresní funkcí. Řetězec omezené délky je mapován do řetězce kratší délky řetězce původního.

Pro potřeby algoritmů je nutno zachovat vysokou rychlost strojového zpracování pří výpočtu $h(x), \forall x \in D$. Funkce h je funkcí jednostrannou, což plyne z toho, že není možno určit funkci inverzní k funkci h. Kolize funkce je takový stav, kdy h pro dvojici $(x, x') \in D^2$ současně platí $x \neq x'$ a $h(x) = h(x')$. Kolize funkcí

Slabě bezkolizní, kolize dvojice (x, x') je možná
Silně bezkolizní, ke kolizi dvojice (x, x') nemůže dojít

2.11.2 Otisk (fingerprint), otiskování (fingeprinting)

Pro potřeby strojového zpracování je využívána podoba k bitového *otisku* (anglicky zvaného *fingerprint*) pro *členy* a *pozice*, kdy *otisk* je definován (podle [14]) jako:

$$F = \{f : \Omega \rightarrow \{0, 1\}^k\} \tag{2.8}$$

Tvorba k bitového vektoru probíhá prostřednictvím transformační funkce, podle které jednotlivé dimenze vektoru nabývají hodnoty pouze $\{0, 1\}$. Požadavkem na transformační funkci je převod textového řetězce libovolné délky na řetězec konstantní délky.

Pro totožné *členy* a *pozice*, jsou výsledkem transformační funkce vektory A, B, které jsou definovány jako:

$$\forall A, B \in \Omega \tag{2.9}$$

$$f(A) = f(B) \Rightarrow A = B \tag{2.10}$$

V případě využití transformačních funkcí *otiskování* (z anglického *fingerprinting*), může nastávat situace falešně pozitivních shod:

$$f(A) = f(B) \; kdy \; A \neq B \tag{2.11}$$

Tento jev je způsoben použitou transformační funkcí, která je slabě bezkolizní, a při praktickém použití transformační funkce je nutno brát v úvahu, že se tímto způsobem zanáší do výsledku určených podobností jistá míra chyby.

2.11.3 Rabin fingerprint

Pro potřeby transformace textových dat, kdy jedním z hlavních požadavků je časová nenáročnost a dodržení požadovaných vlastností. Mezi tyto vlastnosti patří zmenšení velikosti *otisku* oproti původní velikosti textových dat. Takovou funkcí je transformační funkce, *Rabin fingerprint*, která tyto požadavky splňuje. Vytvoření k bitového *Rabin fingerprint* probíhá v těchto krocích [14]:

Je dán textový řetězec A:

$$A = a_m a_{m-1} \dots a_1 \tag{2.12}$$

K bitový otisk *Rabin fingerprint* je spočítán následujícím způsobem:

1. Nechť
$$A(t) = a_m t^{m-1} + a_{m-1} t^{m-2} + \dots a_2 t + a_1 \tag{2.13}$$

2. Volba neredukovatelného polynomu
$$P(t) = p_k t^k + p_{k-1} t^{k-+} + \dots a_0 \tag{2.14}$$

3. Vypočet *otisku Rabin fingerprint*
$$f(A) = A(t) \ mod \ P(t) \tag{2.15}$$

2.11.4 N-tice, skupiny slov

Pokud je dána množina slov, pak podmnožinou *n-slov* této množiny je *n-tice* (*n-gram*, skupina slov) [6]. Jedná se o souhrnné označení skupiny slov se zachováním jejich vzájemného pořadí, kterého bývá při jejich dalším zpracování využíváno. Příklad *n-gramů* je uveden v následujícím výčtu:

- *3-gram* (*trigram*): “vítáme všechny dámy”, “pořadí textu je”
- *5-gram* (*pětice*): “vítáme všechny dámy a pány”, “pořadí textu je závislé na”

2.11.5 Shluk (chunk)

Jednotkou rozdělení textových dat je *shluk* (překlad anglického termínu *chunk*), který vzniká na základě činnosti systému pro dělení dokumentu. Velikost *shluku* závisí především na dalším použití a požadované přesnosti rozlišení.

2.11.6 Pozice (token), člen (term)

Rozdělení textových dat dokumentu na menší jednotky probíhá specificky podle daných pravidel. Tyto menší jednotky se nazývají *pozice* (překlad anglického výrazu *token*) a proces rozdělení textových dat na tyto jednotky se nazývá *tokenizace* [9].

Vyšší úroveň oproti *pozici* tvoří *člen* (anglický překlad původního *term*), který vzniká z *pozice* po splnění zadaných lingvistických pravidel. Příkladem *členu* je *pozice*, která se v dokumentu objeví pět a vícekrát nebo *pozice*, jejíž délka řetězce je menší *10*.

2.11.7 Zákryt (shingle), super-zákryt (supershingle)

Při transformaci množiny *pozic* dokumentu v rámci přípravné fáze značení dokumentu vzniká množina vektorů *otisků*, které nazýváme *zákryt* (překlad anglického slova *shingle*). *Zákryt* je transformovaná podoba *pozice* tedy textové jednotky dokumentu [5].

Optimalizace zákrytů

Množina takto vzniklých *zákrytů* může být značně rozsáhlá a pro potřeby rychlého zpracování proto nevhodná. Způsobem, jak zajistit dostatečně rychlé zpracování, je vytvoření tak zvaných *super-zákrytů* (překlad anglického slova *supershingle*), které jsou výsledkem transformace nepřekrývající se podmnožiny *otisků pozic*, *zákrytů*. V závislosti na velikosti podmnožin, které jsou podruhé transformovány, dochází k úspoře místa i výpočetních prostředků.

Kapitola 3

Praktická část práce

Praktická část bakalářské práce se zaměří na aplikování vybraných principů uvedených v teoretické části práce. Aplikováním představených principů položíme základy pro systém, který zvládne provedení potřebných kroků, které před připraví textová data určená pro vložení nebo zpracuje data již obsažená v korpuse dat. Jako prostředek pro implementaci algoritmů byl zvolen skriptovací jazyk *python* (ve verzi 2.5) a programovací jazyk *C++*. Kromě samostatně vytvořené programové části bylo hojně využito i *GNU utilit*, které zprostředkovávají zpracování textových dat optimalizovanými nástroji. Mezi těmito nejčastěji využívanými nástroji jsou například *sed*, *grep*, *wc*, *cat*.

3.1 Algoritmy pro kontrolu podobnosti dokumentů

Při výběru metody pro značení a porovnání podobnosti dokumentů jsou hlavními kritérii vlastnosti použité metody jako rychlost, efektivita využití systémových prostředků. Dalším a neméně významným kritériem je i nezávislost transformačních funkcí v přípravné fázi, kdy je vyžadováno, aby *otiskování* jednoho dokumentu bylo nezávislé na *otiskování* dokumentu jiného. Tento přístup navíc dovoluje souběžný běh přípravné fáze na několika dokumentech současně, tedy paralelní zpracování. Výhodou je i ten fakt, že není potřeba žádný další prostor pro uchování slovníku s četnostmi slov a jiných slovních celků.

Na základě takto stanovených priorit, i při zachování jednoduchosti řešení, byly zvoleny pro implementaci právě dvě metody a to určování podobnosti odstavců pomocí *Rabin fingerprint* odstavců, *384 bitový vektor* připadající na dokument. Obě tyto metody byly primárně připraveny pro zpracování textu ve vertikální reprezentaci, což souvisí s dostupnými testovacími daty právě v tomto formátu. Korpus textů formátovaných ve vertikální reprezentaci je navíc doplněn identifikačními řádky, které jednoznačně identifikují specifický dokument. Formát korpusu

```
<doc id="autodesk/1995/04/3" lang="cs" type="sci1"
    title="CADKON pro AutoCAD LT" source="CD Modrých stránek"
    medium="cdrom" author="Robert Krňávek" published="1995-08"
    subtype="inf">
```

Spuštění skriptů, které provádí přípravnou fázi probíhá v podobě zřetězeného zpracování, kdy daty na standardním vstupu jsou textová data korpusu a na standardním výstupu jsou výsledná data, získaná z činnosti přípravné fáze. V závislosti na stanovených výkonnostních požadavcích nebylo využito M-Tree pro podobnostní vyhledávání. Metody, které budou dále v textu uvedeny byly dostačujícími i efektivními nástroji současně.

3.1.1 Rabin fingerprinty odstavců

Hlavním třídou, která řídí přípravnou fázi činnosti skriptu je `trida_duplicity_korpusu`, která pro vytváření *otisků pozic* a následně i *otisků* pro celý dokument využívá třídu `CTvorba_fingerprintu`. Před použitím `CTvorba_fingerprintu` třídy je nutné provést její inicializaci a to následujícím způsobem:

Jméno proměnné instance třídy `CTvorba_fingerprintu`	Popis významu
`inicializovano`	Při dávkovém zpracování většího množství dokumentů je možno provést počáteční nutnou inicializaci na základě vynuceného vykonání pouze před zpracováním prvního dokumentu
`uvazovat_odstavce`	Dokument bude před zpracováním rozdělen na odstavce, které budou uvažovány jako samostatné dílčí podjednotky
`ke_slouceni_fingerprintu`	Velikost množin *zákrytů*, které budou nepřekrývavě sloučeny
`sdruzovani_fingerprint_odstavce`	V případě slučování podmnožiny *zákrytů* (menší než celá množina *zákrytů*) odstavce do *super-zákrytu*, provede vypsání hodnot skupiny *super-zákrytů* v rámci daného odstavce
`minimalni_pocet_slov_odstavce`	Hranice počtu *pozic*, kdy bude odstavec označen za krátký, případně dlouhý

Tabulka 3.1: Inicializace tvorby otisku

Při přípravné fázi a zpracování dat ze standardního vstupu jsou načítány identifikace dokumentů a jejich textová data. Budeme uvažovat činnost popsanou podle parametrů specifikovaných výše. Způsob zpracování dat u této metody využívá specifika vertikální reprezentace textových dat a konkrétně dělení textu na odstavce. Textová data dokumentu jsou rozdělena na jednotlivé odstavce, které neobsahují žádné další formátovací značky, protože jejich výskyt je v dalším zpracování nepotřebný a dokonce i nežádoucí. Po provedení *tokenizace* je pro každou *pozici* vypočten její *zákryt* a následně pro celý odstavec je vytvořen jeho *super-zákryt*. Tento *super-zákryt* je vytvářen ze všech *zákrytů* odstavce. Množina *super-zákrytů* již není žádným dalším způsobem redukována a představuje výslednou podobu značení celého dokumentu. Formátovaný výstup identifikačních řádků a *otisků* dokumentu má následující podobu:

Výstup přípravné fáze

```
1 <doc id="autodesk/1995/04/3" lang="cs"...>
2 24D0C6FDFABD94FD.k 577D7B5C4E3CE15E.k 9C268CBE4CFE799C.d
```

Tento formátovaný výstup je zapsán do souboru, nad kterým je následně vykonána dále popsaná posloupnost zřetězených příkazů.

1. Čtení výstupu přípravné fáze, ve které jsou obsaženy jak řádky identifikační, tak i řádky s obsahem *otisků* odstavců. Činnost určení podobnosti dvou dokumentů není závislá na přítomnosti identifikačních řádků, proto se již ve zřetězeném zpracování finální fáze tyto řádky nenacházejí. Identifikační řádky jsou později pro dohledání příslušných dokumentů sestaveny do samostatného souboru, kdy číslo řádku takto

sestaveného dokumentu a pořadové identifikační číslo z finální fáze určení podobnosti sobě odpovídají 1 : 1. Pro již zmíněné výhody vertikální reprezentace dat převádíme *otisky* odstavců dokumentu do řádkové reprezentace. Na jednom řádku se tak nachází pořadové identifikační číslo dokumentu oddělené bílou mezerou a samotný *otisk* odstavce.

Sestava *otisků* dokumentů:

```
1  1  24D0C6FDFABD94FD.k
2  1  9C268CBE4CFE799C.d
3  2  577D7B5C4E3CE15E.k
4  2  98C92EEF8E2F4737.d
5  2  2A4002FFC05B087.d
```

Sestava identifikačních řádků dokumentů:

```
1  <doc id="autodesk/1995/04/3"  lang="cs" ...>
2  <doc id="techmag/1996/05/2"  lang="cs" ...>
```

2. Seznam vertikálně uspořádaných *otisků* seřadíme podle druhého sloupce, ve kterém se nachází *otisky* odstavců dokumentu. Tímto způsobem jsou odhaleny vzájemně se shodující dokumenty a to nejméně v jediném odstavci. Z tohoto seřazeného seznamu odstraníme řádky, ve kterých se nachází *otisk* odstavce pouze v jediném výskytu a nemá tedy dalšího kandidáta na vzájemné porovnání. Některé dokumenty obsahují jeden odstavec i násobně-krát, proto před výstupem tohoto kroku provedeme pročištění výsledků od duplicitních řádků. Tímto způsobem jsme schopni redukovat velké množství unikátních dokumentů, které v dalším zpracování již není třeba uvažovat a získáme tím pádem i zrychlení celého systému určování podobnosti dokumentů.

3. Jelikož se jediný odstavec může nacházet i ve více jak dvou dokumentech je potřeba sestavit soupis všech pořadových identifikačních čísel dokumentů, která daný odstavec obsahují. Výsledkem je tedy soupis množin dokumentů, které obsahují alespoň jeden společný odstavec. Binární utilita provádí rychlé určení podobnosti množiny dokumentů prostřednictvím pravidla každý s každým. Porovnání dvou dokumentů se řídí nastaveným spodním hraničním limitem (*threshold*). Jelikož je výsledná podobnost dvou dokumentů určená touto metodou v rozsahu *0..100*, která vyjadřuje procentuální podobnost dvou dokumentů, může těchto hodnot nabývat i hraniční limit. V případě splnění hranice podobnosti dvou dokumentů je tato dvojice identifikačních čísel dokumentů vypsána i s hodnotou podobnosti, ve které se tato dvojice dokumentů shodla. Výsledný soubor soupisu podobných dokumentů s určením procentuální míry podobnosti, ve které se shodují, vypíšeme na standardní výstup a to ve formátu:

Hraniční limit

```
1  10550  19501  95.1
```

Pořadové ID 1.dokumentu	Pořadové ID 2.dokumentu	Míra podobnosti dvojice dokumentů
10550	19501	95.1

Tabulka 3.2: Popis formátu výstupu metody Rabin fingerprint

3.1.2 384 bitový otisk na dokument

Součástí hlavní třídy `trida_duplicity_korpusu` je funkce `Vytvor_384_vektor`, která vytváří 384 bitový vektor na dokument. Po nastavení parametru, který určuje velikost výsledného vektoru, přípravná fáze vykonává načtení textových dat dokumentu (ve vertikální reprezentaci) s tím, že formátovací značky nejsou uvažovány. Pro každou *pozici* dokumentu je vytvořen její *otisk* a to takovým způsobem, že je k tomu využita *SHA-384* funkce, SHA-384
která produkuje 384 bitový *otisk* zadané *pozice*. Tím si zajistíme, že pro stejné *pozice* bude vždy stejný výsledek této hashovací funkce. Tato funkce produkuje *otisk* s velikostí 384 bitů. Očekávanými hodnotami dimenzí vektoru *pozice* jsou hodnoty $\{-1, 1\}$, proto převod vzniklého *otisku*, kdy bit s hodnotou 0 představuje hodnotu dimenze -1 a bit s hodnotou 1 představuje hodnotu dimenze 1. Následně tímto způsobem vygenerovaný vektor přiřadíme do výsledného vektoru dokumentu přičtením hodnoty dimenze vektoru k příslušné dimenzi výsledného vektoru. Po zpracování jedné pozice je pokračováno zpracováním další *pozice* dokumentu. Na konci, po zpracování všech *pozic* dokumentu je nutno provést ještě vyvážení výsledného vektoru z důvodu potřeby jeho elektronického uchování. Dimenze výsledného vektoru větší rovno 0 odpovídají 1, v opačném případě 0. Tímto způsobem vytvořený výsledný vektor je vypsán na standardní výstup a to v podobě:

```
1 <doc id="techmag/1996/04/1" lang="cs"...>
2 11111010001001001101100011010011011001100...000100110100110
```

Proces samotného určení podobnosti dokumentů je až na malé výjimky podobný procesu uvedeném v předchozí podkapitole, z toho důvodu budou popsány pouze odlišné části tohoto procesu s případným odvoláním na předchozí podkapitolu.

1. Nad 384 bitový vektorem, který je výsledkem označení dokumentu, je provedena konverze na posloupnost 12 32 bitových celých neznaménkových čísel. Konverze probíhá rozdělením 384 bitového vektoru na jednotlivé bitové dimenze, které jsou shlukovány do celků s velikostí 32 dimenzí. Nejenom, že tímto způsobem je možno využít běžné datové typy pro uchování proměnných, ale navíc umožňuje využít výhody eliminace unikátních (neduplicitních) dokumentů již v rané fázi činnosti procesu porovnání dokumentů. Další nespornou výhodou je velmi rychlé porovnání všech 12 celých čísel mezi dvěma dokumenty a dostupnost bitových operací mezi datovými typy celé číslo.

3. V poslední fázi binární utilita porovnává skupiny dokumentů, které se shodly alespoň na jednom 32 bitovém čísle vzniklém po rozdělení 384 bitového vektoru. Vždy jsou porovnávány sobě příslušné dimenze a výsledkem každého porovnání je exkluzivní disjunkce, která udává počet bitů, ve kterých se daný rozsah dimenzí liší. Tímto způsobem získáme celkový počet dimenzí, ve kterých se vektory dvou dokumentů liší. V závislosti na nastavené prahové hodnotě (*threshold*), která může nabývat hodnot v rozsahu *0..384*, je provedeno určení, které dokumenty mají býti považovány za podobné. Míra podobnosti odpovídá tomu, že čím vyšší hodnota, tím vyšší míra nepodobnosti dvou dokumentů. Absolutně shodné dokumenty jsou ty, jejichž hodnota podobnosti je určena jako 0, protože to znamená, že vektory obou dokumentů se v žádné dimenzi neliší. Tímto způsobem je určena podobnost mezi dokumenty. Míra podobnosti

   Výstupu obsahuje pořadová identifikační čísla obou dokumentů s určením počtu rozdílných hodnot dimenzí, ve kterých se oba dokumenty liší. Formát výstupu je následující:

```
1 58550 59501 2
```

Pořadové ID 1.dokumentu	Pořadové ID 2.dokumentu	Počet rozdílných hodnot dimenzí
58550	59501	2

Tabulka 3.3: Popis formátu výstupu metody 384 bit vektor

3.2 Další algoritmy a nástroje

Kromě podobnosti dokumentů existuje i řada dalších převážně statistických pohledů na kolekce dat, které slouží pro udržení jejich relevantnosti a sledování kvality. Pro tyto účely byly vytvořeny nástroje, které toto sledování aktivně provádí a poskytují tak základ pro další rozhodnutí o nutných modifikacích například z důvodu zachování unikátního obsahu.

Stahování obsahu webu

Součástí systému pro automatizovanou tvorbu korpusu je kromě kontroly stávajících a nových dat i modul pro stahování dat z internetu. Jako nejvhodnější kandidát pro tato data byl vybrán zpravodajský server pro svoje přednosti, kterými je také relativně častá aktualizace článků přiměřené délky. Následná volba seriózního zpravodajského serveru by měla zajistit požadavek na kvalitu jazykové stránky získávaných dat.

3.2.1 Podobnost v krátkých a dlouhých odstavcích

Pro zjištění podobnosti v dlouhých odstavcích a nepodobnosti v krátkých odstavcích nad kolekcí dat (dokumentů) byl vytvořen systém, který tyto skupiny dokumentů odhalí a zpětně o nich informuje. Za účelem získání informací o délce odstavce je potřeba pro potřeby tohoto nástroje provést označení dokumentů s rozšířenou informací o *otisku*. Touto rozšířenou informací *otisku* je určení délky odstavce ze kterého byl sestaven, rozlišovány jsou dvě kategorie odstavce krátký a dlouhý. Pro tyto potřeby použijeme přípravnou fázi metody *Rabin fingerprint* pro určení *otisků* odstavců s nastavením parametru instance třídy `CTvorba_fingerprintu`, `minimalni_pocet_slov_odstavce` na hodnotu 9. Nastavení této hodnoty definuje hranici počtu *pozic* mezi krátkým a dlouhým odstavcem.

Váhování otisků

Tímto způsobem značená kolekce dat bude vypadat následovně:

```
1 <doc id="autodesk/1995/04/3" lang="cs"...>
2 24D0C6FDFABD94FD.k 577D7B5C4E3CE15E.k 9C268CBE4CFE799C.d
3 ...
```

Vykonání této části systému určování podobností využívá soubor s množinami podobných dokumentů, které vznikly na základě činnosti nástroje pro určení podobnosti dokumentů metodou *Rabin fingerprint*. Dalším nutným souborem je i kolekce *otisků* všech dokumentů, rozšířená o informace o délce jednotlivých odstavců. Výstupem této binární utility je soupis dokumentů podobných v dlouhých odstavcích a současně nepodobných v krátkých odstavcích. Výpis je formátován následujícím způsobem:

```
1 1558262 1558543 100 4.16667
```

Pořadové ID 1.dokumentu	Pořadové ID 2.dokumentu	Podobnost v dlouhých odstavcích	Podobnost v krátkých odstavcích
1558262	1558543	100	4.16667

3.2.2 Web crawling zpravodajského serveru

Systém pro získávání obsahu zpravodajského serveru je plně automatizovaný schopný kdykoliv provádět svou činnost, kdy je na něj vznesen dotaz o stažení dat. Pravidelnost provádění kontroly na dostupnost nových dat s případným požadavkem na jejich stažení je řízena plánovačem úloh. Jako nejvhodnější pravidelnost spouštění se jeví hodinová četnost kontroly a stažení článků.

Automatizace činnosti

Činnost získávání dat je kompletně v režii skriptů jazyka python, kdy je využíváno především snadné práce s textovými daty a jednoduchost provádění operací nad nimi. V dalších odstavcích bude stručně popsán postup monitorování a získávání dat ze zpravodajského serveru. Tento systém bude pro jednoduchost v textu dále označován jako nástroj.

1. Na začátku vykonání činnosti skriptu je provedeno nastavení všech potřebných proměnných (konstant), které specifikují samotný zpravodajský server i proces získávání dat s jejich uložením. V krátkém výčtu budou uvedeny některé z nejdůležitějších proměnných s krátkým popisem jejich významu.

Jméno proměnné	Příklad hodnoty	Popis významu
web_page	http://servis.idnes.cz/rss.asp	URL adresa RSS čtečky, která provádí zjištění článků dostupných na serveru
KODOVANI_CESTINY	iso-8859-2	Kódování češtiny stažených a uložených článků
KODOVANI_WEBU	cp1250	Kódování češtiny použité při kódování článků na serveru
LOG_FILE	my_log.log	Název souboru pro záznam činnosti skriptu

2. Z důvodu primárního užití systému jako nástroje běžícího na pozadí po dlouhé časové období, spouštěného plánovačem úloh, je nezbytná přítomnost logovacího systému. Vytvoření logovacího souboru je nezbytnou součástí i jako zpětná vazba při testování správné činnosti nástroje.

3. Po aktivaci RSS čtečky, která zkontroluje všechny dostupné články a odkazy na ně, je provedeno rozdělení článků podle jejich kategorie zaměření například na sport, kulturu a zpravodajství z domova. Jelikož zvolený zpravodajský server využívá pro každou kategorii více či méně modifikovanou podobu šablony článků, je nutno ke každé kategorii přistupovat individuálně. Tento jev je nutno brát v úvahu a proto je pro získávání obsahu článků vytvořeno několik šablon, které získávání potřebných dat z článků provádí. V souvislosti s tímto jevem je potřebné i pravidelné sledování logu tohoto nástroje z důvodu možné změny všech nebo pouze jedné dílčí šablony, což může vést k nemožnosti nebo nesprávně získaným informacím z článků.

Šablony článků

4. Pokud pro daný článek existovala šablona pro získání dat a podařilo se získat potřebná data článku z odkazu předaného RSS čtečkou, je z titulku článku vytvořen *otisk Rabin fingerprint*, který tento článek jednoznačně specifikuje. Požadovanými daty, které musí být z článku získány jsou **datum vydání**, **text samotného článku** a **titulek článku**. Na základě takto získaných dat je sestaven název souboru, do kterého bude článek uložen, tento název souboru sestává z *otisku* titulku článku a data, kdy byl článek získán.

5. Provedením kontroly jestli se sestavený název souboru již na disku ve specifikované složce nachází, určíme zda-li článek máme už uložený. V případě, že se stejný název souboru na disku nachází, provedeme přečtení tohoto souboru za účelem získání informací o čase, kdy byl tento článek vydán. Pokud se časy aktuálně staženého a uloženého článku liší, jedná se v případě staženého článku o jeho aktualizovanou podobu. Aktualizovaný článek uložíme v nově stažené podobě. Pokud však název souboru na disku není nalezen, pak článek ještě není uložen a je pokračováno v dalším kroku.

Kontrola aktualizace článku

6. Formátování článku je realizováno ve vertikální podobě, která byla zvolena pro své výhody, které jsou zmíněny v tomto textu dříve. Jelikož systémové nároky celého nástroje nejsou příliš veliké, v průběhu stahování a ukládání dokumentů na pevný disk je prováděna i přípravná fáze určování podobnosti dokumentů. V této fázi jsou odstavce článku označeny metodou *Rabin fingeprint* a výsledek označení je uložen do souboru se stejným názvem jako článek, který se liší akorát v příponě tohoto souboru. Díky použití vertikální reprezentace textu je nutné doplnění identifikačního řádku dokumentu, který je sestaven na základě údajů získaných z RSS čtečky i samotného článku. Těmito údaji, které jsou zapsány do souboru společně s článkem jsou:

Otisk článku

Jméno proměnné	Popis významu
Jednoznačná identifikace článku	Jméno serveru, datum, čas vydání, u aktualizovaného článku i datum a čas aktualizace článku
Jazyk článku	Určen předvolbou pro použitý zpravodajský server
Titulek článku	Nadpis článku získaný z RSS čtečky
Zdroj	Internet, navíc je doplněna i URL adresa článku
Autoři článku	Informace o autorství článku, kterým je doplněn zpravidla každý článek

Kapitola 4

Výsledek implementace

Při testování algoritmů pro určení podobností nad dokumenty byla využita kolekce dat, která je charakterizována následujícími parametry:

Název kategorie	Hodnota kategorie
Počet dokumentů	1 560 509
Počet odstavců	4 064 931
Počet slov	690 093 678
Počet znaků (bez mezer)	3 169 346 842
Počet znaků(s mezerami)	3 784 068 206
Počet značek	89 807 959

Tabulka 4.1: Statistika testovacího korpusu

Takto rozsáhlá kolekce poskytla dostatečně významný vzorek testovacích dat, aby naměřené hodnoty představovaly reprezentativní výsledky.

4.1 Časové náročnosti metod určování podobností

Pro obě použité metody určení podobnosti dokumentů je společné, že se skládají ze dvou významných fází a to přípravné fáze a fáze finálního porovnávání dokumentů (této fázi předchází minimalizace počtu porovnání). Každá z těchto fází má svůj nepostradatelný význam a při použití uvedeného rozsahu dat dosahuje nezanedbatelné časové náročnosti. Názornosti časových náročností poslouží rozdělení do jednotlivých fází s vyjádřením příslušné doby výpočtu. Nastavení parametrů při testování systému pro automatizované určování podobnosti dokumentů bylo pro jednotlivé metody zvoleno podle hodnot, které jsou uvedeny v tabulce 4.2. Podobnost dokumentů na základě metody *Rabin fingerprint* byla nastavena na hodnotu 80 a vyšší. Pro metodu vytvářející 384 bitový vektor byl počet dimenzí, ve kterých se mohou dva dokumenty maximálně lišit, nastaven na hodnotu 5 a nižší. Tyto hodnoty byly zvoleny na základě úvahy a série provedených měření i kontroly výsledků, kdy právě tato nastavení nejlépe splňovala požadavky na přesnost a rychlost zpracování.

Nastavené parametry

Přípravná a před přípravná fáze, kde probíhá označení jednotlivých dokumentů a provedení přípravy dat pro vzájemné porovnání mezi dokumenty. Fáze finální, porovnání dokumentů (hlavní činnost), která již provádí samotné porovnání mezi jednotlivými dokumenty a jejímž výsledkem jsou určené páry podobných dokumentů. Časové náročnosti těchto uvedených fází jsou shrnuty v tabulce 4.3.

Jméno proměnné instance třídy `CTvorba_fingerprintu`	Hodnota proměnné
`inicializovano`	True
`uvazovat_odstavce`	True
`ke_slouceni_fingerprintu`	-1
`sdruzovani_fingerprint_odstavce`	True
`minimalni_pocet_slov_odstavce`	9

Tabulka 4.2: Inicializace metody Rabin fingerprint

Fáze činnosti	Název metody	Časová náročnost
Před přípravná a přípravná	Rabin fingeprint	5,3h
Před přípravná a přípravná	384 bit vector	19,4h
Finální	Rabin fingeprint	11,0h
Finální	384 bit vector	0,1h

Tabulka 4.3: Časové náročnosti fází

Na základě vykonaných testů, v závislosti na nastavených parametrech programů a skriptů, vznikly dvě množiny dvojic podobných dokumentů. Množství nalezených podobných dvojic dokumentů oběma metodami je uvedeno v tabulce 4.4. K názornosti slouží i hodnota vzájemného průniku množin podobných dokumentů, které jsou výsledkem určení obou metod.

Podobnosti korpusu

Rabin fingerprint	384 bit vektor	Průnik obou množin
52 564	582 376	50 187

Tabulka 4.4: Množství nalezených duplicit korpusu dat

4.2 Časová náročnost dalších algoritmů

Shrnutí testů provedených nad dalšími nástroji, které představují základní prostředky pro automatizovaný systém spravující korpus dat.

4.2.1 Podobnost krátkých a dlouhých odstavců

Algoritmus, který se zaměřuje na určení podobnosti dokumentů s využitím váhování odstavců (podle počtu *pozic* odstavce), je určení podobnosti dokumentů v dlouhých a zároveň i nepodobnosti dokumentů v krátkých odstavcích. Výpočetně je využita část zpracování provedená metodou *Rabin fingerprint*, konkrétně se jedná o soupis vzájemně podobných dokumentů alespoň v jednom odstavci. Jako vhodná hranice pro hodnotu podobnosti dlouhých odstavců, byla zvolena hodnota 80. Stejná hodnota byla zvolena i pro podobnost krátkých odstavců, ale tato hodnota již není brána jako minimální podmínkou pro splnění kritéria, ale jako maximální hodnotu podobnosti mezi krátkými odstavci.

Výsledkem zpracování je množina dvojic dokumentů splňujících vlastnosti podobnosti v dlouhých odstavcích a nepodobnosti v krátkých odstavcích. Celkový výpočetní čas spotřebovaný tímto nástrojem pro zpracování testovacího korpusu dat je **4,3h**, součástí tohoto

Podobnost váhování

intervalu však není čas potřebný pro před přípravnou a přípravnou fázi metody *Rabin fingerprint*, jejíž výsledky jsou využívány. Množství dokumentů, které byly tímto nástrojem v testovacím korpuse dat označeny za podobné v dlouhých odstavcích a nepodobné v krátkých odstavcích je **25 319**.

4.2.2 Web crawling zpravodajského serveru

Rychlost činnosti nástroje pro stahování obsahu zpravodajského serveru je nekritická, ale i přesto budou uvedeny doby potřebné pro načtení článků z RSS čtečky a jejich uložení na pevný disk. Testování bylo zahájeno o víkendovém dni (*Víkend 00:00*), všechny další spuštění již tedy v sobě zahrnují i čas potřebný pro kontrolu aktualizací článků a uložení článků nových. Tyto časové náročnosti jsou vedeny v tabulce 4.5.

Počáteční čas	Doba trvání	Množství získaných článků
Víkend 00:00	0m7.614s	28
Víkend +03:00	0m7.597s	20
Víkend +17:00	0m7.710s	25
Víkend +22:00	0m8.961s	20
Víkend +24:00	0m11.992s	15
Pr. den 00:00	0m9.963s	29
Pr. den +03:00	0m9.622s	10
Pr. den +17:00	0m10.280s	27
Pr. den +22:00	0m11.665s	27
Pr. den +24:00	0m14.740s	15
Celkem	1m40,144s	216

Tabulka 4.5: Časová náročnost, web crawling

Součástí zkoumání stahování obsahu zpravodajského serveru je i zjištění množství a přesnější popis dat, které je schopen tento nástroj za vyhrazený časový úsek získat. Toto množství dat může potenciálně dále posloužit k rozšíření korpus dat. Tabulka 4.6 vyjadřuje množinu článků získanou během testování časových náročností. Význam nástroje

Kategorie článků	Množství (%)
Zpravodajský	131 (60,6%)
Sportovní	77 (35,6%)
Technický	7 (3,2%)
Mobilní	1 (0,5%)

Tabulka 4.6: Získaná data, web crawling

Zajímavým faktem jsou statistiky, uvedené v tabulce 4.7, nad těmito získanými články, které reprezentují významnost této metody získávání dat.

Název kategorie	Hodnota kategorie
Počet dokumentů	216
Počet odstavců	6 395
Počet slov	135 453
Počet alfanumerických znaků	573 305
Počet znaků (bez mezer)	576 380
Počet znaků (s mezerami)	666 490
Počet značek	51 954

Tabulka 4.7: Statistika sestaveného korpusu

Kapitola 5

Závěr

Výsledkem práce bylo seznámení se s existujícími metodami formátování a značení textových korpusů. Po získání potřebných znalostí se pozornost zaměřila na nástroje, které umožní efektivní zpracování textových dat s následným sestavením korpusu. S tím spojené určování podobnosti dokumentů i jejich rozšířené varianty s váhováním odstavců. Z představených metod byly vybrány takové, které na základě zvážení a zhodnocení jejich parametrů byly uznány za nejlepší vhodné, s přihlédnutím k jejich zpracování i samotné implementaci. Tyto metody splňovaly požadavky na použití v rozsáhlých kolekcích dat i nad daty, která se v pravidelných intervalech rozšiřují a kontrola zachování kvality je tak kritickým parametrem.

Programové řešení představuje plně použitelné nástroje automatizovaného systému, jejichž úspěšnost byla v této technické zprávě prezentována. Navíc architektura navrženého systému umožňuje jeho snadnou rozšířitelnost a vylepšení dílčích částí. Fáze zpracování vstupních dat, fáze přípravná i fáze finální jsou schopny v rámci samy sebe býti nezávislé, což dovoluje možnost jejich paralelního zpracování a tím i škálování výkonu celého systému.

5.1 Cíle pro další práci

Mezi úlohy pro další práci patří snížení úrovně rozlišení při určování podobností z dokumentů a odstavců na menší větné celky. Těmi jsou například souvětí nebo i jednotlivé věty. Takový systém dovolí přesnější odhalení podobností mezi dokumenty. Společně s tímto cílem souvisí i požadavek na snížení výpočetních nároků a zavedení systému statistické eliminace *pozic*, které jsou brány v úvahu při zpracování a značení dokumentů. Případným objektem ke zkoumání jsou i nové metody značení *pozic* a větných celků.

Literatura

[1] Bayardo, R. J.; Ma, Y.; Srikant, R.: Scaling up all pairs similarity search. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-654-7, s. 131–140, doi:http://doi.acm.org/10.1145/1242572.1242591.

[2] Buchmann, J. A.: *Introduction to Cryptography*. Springer, 2000, ISBN 0-387-95034-6.

[3] Daciuk, J.: *Incremental Construction of Finite-State Automata and Transducers, and their Use in the Natural Language Processing*. Dizertační práce, Technical University of Gdańsk, 1998.

[4] Černý, S.: Dokumentace knihovny LIBMA. Fakulta Informačních Technologií, Vysoké Učení Technické v Brně, 2008.

[5] Henzinger, M.: Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA: ACM, 2006, ISBN 1-59593-369-7, s. 284–291, doi:http://doi.acm.org/10.1145/1148170.1148222.

[6] Manku, G. S.; Jain, A.; Das Sarma, A.: Detecting near-duplicates for web crawling. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-654-7, s. 141–150, doi:http://doi.acm.org/10.1145/1242572.1242592.

[7] Poljak, P.: *Testování výkonnosti indexové struktury M-tree: bakalářská práce*. Masarykova univerzita v Brně, Fakulta informatiky, 2008.

[8] Pugh, W.: Detecting duplicate and near-duplicate files. United States Patent No. 6,658,423, December 2003, `http://www.cs.umd.edu/ pugh/google/Duplicates.pdf`.

[9] Rychlý, P.: *Korpusové manažery a jejich efektivní implementace*. Dizertační práce, Masarykova univerzita v Brně, Fakulta informatiky, 2000.

[10] Sedláček, R.: *Morfologický analyzátor češtiny*. Diplomová práce, Masarykova univerzita v Brně, Fakulta informatiky, 1999.

[11] Šulc, T.: ThinkPad W700ds - profesionál se dvěma displeji [online]. `http://pctuning.tyden.cz/thinkpad-w700ds-profesional-se-dvema-displeji`, 2009.

[12] Yang, H.; Callan, J.: Near-duplicate detection for eRulemaking. In *dg.o 2005: Proceedings of the 2005 national conference on Digital government research*, Digital Government Society of North America, 2005, s. 78–86.

[13] Zezula, P.; Amato, G.; Dohnal, V.; aj.: *Similarity Search, The Metric Space Approach*. Springer, 2006, ISBN 0-387-29146-6.

[14] Zhou, T.: Rabin's Fingerprinting. `http://discovery.csc.ncsu.edu/~aliu3/reading_group/Rabin%CC%81%20Fingerprinting.ppt`.

Příloha A

Vertikálně reprezentovaný text

```
1  <doc id="mf/2008/8/4/10/43" lang="cs" source="http://tinyurl.com/pshs5p"
       title="Nepoučitelný~řidič~si~musí~alkohol~za~volantem~odpracovat"
       author="MF~DNES/taj">
2  <p>
3  Hodonínští
4  policisté
5  dopadli
6  nepoučitelného
7  řidiče
8  <g/>
9  ,
10 který
11 si
12 dal
13 několikrát
14 "
15 <g/>
16 na
17 kuráž
18 <g/>
19 "
20 a
21 pak
22 vyrazil
23 na
24 jihomoravské
25 silnice
26 <g/>
27 .
28 Pokaždé
29 mu
30 naměřili
31 přes
32 jednu
33 promile
34 <g/>
35 .
36 </p>
37 </doc>
```