SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE

Fakulta informatiky a informačných technológií

Ivan Gulis

Útoky na detekcie plagiátorstva

Bakalárska práca

Študijný program: Informatika
Študijný odbor: 9.2.1 Informatika
Miesto vypracovania: Ústav informatiky a softvérového inžinierstva, FIIT STU, Bratislava
Vedúci práce: doc. Mgr. Daniela Chudá, PhD.
december 2015

ANOTÁCIA

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Študijný program: Informatika

Autor: Ivan Gulis

Bakalárska práca: Útoky na detekcie plagiátorstva

Vedúci bakalárskej práce: doc. Mgr. Daniela Chudá, PhD.

december, 2015

Cieľom tejto práce je analyzovať súčasný stav metód detekcie plagiátov v textoch a navrhnúť mechanizmy pre vylepšenie týchto metód proti existujúcim útokom. Metódy detekcie a útoky budú rozdelené do kategórii podľa princípov, na ktorých sú založené, kde budú opísané ich slabé a silné stránky. Po analýze bude nasledovať experiment so súčasnými aplikáciami pre detekciu plagiátov, kde budú všetky aplikácie a ich metódy detekcie porovnané navzájom, spolu s percentuálnou úspešnosťou odhalenia plagiátov. Ako testovacie vzorky budú použité dvojice textov, originál a jeho plagiát - na ktorom budú postupne aplikované techniky jednotlivých útokov. Okrem tabuliek bude vytvorený aj graf s priemernými hodnotami úspešnosti všetkých aplikácii. Po zhodnotení experimentu budú navrhnuté nové mechanizmy pre vylepšenie detekčných metód proti úspešným typom útokov z experimentu. Tieto mechanizmy budú následne implementované a otestované na vzorkách slovenských textov. Výsledky oboch experimentov budú navzájom porovnané.

ANNOTATION

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Informatics

Author: Ivan Gulis

Bachelor Thesis: Attacks on plagiarism detection

Supervisor: doc. Mgr. Daniela Chudá, PhD.

2015, December

The purpose of this work is to analyze the state of current detection methods of plagiarism in texts and to design mechanisms for improvement of these methods against existing attacks. Methods of detection and methods of attacks will be divided according to principes, on which they are based. Their weak and strong aspects will be described there. Experiment with current applications for plagiarism detection will follow up after the process of analysis. All of the applications and methods of detection will be compared along with percentual success of plagiarism detection. Pairs of texts (original and it's copy, on which one of plagiarism methods is applied) will be used as testing samples. These texts will undergo applied technics of various attacks. Besides the table a graph with average values of all application success will be designed. New mechanisms for enhancing the detection methods against successful types of attacks will be designed. Afterwards these mechanisms will be implemented and tested on samples of slovak texts. The results of both experiments will be compared.

Obsah

1. Úvod

V súčasnosti je plagiátorstvo celkom bežná záležitosť. Výskyt plagiátov je hlavne v študentských prácach, kde študenti radšej odovzdajú prácu niekoho iného s vedomím, že ich určite nikto neodhalí. Študenti sú vynaliezaví a vymýšľajú rôzne metódy, ako zamaskovať svoj plagiát. Podobné je to s podvádzaním pri testoch - možnosť dosiahnuť lepšie hodnotenie za minimum námahy je lákavé a riziko nie je dostatočne veľké. S technikou narastá počet rôznych metód tvorby plagiátov, rovnako ako narastá počet metód tvorby "ťahákov". Plagiátori už nezostávajú u jednoduchého ctrl+c, ctrl+v, ale používajú obrázky či neznáme symboly, aby ich dokument vyzeral rovnako, no zvnútra bol úplne odlišný. Rozšírené je hlavne kopírovanie častí článkov z internetu, a následné upravenie týchto častí ako len najviac je to možné. Z dôvodu absencie slovenských vedeckých článkov na internete študenti prekladajú a kopírujú aj anglické články, čo podstatne komplikuje detekciu plagiátov - každý preklad je určitým spôsobom iný. Ak viacero študentov kopíruje z rovnakého zdroja na internete, je možný výskytu plagiátov aj bez priameho kopírovania študenta od študenta. Ako útok označujeme pokus skopírovať určité časti iného dokumentu a prehlásiť dokument za svoj vlastný, bez príslušných citácii alebo uvedenia zdroja. Ochrana pred útokmi je odhalenie zhody dvoch dokumentov, ktoré obsahujú rovnakú myšlienku alebo zhodné časti textu, ale sú prezentované ako dva odlišné dokumenty odlišných autorov. Nikde však nie je definované, koľko percentná zhoda musí nastať, aby bol dokument označený ako plagiát - tento aspekt zostáva na posúdení človekom, program len zobrazí podozrivé zhody.

1.1 Hlavný cieľ tejto práce

Hlavným cieľom tejto práce je analyzovať jednotlivé metódy plagiátorských útokov, zistiť ich slabiny, a navrhnúť techniky na ochranu pred nimi. Útoky aj metódy detekcie kategorizujeme do skupín. Po analýze silných a slabých stránok detekčných metód vykonáme experiment s tromi známymi nástrojmi na detekciu plagiátov, kde sa zameriame iba na aplikácie, ktoré porovnávajú vstupnú množinu dokumentov - nehľadajú plagiáty na internete. Výsledky experimentu zapíšeme do tabuliek a grafu, a úspešnosť útokov a metód ochrany navzájom porovnáme v zhodnotení experimentu. Nakoniec uvedieme stručný návrh mechanizmov, ktoré neskôr implementujeme a preveríme vzorkou slovenských textov - nebudeme sa zaoberať plagiátmi z angličtiny do slovenčiny. Výsledky experimentov porovnáme.

2. Metódy útokov na detekciu podobnosti textov

Tieto metódy útokov rozdelíme na tri skupiny:

1. pôsobiace na úrovni slov
2. pôsobiace na úrovni znakov
3. pôsobiace na úrovni vzhľadu dokumentu ako celku

2.1 Pôsobiace na úrovni vzhľadu dokumentu ako celku

Tieto techniky sa snažia upraviť dokument tak, aby zachovali vzhľad dokumentu podľa originálu.

Útok pridaním úvodzoviek pred a za text sa zakladá na myšlienke, že detektor plagiátov označí tento úsek za citáciu. Teoreticky by potom bolo možné dať úvodzovky na začiatok a koniec dokumentu, a program na detekciu plagiátov by text vôbec nekontroloval.

Útok výmenou medzier za biele znaky je veľmi silný typ útoku. Vizuálnou slabinou je šírka medzery a šírka znakov. Podstatne odhaliteľnejšou slabinou útoku je vznik podozrivo dlhých slov (alebo celý dokument sa stane jedným dlhým slovom). Existujú dve variácie: Náhrada medzier za znaky, ktoré sa v dokumente bežne nachádzajú (písmená, čísla). Pokusom je možné zistiť, aké znaky sú vhodné pre tento typ útoku (väčšinou úzke znaky ako napríklad “l,1,j”). Slabinou útoku je, že vznikne abnormálny výskyt použitého znaku. Je možné použiť náhodný biely znak na úkor vizuálnej stránky (medzery nebudú rovnako široké).
Druhá možnosť je použiť symboly mimo bežne používanej abecedy (znaky cyriliky alebo rôzne čínske znaky). Takto nevznikne nadbytok použitia určitého písmena. Tento útok je veľmi ľahko vykonateľný pomocou funkcii v napríklad Microsoft Worde (vyhľadať všetky medzery a nahradiť ich požadovaným znakom).
Príklad plagiátu: “Žuvanie je mechanické rozmelňovanie potravy alebo tabaku v ústnej dutine.” Plagiát:
“Žuvanie je mechanické rozmelňovanie potravy alebo tabaku v ústnej dutine.”
(Žuvanieijeimechanickéirozmelňovanieipotravyialeboitabakuiviústnejidutine.)

Útok využitím textovej a vizuálnej vrstvy PDF formátu je založený na fakte, že PDF sú tvorené vo vrstvách. Na pozadí je vrstva textová, a na povrchu je vrstva

vizuálna. Základom útoku je do vizuálnej vrstvy vložiť originálny text, a do textovej vrstvy vložiť náhodné znaky (“odpad”). Vylepšenie tejto metódy je použiť originálny text, a jeho znaky len premeniť na iné, takto sa zachová dĺžka dokumentu. Tento plagiát je extrémne dôveryhodný. Text sa dá myšou označiť, čo vyzerá ako označenie originálneho textu, ale kopírovanie kopíruje textovú vrstvu.
Postup tvorby plagiátu:

1. Strany pôvodného dokumentu sa konvertujú na obrázky (napríklad JPG). Tieto obrázky by mali byť čo najkvalitnejšie - budú tvoriť vzhľad strán dokumentu.
2. Vytvorí sa text rovnakej dĺžky s náhodnými znakmi (“odpad”).
3. Z vytvoreného textu sa spraví PDF dokument.
4. Obrázky originálneho textu sa nakopírujú cez strany textu.

Slabinou môže byť horšia kvalita vizuálnej vrstvy (je nutné text konvertovať na obrázok), a tiež samotná veľkosť plagiátu .
Menším pokusom sme zistili, že jedna strana textu v PDF má približne 128 kB. Ak použijeme aj vizuálnu aj textovú vrstvu, celý plagiát má až 381kB, čo je približne 3x viac ako originálny dokument. Čísla závisia od kvality obrázku - čím kvalitnejšie obrázky, tým väčší dokument.

Útok využitím obrázkov miesto častí textu má podobne ako predošlý typ slabinu v kvalite obrázkov. Je možné použiť okrem PDF formátu aj DOC.
Ak použijeme iba spomenutú vizuálnu vrstvu PDF (bez odpadu - nebude možné text označiť), plagiát má približne dvojnásobok pôvodnej strany textu.
DOC pri 1 strane toho istého textu má približne 5-násobné zväčšenie veľkosti.
Hodnoty podrobnejšie preveríme väčšími experimentmi, čísla uvádzame len pre názornú ukážku.

3. Metódy detekcie plagiátorstva v textoch

Kategorizácia:

1. Metódy založené na porovnávaní N-prvkových blokov
    a) N-gramy
2. Metódy založené na dynamickom programovaní
    a) Greedy String Tiling
    b) LCS - substring (Longest Common Substring)
    c) LCS - subsequence (Longest Common Subsequence)
    d) Levenstheinova vzdialenosť
3. Metódy založené na frekvencii použitia slov
    a) TF-IDF (term frequency, inverse document frequency)
    b) Cosine similarity (Weirdness) - Kosínová podobnosť
4. Metódy založené na netextovom porovnávaní
    c) Metadáta

3.1 Metódy detekcie založené na porovnávaní N-prvkových blokov

3.1.1 N-gramy

Analýza: Základom je rozdelenie dokumentov na skupiny po N slov (N-gramy), ktoré sa následne porovnávajú s ostatnými skupinami v druhom dokumente. Bližšie o N-gramoch sa dozviete v článku [1], kde je aj experimentom ukázaná ich funkčnosť a efektivita. Pre čo najväčšie zmenšenie veľkosti týchto N-gramov je vhodné využiť hašovacie funkcie. Ako príklad uvedieme hašovaciu funkciu - súčet všetkých ASCII hodnôt spoluhlások. Tieto haš hodnoty sa následne porovnávajú s haš hodnotami druhého dokumentu. Na toto sa využíva Rabin-Karp algoritmus, ktorý v pomerne dobrom čase dokáže porovnávať viac haš hodnôt naraz. Pre veľký počet haš hodnôt (a počet N-gramov) sa môže použiť winnowing algoritmus - zahašované N-gramy sa rozdelia do prekrývajúcich okien po 4 hodnoty (okno sa posúva vždy o jeden haš ďalej, aby vytvoril nasledujúce okno), a následne sa vyberú neprekrývajúce minimá haš hodnôt. Rabin-Karpom sa porovnávajú už len tieto pretriedené hodnoty - zníži sa presnosť metódy, ale podstatne sa zvýši rýchlosť. Viac o týchto technikách sa dozviete v článku [2], kde je opísané využitie Rabin-Karp algoritmu a samotný Winnowing algoritmus.

Klady a zápory: Haš hodnota N-gramu sa nezmení ani pri prehodení slov v rámci N-gramu. Pri použití Winnowings algoritmu sa stráca presnosť - je možné, že kopírované časti nebudú vybrané ako minimá. Táto metóda všeobecne nedáva dostatočne dobré výsledky (viz. tabuľky 4.1 a 4.2), pretože stačí zmena jediného písmenka (napr. na nejaký homoglyf), a haš hodnota celého N-gramu sa dramaticky zmení.

3.2 Metódy detekcie založené na dynamickom programovaní

3.2.1 Greedy String Tiling

Analýza: Táto metóda je založená na hľadaní vždy najdlhšieho spoločného podreťazca dvoch dokumentov. Vstupným parametrom je minimálna dĺžka spoločného podreťazca. Algoritmus nájde najdlhší podreťazec, označí ho za nájdený v oboch dokumentoch, a hľadá ďalší neoznačený najdlhší spoločný podreťazec. Ak už v dokumentoch neexistujú spoločné podreťazce dlhšie ako minimálna dĺžka, algoritmus končí. Výsledná podobnosť je potom pomer dĺžky spoločných podreťazcov * 2, a dĺžky oboch dokumentov spolu. Viac o Greedy String Tiling sa dočítate v článku [3], kde je opísané fungovanie programu JPlag, ktorý je na Greedy String Tiling metóde založený.

Klady a zápory: Táto metóda dáva všeobecne veľmi dobré výsledky (viz. tabuľka 4.3 nižšie v experimente). Algoritmus je odolný voči zmene poradia viet či odsekov (ak sú väčšie ako minimálna dĺžka spoločného stringu). Je závislá na vhodnom určení minima - pri výmene dvoch slov by minimum muselo byť 1, čo by znamenalo spustiť hľadanie najdlhšieho podreťazca až pre každé jedno slovo dokumentu. Algoritmu tiež nezáleží na pozícii zhodných podreťazcov - pri útokoch s homoglyfami hľadá podreťazce medzi upravenými slovami.

3.2.2 LCS - substring

Analýza: Veľmi podobná metóda ako Greedy String Tiling, ale jej úlohou je nájsť len jeden najdlhší reťazec. Algoritmus je založený na dynamickom programovaní a porovnávajú sa celé dokumenty ako dva dlhé podreťazce. Môže byť upravený na nájdenie N najdlhších podreťazcov so vstupným parametrom N - najdlhší spoločný podreťazec sa rovnako ako v Greedy String Tiling označí za nájdený, a hľadanie sa opakuje N-krát.

Klady a zápory: Pre správne fungovanie algoritmu je nutné v predspracovaní odstrániť hlavičky a časti dokumentu, ktoré sa opakujú vo veľkom množstve dokumentov (napr. názov fakulty alebo študovaný predmet). Mohlo by sa totiž stať, že tento reťazec bude označený za najdlhší spoločný podreťazec, a tak budú všetky porovnávané dokumenty podozrivé z kopírovania. Tiež je teda vhodné určiť vstupný parameter N>1, pretože najdlhší spoločný podreťazec môže tvoriť len malé percento z celého dokumentu - napríklad pri útokoch homoglyfami, synonymami alebo zmenou slovosledu vzniknú len menšie zhodné podreťazce. Tento algoritmus je vhodný na odhalenie veľkého zhodného úseku (alebo N úsekov) v rýchlom čase.
Všeobecne dáva táto metóda len veľmi slabé výsledky (viz. tabuľka 4.1).

3.2.3 LCS - subsequence

Analýza: Podobne ako predošlá LCS metóda je aj hľadanie najdlhšej podpostupnosti založená na dynamickom programovaní. Tiež je vhodné rozdeliť dokument na slová - človek myslí v slovách, prípadne ich zahašovanie - zníži sa tým veľkosť porovnávanej množiny, a ďalej sa hľadá podpostupnosť hašov. Oproti hľadaniu najdlhšieho substringu, subsequencia nemusí nasledovať nutne za sebou. Viac o tejto metóde sa dočítate v článku [4], ktorý sa celý zaoberá touto metódou aj s jej implementáciou.

Klady a zápory: Tento algoritmus vracia veľmi dobré výsledky (viz. tabuľka 4.1). Je v určitej miere odolný voči lokálnej zmene slovosledu, pretože podpostupnosť sa v širšom hľadisku nikdy úplne nestratí. Je vo veľkej miere odolný voči útokom, ktoré menia konkrétne slová - dokáže tieto slová jednoducho ignorovať (útoky zámenou písmen, zámena čísloviek za slovné ekvivalenty).

3.2.4 Levenstheinova vzdialenosť

Analýza: Touto metódou sa dá jednoducho číselne vyjadriť odlišnosť dvoch porovnávaných dokumentov. Vzdialenosť sa meria počtom operácii, ktoré treba vykonať, aby sa z reťazca A stal reťazec B. Tento algoritmus využíva opäť dynamické programovanie, a rozpoznáva 3 operácie: zmena, odobranie a pridanie znaku. Pre identické dokumenty je vzdialenosť = 1, pre úplne odlišné to je dĺžka dlhšieho dokumentu. Viac o tejto metóde sa dočítate v článku [5], kde je opísaný celý algoritmus aj s jeho implementáciou, a jeho kombinácia s Smith-Watermanovým algoritmom na efektívne a rýchle odhalenie plagiátov.

Klady a zápory: Na rozdiel od ostatných metód, výsledkom tejto metódy je jedno číslo - vzdialenosť dvoch dokumentov. Táto metóda je odolná voči útokom s použitím homoglyfov - zmenený znak len pridá vzdialenosť +1 o operáciu zmenu písmena, čo je oproti ostatným metódam výhodou. Naopak tento algoritmus zmätie prehodenie poradia dvoch podobných častí - zvýši sa vzdialenosť (napr. pri útoku zmenou slovosledu, alebo len výmena odsekov).

4. Experimenty s nástrojmi na detekciu

4.1 Vybrané nástroje k testovaniu

PlaDes [6] je to desktopová applikácia na detekciu plagiátov v textoch, vytvorená študentami Fakulty informatiky a informačných technológii. Oplýva všetkými vyššie analyzovanými metódami detekcie okrem Levensteinovej vzdialenosti. Podporuje všetky bežné textové formáty ako doc/docx, .pdf, .txt. Pre predspracovanie textu využíva odstránenie čísel, odstránenie stop slov (slov, ktoré nemajú samy o sebe žiaden význam - a, aby, aj, ale…), nahradenie synoným, lemmatizáciu (nájdenie koreňa slova) a stemming (odstraňovanie prípon). O niektorých vyššie uvedených metódach, ako aj o dôležitosti pre zefektívnenie detekčných metód sa dočítate v článku [7], kde nájdete aj experimentálne overené hypotézy. V aplikácii je možné prepnúť do módu anglického jazyka.

Ferret [8] je aplikácia vytvorená skupinou na University of Herfordshire. Obsahuje len metódu detekcie 3-gram, a podporuje formáty doc/docx, .pdf, .txt. Vie hľadať plagiáty aj v kódoch vo formátoch cpp, c, h, java. Text žiadnym spôsobom nepredspracúva, len ho konvertuje na .txt.

WCopyFind [9] je open-source aplikácia pre Windows bez potreby inštalácie. Texty kontroluje porovnávaním podreťazcov s množstvom nastavení. Metóda v aplikácii uvedená nie je, ale podľa pokusov by to mala byť Greedy-String-Tiling - dáva veľmi podobné výsledky ako GST metóda PlaDesu (viz. tabuľky 4.1 a 4.3). Je teda možné určiť dĺžku najkratšieho podretazca a minimálne % zhody pre report. Obsahuje tiež možnosť použiť rôzne ignorovanie či preskakovanie častí textu: ignorovanie diakritiky či interpunkčných znamienok, ignorovanie čísel, nerozlišovanie veľkých a malých písmen, preskakovanie neplatných slov a slov dlhších ako N znakov. Pre DOC formát aj nastavenie pre zohľadňovanie len základných znakov. Tiež dokáže nastaviť kontrolu v slovenskom jazyku - je možné pri kontrole vybrať z širokého výberu jazykov.

4.2 Výsledky experimentu

Plades 4.0.0 (%)	3-gram	LCS-substring	LCS-subsequence	Greedy-string-tilling	Cosine similarity	Priemer
presná kópia	100	100	100	100	100	100
uvozovky pred a za	100	100	100	100	100	100
synonymá	18,33	12,9	67,74	31,75	31	32,344
výmena čísiel za text	39,2	1,78	99,84	76,38	92	61,84
zmena slovosledu	16,67	25	60	23,81	71	39,296
platné homoglyfy	14,63	1,46	55,75	27,51	9	21,67
homoglyfy - cyrilika	3,32	0,52	16,15	0,51	2	4,5
biele znaky za medzery	0	0	0	0	0	0
využitie vrstiev PDF formátu	0	0	0	0	0	0
obrázky miesto textu	0	0	0	0	0	0

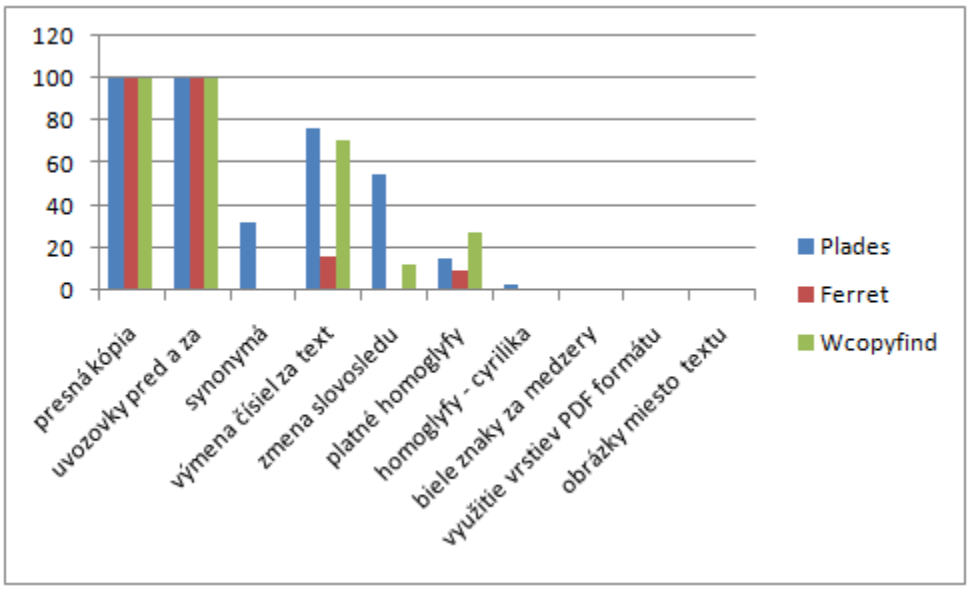
Tabuľka 4.1.Výsledky PlaDesu

Ferret 5.0 (%)	3-gram
presná kópia	100
uvozovky pred a za	100
synonymá	0
výmena čísiel za text	15,67
zmena slovosledu	0
platné homoglyfy	9,28
homoglyfy - cyrilika	-
biele znaky za medzery	-
využitie vrstiev PDF formátu	0,39
obrázky miesto textu	-

Tabuľka 4.2.Výsledky Ferretu

WCopyFind 4.1.4 (%)	Greedy-string-tilling
presná kópia	100
uvozovky pred a za	100
synonymá	0
výmena čísiel za text	71
zmena slovosledu	12
platné homoglyfy	27
homoglyfy - cyrilika	-
biele znaky za medzery	0
využitie vrstiev PDF formátu	1
obrázky miesto textu	-

Tabuľka 4.3.Výsledky WCopyFindu



Obrázok 4.1.Porovnávací graf

4.3 Zhodnotenie experimentu

4.3.1 Zhodnotenie výsledkov pre aplikácie

Aplikácia **PlaDes** bola najúspešnejšia z porovnávaných 3 aplikácii (viz. obrázok 4.1), hlavne pre výhodu práce v slovenskom jazyku a vďaka úspešnej metóde LCS - subsequence, ktorá sa ukazuje ako najefektívnejšia - dokáže preskočiť veľa prekážok, ktoré sú koreňmi útokov.

Aplikácia **Ferret** pre jej jedinú metódu 3-gramov ukázala len veľmi slabé výsledky (viz. tabuľka 4.2), a mala problémy so začatím detekcie posledných silných útokov.

Aplikácia **WCopyFind**, ktorá má pomerne silnú detekčnú silu cez jej unikátnu metódu porovnávania a hľadania rovnakých podreťazcov (a nie len jedného LCS), vrátila porovnateľne dobré výsledky (viz. tabuľka 4.3) ako PlaDes. Obsahuje pomerne dobré techniky predspracovania textu ako PlaDes, ale chýba jej synonymický slovník.

4.3.2 Otázky a hypotézy pre nasledujúce experimenty s plánom ich riešenia

1. Je možné na základe porovnania celkovej reálnej veľkosti dokumentu s vypočítanou veľkosťou dokumentu (spočítanie znakov * veľkosť znaku) označiť dokument za podozrivý?
2. Je možné získať z predspracovania dokumentu dostatok informácii pre štatistiky, na ktorých základe bude postavená detekcia útokov homoglyfami a bielymi znakmi?
3. Bude možné zo získaných štatistík experimentálne určiť použiteľné hranice označenia dokumentov za podozrivé?

4. Nespomalí proces detekcie pre ostatné metódy toto získavanie a spracovávanie štatistík?
5. Nakoľko úspešná bude detekcia plagiátorských útokov založená len na štatistických údajoch, bez porovnávania s korpusom?

Nasledujúcimi experimentmi a implementáciou navrhnutých mechanizmov sa pokúsime tieto hypotézy potvrdiť alebo vyvrátiť. Najbližšie experimenty sa budú týkať určovania veľkosti dokumentu a porovnanie tejto veľkosti s vypočítanou veľkosťou dokumentu na základe početnosti znakov, kde sa predpokladá pri obrázkových plagiátoch niekoľkonásobný rozdiel týchto hodnôt, čo by mohlo prvú hypotézu potvrdiť. Jedná sa o pomer množstva obrázkov a skutočného textu. Testované dokumenty budú obsahovať rôzne množstvo textu a obrázkov. Pre druhú hypotézu sa zameriame na efektívne získanie štatistiky znakov a slov počas predspracovania dokumentu, aby sme získali neskreslené štatistické údaje - slová pre štatistiku nesmú byť stemmované ani po lemmatizácii. Nasledujúci experiment bude pozostávať z množstva slovenských textov, kde predpokladáme získanie priemerných hraníc dĺžok slov a početnosti písmen pre našu detekciu. Po preverení tretej hypotézy porovnáme časovú zložitosť pred implementáciou spracovania štatistík a po nej. Predpokladá sa len mierne spomalenie. Predpokladáme značné zlepšenie detekcii útokov, pretože doterajšie výsledky boli veľmi slabé (0% detekcia, viz. tabuľka 4.1), čo potvrdí piatu hypotézu ako pozitívnu. Finálne sa úspešnosť potvrdí porovnaním výsledkov PlaDes-u z prvého experimentu a výsledkov PlaDes-u s našou vylepšenou implementáciou (naše softvérové riešenie).

5. Zhodnotenie

V tejto správe sme analyzovali známe metódy útokov na detekcie plagiátorstva aj s uvedením príkladov a silných či slabých stránok. Následne sme popísali detekčné metódy s analýzou, prečo sú odolné a prečo slabé voči niektorým útokom. Po experimente s aplikáciami PlaDes, Ferret, WCopyFind sme metódy detekcie navzájom porovnali a názorne predviedli ich slabé a silné stránky. Zároveň sme porovnali silu jednotlivých útokov. Po experimente sme vyslovili hypotézy a predpoklady pre ďalšie experimenty. Nakoniec sme navrhli mechanizmy s návrhom ich implementácie na ochranu pred plagiátorskými útokmi, so zameraním na úspešné útoky z predošlých experimentov.

5.1 Plán pre nasledujúci semester

V druhej časti tejto práce sa zameriame na zodpovedanie položených otázok a hypotéz, a na implementáciu navrhnutých mechanizmov do prostredia už testovaného PlaDes-u, kde vykonáme podobné testy so vzorkami slovenských textov. Výsledky porovnáme s výsledkami z experimentov vykonaných v tejto časti - očakávame výrazné zlepšenie v percentách zhody.

Začneme experimentmi s veľkosťou dokumentov a ich obsahom, pokračovať budú experimenty s efektívnym a rýchlym získaním správnych štatistických údajov popri predspracovaní dokumentov a získanie hraníc pre určenie podozrivosti dokumentu. Následne implementujeme spracovanie týchto štatistík a spravíme experimenty s označovaním podozrivých dokumentov na základe spracovaných výsledkov. Finálnu implementáciu otestujeme na väčšej vzorke slovenských textov a porovnáme úspešnosť s predošlou implementáciou PlaDes-u.

6. Bibliografia

1. Kučečka, Tomáš.: Plagiarism Detection in Obfuscated Documents Using an N-gram Technique. *Information Sciences and Technologies Bulletin of the ACM Slovakia, Special Section on Student Research in Informatics and Information Technologies* 3.2 : 67-71, 2011.
2. Schleimer, Saul, Daniel S. Wilkerson, and Alex Aiken.: Winnowing: local algorithms for document fingerprinting. *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 2003.
3. Prechelt, Lutz, Guido Malpohl, and Michael Philippsen.: Finding plagiarisms among a set of programs with JPlag. *J. UCS* 8.11: 1016, 2002.
4. Hirschberg, Daniel S.: Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM)* 24.4: 664-675, 1977.
5. Su, Z., Ahn, R.B., Eom, Y.K., Kang, K.M., Kim, P.J., Kim, K.M.: Plagiarism Detection Using the Levenshtein Distance and Smith-Waterman Algorithm. In: Proceedings of the 2008 3rd International Conference on Innovative Computing, Information and Control, Washington, DC, USA, 2008.
6. Chudá, D., Návrat, P.: Support for checking plagiarism in e-learning. Procedia - Social and Behavioral Sciences, Innovation and Creativity in Education, vol. 2, no. 2, pp. 3140-3144, 2010, ISSN 1877-0428.
7. Chudá, Daniela, Ján Chlpek, and Andrej Kumor.: The impact of text pre-processing to determine the similarity in students assignments. *Proceedings of the 12th International Conference on Computer Systems and Technologies*. ACM, 2011.
8. Lyon, Caroline, Ruth Barrett, and James Malcolm.: A theoretical basis to the automated detection of copying between texts, and its practical implementation in the Ferret plagiarism and collusion detector. *Plagiarism: Prevention, Practice and Policies*, 2004.
9. Balaguer, Enrique Vallés.: Putting ourselves in SME's shoes: Automatic detection of plagiarism by the WCopyFind tool. *Proc. SEPLN*. 2009.